Lecture Notes in Networks and Systems 17

Andrzej Jankowski

Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective

Springer

Lecture Notes in Networks and Systems

Volume 17

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

More information about this series at http://www.springer.com/series/15179

Andrzej Jankowski

Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective

Springer

Andrzej Jankowski
The Piotr Dziubański Foundation
of Knowledge Technology
Warsaw
Poland

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-319-57626-8 ISBN 978-3-319-57627-5 (eBook)
DOI 10.1007/978-3-319-57627-5

Library of Congress Control Number: 2017939091

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

This book would not have come to a reality without the continuous encouragement and patient support of a great teacher and friend, whose presence always worked as a motivation for pursuing research despite of all adverse situations. Thus, the book is dedicated to Prof. Andrzej Skowron, my teacher of Computer Science, Information Technology, and Artificial Intelligence, a human being of great endurance and generosity, and a wonderful friend one feels fortunate to have.

Andrzej Jankowski

Human Interaction, Computational Emergence, Design, Computational Engineering, Adaptive System Infrastructure, Adaptable and Predictable System Quality, Policy, Acquisition, and Management, ... Progress has been made on all these fronts and others.
And yet ... there is a fast growing gap between our research and reality.

—Linda Northrop[1]: Does Scale Really Matter?: Ultra-Large-Scale Systems Seven Years after the Study. Software Engineering Institute, Carnegie Mellon University (2013)

[. . .] *Manipulation of perceptions plays a key role in human recognition, decision and execution processes. As a methodology, computing with words provides a foundation for a computational theory of perception - a theory which may have an important bearing on how humans make - and machines might make - perception-based rational decisions in*

[1] http://2013.icse-conferences.org/documents/publicity/ICSE_Keynote-Northrop.pdf.

an environment of imprecision, uncertainty and partial truth.
[...] *computing with words, or CW for short, is a methodology in which the objects of computation are words and propositions drawn from a natural language.*

—Lotfi A. Zadeh[2]: From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions. IEEE Transactions on Circuits and Systems **45**(1), 105–119 (1999)

Traditional statistics is strong in devising ways of describing data and inferring distributional parameters from sample. Causal inference requires two additional ingredients:
a science-friendly language for articulating causal knowledge, and
a mathematical machinery for processing that knowledge, combining it with data and drawing new causal conclusions about a phenomenon.

—Judea Pearl[3]: Causal inference in statistics: An overview. Statistics Surveys **3**, 96–146 (2009)

[2]Founder of Fuzzy Sets, Granular Computing, and Computing with Words; http://www.cs.berkeley.edu/~zadeh/presentations.html.

[3]The 2011 winner of the ACM Turing Award "for fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning."

Preface

Mathematics and the physical sciences made great strides for three centuries by ***constructing simplified models of complex phenomena****, deriving, properties from the models, and verifying those properties experimentally. This worked because the complexities ignored in the models were not the essential properties of the phenomena.* ***It does not work when the complexities are the essence.***

Frideric Brooks [68]

This book was mainly written on the basis of the author's conclusions, drawn over four decades of acquiring of practical experience. During that time, the author had an occasion to participate in and/or monitor the implementation of about real-life 200 projects.

One of the most important conclusions from the gained practical experience is that one can distinguish some processes playing a relatively crucial role when it comes to the successful implementation of complex projects. In this context, *processes aimed at discovering, processing, using, developing, communicating, and interactively improving comprehension of "important concepts" for practical uses* (especially those relevant to **effective and efficient problem-solving** as part of a given project) are of utmost importance. These "important concepts" are modeled and represented in the book by *complex granules* (*c-granules*, in short) that are used, approximated, and processed by networks of agents. These granules play a primary role in processes aimed at generating, processing, and archiving c-granules by networks of c-granules belonging to a single agent or to a team or a network of agents. Any agent is perceiving other agents or their networks using c–granules.

In other words, the processes aimed at processing "important concepts" are represented and investigated in the book by means of computations performed on c-granules, modeled using concepts and tools based on Interactive Granular Computing (IGrC). In this approach, "important concepts" are approximated and represented by c-granules as well.

It should be noted that “agents” pertain both to human beings and all physical devices that process data (including “intelligent” robots). Networks of these agents may cooperate with one another to solve various types of problems (e.g., related to satisfying the needs of individual agents and/or a network of agents). Networks of agents may also compete with other networks of agents to satisfy certain needs of agents. It should be emphasized that in recent decades, we witness a dynamic development of research conducted by numerous scientific centers and companies, which is devoted to networks and their use in supporting the processes of solving problems that occur in complex systems, especially in the field of Complex Systems Engineering (CSE). This is evidenced, e.g., by extensive scientific literature related to the complex network and CSE topics (e.g., [14, 20, 23, 58, 78, 79, 84, 85, 89, 95, 97, 102–104, 148, 149, 170, 179, 208, 212, 271, 272, 303, 314, 315, 332, 334, 335, 345, 354, 357, 380, 432, 439, 478, 485, 503]).

At this point, we should distinguish some characteristic features of the approach presented in this book, which are related to complex networks of agents and/or complex networks of c-granules. These features include:

1. *Physical nature of networks, c-granules, and computations performed by c-granules considered in IGrC.* This physical nature means that networks, c-granules, and computations are grounded in the physical reality.
2. *Assumption of a limited access of agents and c-granules to knowledge about the world, which can be accessed only through interactions between c-granules and the environment.*
3. *Openness and complexity of the world, leading to occurrence of unpredictable disturbances.* This generates a need to construct numerous models of local aspects with regard to the reality in which an agent is interested. These models have mechanisms for a continuous verification how well they match the phenomena that occur in the surroundings (through continuous interactions) and adaptation of these models according to observed changes.
4. *Assumption that analyzing, modeling, and solving problems related to complex systems cannot be successful without taking into consideration interactions between agents (networks of agents) and the surroundings.* Through interactions, agents may become more and more familiarized with the environment and conceptual tools for solving problems (these tools are represented as networks of c-granules).
5. *Perception of objects can be either by direct means or by indirect means.* Perception by direct means is realized by directly measuring the states of objects, represented, e.g., as the values of attribute or a satisfiability degree of a formula which defines the property of objects. Perception by indirect means is realized by creating an adequate configuration of physical objects and using it for launching in it new interaction processes to be able to identify the states of these objects by measuring the states of relevant measurable objects influenced by these interactions.
6. *Relations with issues concerning the self-organization of complex systems* [302, 420, 555].

7. *Relations with Valiant's ecorithms* [521, 522]. Here, learning of interaction rules by c-granules plays a key role. Learning these rules of interaction helps acquire knowledge, which was not possessed a priori by an agent (agents), and at the same time, plays a significant role in the process of controlling computations performed on c-granules.

The above list does not include all characteristic features of the approach to complex networks of agents and/or c-granules presented in the book. However, they show the key role of interactions in this approach.

It is worth emphasizing that apart from complex networks, there are many more currently developing domains, which are significantly related to various aspects of interactive granular computations in complex systems. In particular, this remark applies to complex systems involving human beings (e.g., social networks) and/or networks of data processing devices. These intensively developing domains include, among others:

a. machine learning, data mining, and pattern recognition (in particular big data analytics);
b. intelligent agents and intelligent multi-agent systems;
c. nonconventional models of computing (in particular computing models based on interaction, ecorithms);
d. cognitive science;
e. decision making under uncertainty, decision support systems;
f. human–computer interaction;
g. self-organizing systems;
h. adaptive control and complex adaptive systems;
i. complex systems engineering;
j. internet and the semantic web, cyber-physical systems, internet of things, wisdom Web of things, and ultra-large-scale systems;
k. complex biomedical systems and signals;
l. social networks;
m. linguistic and cognitive systems;
n. risk analysis, management, and minimization.

More examples of these domains and adequate references to literature may be found in Chap. 3, for example.

In the book, we foreground relations between the above-mentioned domains (particularly from the perspective of IGrC). At the same time, we emphasize that all these domains could be enriched through a more in-depth development of the foundations of IGrC. At the same time, the results achieved within these domains may enrich the knowledge about IGrC.

The above observations are also related to the importance of interactive granular computations performed during the implementation of complex projects. In this context, we have aimed at developing IgrC for controlling interactive granular computations, which model project's implementation processes, so that they contribute to maximizing the chance for a successful completion of a project, while minimizing the emerging risks. With this regard, the **IGrC control of CSE projects should be performed by means of properly selected and enforced (among project participants) project implementation principles**. In particular, the author used both appropriately selected and adapted principles that are commonly known (e.g., [100, 107, 279, 545]) and some other principles that were specifically developed and implemented by him.

The families of principles for the implementation of complex projects (with special regard to the specificity of the projects presented in Part IV of the book), gathered by the author, can be used by sponsors, managers, and complex project engineers. If skillfully applied, these principles may help to increase the efficiency of the project's implementation, decrease the number of project risks, and reduce some negative consequences of various project's activities (including redundant costs), which stem from a *gap between theory and practice* (cf. Part III Chap. 11).

Moreover, the author's conclusions, drawn from years of practical experience, were used in the scientific research, aimed at a deeper understanding of mechanisms related to the functioning and development of complex systems, with special regard to Complex Adaptive Systems (CAS) and to networks of agents and/or CAS.

From the point of view of the book's content, projects specific to CSE seem particularly important. **The projects themselves can be treated as complex adaptive (meta-)systems, aimed at the construction of adaptive complex systems with required properties.**

In his scientific research (e.g., devoted to computational models for CAS), the author focused on various types of CAS, particularly:

1. CAS related to software design and implementation, in which large project teams and teams of stakeholders cooperate in terms of planning and development.
2. CAS, implemented using the Internet of Things [211] and the Wisdom Web of Things (W2T) [445].
3. Complex biological systems that use natural computing [420]. Within these systems, societies of living organisms (e.g., bacteria, insects, birds, fish, mammals) learn various behaviors by acting in societies [420] in order to solve problems of the biological survival (e.g., colonies of bacteria [42]).
4. General models of CAS [543], including current mathematical models of learning systems [119, 400, 521, 527], neural network technology [8, 36, 50, 126, 137, 340], as well as *ecorithms* and computational models in ecological *niches* [210, 521].
5. CAS related to computational models of social life [338].

These systems include "intelligent" agents (e.g., human beings, animals, and/or "intelligent" robots), that are engaged in various interactions. In general, agents have their needs and they try to fulfill them on their own, in cooperation and/or by competition.

The author's reflections presented in the book concentrate around the problems related to:

A. The identification and analysis of the causes that lead to the widely discussed in the literature gap between theory and practice (cf. [19, 279, 530, 531]) during the implementation of CSE projects, especially those, that result in the creation of new software systems and technologies based on Artificial Intelligence (AI). These reflections are based on the conclusions drawn from the implementation of projects, presented in Part IV of the book (Case Studies), as well as on numerous publications and expert reports devoted to this subject (cf. Part III of the book).
   Numerous scientists and engineers agree with the thesis postulated by Linda Northrop, which is expressed in the first motto of this book. It states that despite a great progress in the research on techniques related to the implementation of very large CSE projects, we may still witness a fast growing gap between our research and reality.
B. The development of techniques for modeling mechanisms of "intelligent" interactions between agents, e.g., human beings and/or robots and computations realized by these interactions in complex systems (particularly, CAS systems). By "intelligent interactions," we mean, e.g., influencing the surroundings to stimulate behaviors, aimed at fulfilling the agent's prioritized needs. We assume that these needs include:

   1. the need related to the *perception* (understood as the process by which it is possible to comprehend the perceived situations) and interpretation of interactions, occurring in a system and its environment (especially those, needed to plan and implement the survival and development of an agent, that is as "comfortable" as possible);
   2. the need related to the agent's control over interactions to ensure her/his fulfillment of needs;
   3. the need related to the agent's understanding of perceived interactions to increase the efficiency of her/his activities, e.g., for boosting cooperation or competition among agents;
   4. the needs related to the processes of learning concepts which represent the priority needs of an agent and improving techniques used to fulfill these needs.

C. The development, on the basis of appropriate computational models (cf. B), of a set of advanced technical tools for:

   1. a deeper understanding of the causes that lead to the gap between theory and practice (mentioned in point A),
   2. the reduction of negative consequences of this gap (mentioned in point A),
   3. the maximization of increase in the quality (including the efficiency) of computations, modeled by means of the techniques, mentioned in point B.

When it comes to scientific achievements, the following significant elements should be mentioned:

I. Descriptions of selected projects designed and implemented under the author's supervision (POLTAX, AlgoTradix, Merix, Excavio). In these descriptions, a special emphasis is put on *principles* for development of CSE projects, which were, to a large extent, obeyed during the implementation of the projects presented in the book, in particular, in Parts III and IV, and also in Appendices. These principles were mainly used to control the processes of design and implementation of the projects in order to maximize the chances of achieving the expected results.

II. The analysis of conclusions from designing, modeling, and implementing complex projects, carried out and/or monitored by the author, provide some advices in searching for solutions of the problems, mentioned in points A, B, and C. This was achieved mainly by establishing the foundations for constructing computational models, which provided the basis for the creation of supporting tools for solving A, B, and C problems in complex systems, and for the synthesis (including the design and implementation) and analysis of complex systems. The most significant results include:

   II.a. The formulation of appropriate computational models for analyzing problems and techniques of reasoning about computations, realized using these models, as well as techniques for the representation of knowledge that support the process of reasoning. This encompasses:

      II.a.1. Proposals of IGrC models as the basis for supporting the synthesis (including the design and implementation) and analysis of complex projects and/or systems (especially in terms of the research devoted to problems, mentioned in A, B, and C above).

      II.a.2. Proposals of techniques for the acquisition and representation of knowledge (e.g., by means of interaction principles that support the planning and implementation of the agent's priority needs). It applies to the acquisition and representation of both general knowledge and domain knowledge, which supports the efficient use of information possessed by an agent and/or a society of agents in particular projects or systems.

   II.b. The development of techniques for controlling computations performed by agents (or their societies) in complex systems. The aim of this control is to fulfill the adaptively changing needs of an agent (or a society of agents). These techniques include the basics of computational control in IGrC, with special regard to particular uses, mentioned in A, B, and C above. These basics rely on the author's approach to computational efficiency management in IGrC (including risk and co-risk), performed by agents (or societies of agents) that carry out computations.

Re I.
The book describes selected original achievements, related to the design and implementation of such complex projects as POLTAX, AlgoTradix, Merix, and Excavio.

The POLTAX system (cf. Chap. 16) has been used and developed in Poland for about a quarter of a century. The system ensures an effective replenishment of the Polish budget resources through tax administration (which encompasses, in particular, the central register of taxpayers, accountancy, and tax (PIT, CIT, VAT) settlement).

It should be emphasized that the tax system reform, which took place at the beginning of the 90s in Poland, played a crucial role in the modernization of the Polish economy, based on the so-called Balcerowicz Plan.[4] Owing to the implementation of POLTAX, it was possible to reform the Polish tax system, so that it meets the requirements of the Polish-UE integration, and tailor it to the country's market economy. Consequently, the new tax system (based on POLTAX) paved the way for a significant economic growth of Poland, as it is illustrated in Fig. 16.3 (based on the data from the World Bank[5]). For about a quarter of the century, POLTAX has been effectively supporting the collection of taxes for the Polish budget. According to the Central Statistical Office of Poland,[6] in 2015, around 75% of taxable revenues collected for the Polish budget by the Ministry of Finance were supported by POLTAX from PIT, CIT, and VAT taxes. The growth rate of the net taxes on products[7] collection in Poland is illustrated by Fig. 16.2, prepared on the basis of the data from the World Bank.[8]

POLTAX and other projects implemented under the direction of the author of this book (BGŻ, BGK, PKN ORLEN, PERN/OLPP, RUCH, etc.) are mainly based on the Polish work culture. However, the conclusions presented in the book are also based on the author's experiences from projects implemented in countries with significantly different work cultures and traditions, and with different historical, political, and technological background.

The projects implemented or monitored by the author include:

(a) key projects implemented for the purposes of public administration during the first three decades of systemic changes in Poland, starting from 1989;
(b) research and educational projects implemented by academic centers, as well as R&D and private companies (in Poland, the USA, and Japan);
(c) projects related to industrial applications implemented in Poland, the USA, and Japan.

[4]https://en.wikipedia.org/wiki/Balcerowicz_Plan.

[5]http://data.worldbank.org/indicator/NY.GDP.MKTP.CD?end=2015&locations=PL-UA-SE&start=1990&view=chart.

[6]http://www.finanse.mf.gov.pl/documents/766655/5018330/wplywy_12_2015.pdf.

[7]http://en.eustat.eus/documentos/opt_0/tema_44/elem_3365/definicion.html.

[8]http://data.worldbank.org/indicator/NY.TAX.NIND.CD?end=2015&locations=PL-UA-SE&start=1991.

To illustrate the diversity of the author's practical experiences in the implementation of scientific projects devoted to the development of AI-based technologies (differing in specificity from the POLTAX project), let us list two examples that will be analyzed in detail further in the book:

1. Developing the concept and architecture of an interactive system called Excavio—an "intelligent" search engine for extracting information from a network of document servers. The technologies proposed by the author and developed under his direction as part of the project, described in (cf. Chap. 19), led to the submission of two patent applications in the US Patent and Trademark Office, together with Zbigniew Michalewicz. These scientific and industrial patents were related to the development of artificial intelligence, particularly, the techniques of intelligent dialog document search in the Internet. According to Google Scholar, the patents[9,10] have got around 260 citations and they were referred to by the leading R&D and industrial centers, run by such giants as: IBM, Google, Microsoft, Yahoo!, Hewlett-Packard, Oracle, Saor Kabushiki Kaisha, Matsushita Electric Industrial, MusicMatch, NEC, Endeca Technologies, Fuji, Xerox, Canon, Boeing, Hansen Medical, Sony, Accenture, and Red Hut.
2. Developing the concept and architecture of an algorithmic trade system, implemented on international financial markets (Forex). The concept and architecture were implemented, developed, and introduced by the AdgaM Group (2006–2011) as part of the AlgoTradix project, presented in (cf. Chap. 17). The softbots (i.e., software robots) constructed in this project were tested for one and a half years (2009/IX–2011/III) on numerous international financial platforms (the USA, EU and Japan) and on real money on the OANDA platform (NY, the USA). On the OANDA platform, the robots automatically performed about 24,000 positions, of which 80% were won. The AlgoTradix project was mainly based on technologies of interactive granular computations, described in the book, which were performed by the societies of softbots as part of the Wisdom Technology (WisTech) research development program [246, 247, 445]. The project provided a substantial empirical material for the scientific research on the development of techniques for modeling computations in IGrC, performed by multi-agent systems described in this book.

Re II.
The conclusions presented in the book are the result of long-term and intensive experimental works and scientific research, conducted by the author. The key conclusions are described in more detail in Chaps. 13–36 of the book.

[9] https://scholar.google.pl/citations?view_op=view_citationhl=pluser=zVpMZBkAAAAJcitation_for_view=zVpMZBkAAAAJ:u5HHmVD_uO8C.

[10] https://scholar.google.pl/citations?view_op=view_citationhl=pluser=zVpMZBkAAAAJcitation_for_view=zVpMZBkAAAAJ:9yKSN-GCB0IC.

The formulation of a fundamental cause (precause) of the gap between theory and practice is one of these conclusions (cf. Chap. 13). Some very important conclusions are briefly presented in II.a and II.b.

Re II.a.

In the book, *interactive granular computations* (*computations in* IGrC, for short) were proposed as the basis of computational models and IGrC techniques based on *adaptive judgment*, as techniques for reasoning about the properties of these computations.

Computations in IGrC are performed by *complex granules* (*c-granules*, in short) thanks to which it is possible to register, analyze, and synthesize the properties of interactions among physical objects perceived by agents. In particular, c-granules may be used to support techniques for reasoning about properties of interactive computations.

We assume that the states of certain physical objects, occurring within a specific domain of activity of a given c-granule, are directly recognizable and/or measurable. However, the states of other objects are perceived (approximated, recognized) indirectly by measurable states, through interactions of physical objects from a particular domain of c-granule's activity. Each measurable state of a c-granule (at a given moment of the agent's time) corresponds to a concept. This concept is understood as a set of situations (configurations of physical objects), perceived within this very c-granule, and, thanks to interactions, leads to a specific state. In the approach proposed in this book, the concept of a measurable state means the possibility of representing such a state by, e.g., the value of a corresponding attribute or by the *satisfiability degree* of corresponding concepts/formulas, which represent these concepts. Following the aggregation of c-granules, more complex c-granules, corresponding to structural objects, their properties, or relations over measurable states (e.g., *preference relations*), can be constructed.

Thanks to c-granules, it is possible to register both the results of sensory measurements and their hierarchical aggregation, which is performed to discover new c-granules. The hierarchical c-granules discovered in this manner may ensure a deeper understanding of a perceived situation (cf. [27, 117]). The statement above about the aggregation of c-granules (representing hierarchical aggregations of the results of sensory measures) refers to the main, according to Valiant,[11] AI challenge, which is the characterization of "computational building blocks" [522] for perception.

Adaptive judgment plays a crucial role in the assessment of what is currently important, and what is less important for an agent (from the point of view of hierarchy of her/his needs). Therefore, it constitutes the basis for the evaluation and improvement of interaction plans that are being implemented. In a sense, judgment [181, 183, 263, 276, 349, 524] may be treated as an elaboration of the concept of rational reasoning (especially about the properties of computations in IGrC) due to the

[11] http://www.seas.harvard.edu/directory/valiant.

necessity of taking into account not only mechanisms of logical reasoning, but also constraints and other mechanisms that influence decisions, which are being made. These mechanisms pertain, e.g., to perception, emotions, instinct, habits, intuition, fast thinking [263], and experience. Thus, adaptive judgment is not only limited to deduction, induction, abduction, and/or other techniques of logics. A deeper understanding of the concept of adaptive judgment should be also supported by psychology (in searching for c–granules representing patterns of behavior) and phenomenology (e.g., in making decisions based on experience) [151, 328].

The key role in the approach proposed in this book is played by the techniques of adaptive and interactive discovery of c-granules (through interactions with the environment) and their further use.

It turns out that in order to perform computations on c-granules, *ecorithms*, as understood by Valiant [521], should be used instead of classical algorithms.

Apart from the analogy to Valiant's ecorithms, the IGrC-based algorithms proposed in the book display a number of other features, which correspond to the motivations of scientific research in other domains (e.g., learning systems, CAS, soft-computing, multi-agent systems, natural computations). IGrC models are also related to the very foundations of AI, in particular, to the understanding of the essence of machine learning.

When applying the approach proposed in this book, **the design and implementation of a complex project may be seen as the process of discovering, learning, processing (including communicating), and developing concepts (represented as c-granules), which are necessary to complete a given project**.

The issues related to point II.a are discussed in detail in Chaps. 22–36. Moreover, in Chap. 32, the Framework Postulates for WisTech (FPW postulates) are presented. These postulates constitute frameworks (constrains), within which computations in IGrC should be performed. Due to their complexity, these postulates are often formulated using complex vague concepts from a natural language and thus, must be approximated by agents, who use other languages (e.g., formalized languages). The problems related to this phenomenon refer to the motto of this book, which includes statements by Lotfi Zadeh and Judea Pearl.

Another difficulty is that these approximations are learnt in an environment, in which the acquisition of knowledge is possible mainly through the agent's perception of certain properties of physical objects. Therefore, these difficulties directly affect the issues related to perception, in particular, the process of learning the principles of operation, based on the agent's perception of interactions between physical objects. A great size and complexity of ontologies of concepts for expressing the FPW postulates constitute other problems.

Re II.b.

Problems related to point II.b involve adaptive learning of **principles that govern an evolutionarily changing game, defined as a set of pairs (a concept, defining the conditions required for launching a given action, and an action and/or interaction plan that is being launched when the concept is satisfied to an adequate degree)**. This process of learning is carried out by agents (and/or their societies). In these principles, a significant role is played by learning the approximations of complex and, most often, vague concepts, which, when satisfied to an adequate degree, signalize the need to take some actions (or interaction plans), e.g., actions that protect against a growing risk or actions that support the enforcement of positive results, potentially stemming from a growing co-risk (cf. FPW-13 in Chap. 32).

Moreover, when making a decision about launching or stopping a given action (or interaction plan) in a current situation, we deal with a need to settle conflicts between arguments "for" and "against" the satisfiability of complex, vague concepts, which also decide about the launching of particular actions or interaction plans. In general, the process of reasoning, which leads to a decision about choosing a given action and/or interaction plan that is to be launched (or stopped), is based on adaptive judgment, which takes into account such elements as the assessment of risk involved in a current situation, knowledge about possible consequences of a given action/interaction plan that is to be implemented, or disturbances caused by interactions with the surroundings. In practice, we may find numerous recommendations as for the conditions and actions of launching controls against various types of risks among the existing standards relating to risk management. However, **updating this type of knowledge and sharing it with complex systems (including societies of artificial agents) is still a great challenge**.

When using contemporary standards, it is worth remembering that the classical approach to risk management is—generally—based on stationary foundations of the ontology of concepts which define risk management. On the other hand, according to the approach presented in this book, the semantics of concepts which define risk management (including the concept of "risk" itself) evolves with time. The dynamics of changes to these concepts depend both on the techniques of interactions between agents and the environment and on unknown external factors.

The reflections above show that the concept of ontology, understood as a conceptual apparatus of agents (or a society of agents) used for perceiving, planning, acting, learning, and communicating, plays a key role in IGrC. Moreover, the concepts used by agents constitute the primary building blocks of interaction principles, which provide the basis for controlling computations to satisfy the adaptively changing needs of agents (or societies of agents).

It is worthwhile mentioning that the ontologies present in IGrC computational models, which comply with the FPW postulates described in Chap. 32, are very complex. Moreover, the processes of creating and selecting an essential ontology for acquiring, representing, sharing, and updating domain knowledge, characteristic of a specific IGrC application, are also very complex. These ontologies are based on numerous vague concepts and relations between them.

Moreover, it is important to take into account some serious problems which are encountered while trying to acquire and develop ontologies (especially those, which involve vague, complex concepts) by artificial agents. These problems concern, among others, the need to understand numerous complex, vague concepts by artificial agents. For this purpose, some efforts may be made to help agents in the process of learning the approximations of such ontologies (cf. [27]). However, current results of the research in this field require a further elaboration. Another problem is related with a need to provide artificial agents with an access to techniques of reasoning on concepts from those ontologies, which mimic the reasoning in a natural language at least in an approximate sense. The reader must have noticed that these problems directly refer to the challenges formulated in a form of the motto of this book, which is the statement by Lotfi Zadeh and Judea Pearl. The aforementioned problems appear especially when efforts to share the experience related to managing complex projects with multi-agent systems are being made.

Currently, we are nowhere near developing fully effective techniques for transforming and filtering human experiences related to managing projects into knowledge accessible to specific complex systems consisting of artificial agents, who could take over some of these functions from human beings and share them with “intelligent” robots, ensuring, at the same time, that they are properly used.

On the other hand, due to the fact that there are no perspectives for developing effective techniques of automatic learning and discovery of complex knowledge, let alone its effective use in a way that is analogous to that of human beings, it seems inevitable for complex systems to use knowledge that is accessible to them. The aforementioned lack of perspectives is related to the complexity of the search space, within which concepts necessary to express either interaction principles for controlling interactive computations in IGrC or (semi-) optimal principles for the implementation of CSE projects are being discovered. These concepts may be particularly related to experiences in managing projects.

We have outlined some of the aspects that are important for modeling and control computations of complex systems. This outline shows the complexity of FPW, described in more detail in Chap. 32, which involve all aspects mentioned above. These postulates should constitute key constraints for controlling interactive computations in complex systems by agents.

The results of the research presented in this book may also be analyzed from the point of view of their potential contribution to advancements in dynamically developing scientific disciplines, such as CSE [356, 424], granular computational models [394], interactive computational models [169], models of natural computing [210, 420], models of learning systems [521], and models of computations performed by CAS [543] or multi-agent systems [121, 311, 488].

The main ideas presented in the book have their roots in the research on rough sets, initiated by Zdzisław Pawlak [381, 383, 385, 463]. At this point, we are particularly referring to Pawlak’s approach to concepts such as concept approximations, information systems, decision tables (as they are understood in a rough set theory), and reasoning about vague concepts.

The research fields presented above are described in more detail in the introduction to the book (cf. Part I). It includes such chapters as in-depth research motivations (Chap. 1), research objectives and approaches to selected directions of searching for solutions (Chap. 2), and challenges of WisTech related to CAS Modeling (Chap. 3). It also contains a general description of scientific research results (Chap. 5) and a guide to the book’s content (Chap. 4). Moreover, in the Part I of the book the references to the relevant publications are presented.

In this book there are many quotations. The crucial fragments (from the point of the book) of the quotations are bolded by the author of the book.

The approach to IGrC computations proposed and presented in the book as a scientific achievement of the author is a continuation and expansion of the research, conducted together with prof. Andrzej Skowron and described in several publications (e.g., [244, 245, 245–249, 252, 361, 444–448, 459, 460]).

Warsaw, Poland Andrzej Jankowski

Acknowledgements

This work was partially supported by the Polish National Science Centre grant DEC-2011/01/D/ST6/06981, the grant No. O ROB/0010/ 03/001 within the frame of the Defence and Security Programmes and Projects: “Modern engineering tools for decision support for commanders of the State Fire Service of Poland during Fire and Rescue operations in buildings.”

The author extends his expression of gratitude to Prof. Janusz Kacprzyk, Dr. Thomas Ditzinger, and the series “Lecture Notes in Networks and Systems” staff at Springer for their support in making this book possible.

Warsaw
February 2017

Andrzej Jankowski

Contents

Abbreviations

AC	Axiom of choice
AD	Axiom of determinacy
AI	Artificial intelligence
BCR	Benefit–cost ratio
BOT4IGC	Basic ontological task for the IGrC
BPCD	Basic principles of CSE project development
CAS	Complex adaptive systems
CBR	Case-based reasoning
CMMI	Capability maturity model integration
CPS	Cyber-physical system
CPSs	Cyber-physical systems
CPSS	Cyber-physical systems of systems
CSE	Complex systems engineering
CSE project	Complex systems engineering project
CWW	Computing with words
Deputy PM	Deputy Prime Minister
EEM	Engine for efficiency management
FEAF	Federal enterprise architecture framework
FP3C	Fundamental precause of CSE crisis
FPW	Framework postulates for WisTech
GrC	Granular computing
HR	Human resources
IGrC	Interactive granular computing
infogranule	Information granule
IoT	Internet of things
IRGrC	Interactive rough granular computing
IT	Information technology
IV&V	Independent verification and validation
MACM	Main aim of the CAS monitoring and/or controlling
MT	Model of thinking problem

PAC	Probably approximately correct
PC-TPGP	Principal causes of TPGP
PDCA	Plan-do-check-act (plan-do-check-adjust)
PM	Prime Minister
PoF	Principle of project failure
PoS	Principle of project survival
SAE	System architecture engineering
TPGP	Theory–practice gap problem
UAV	Unmanned aerial vehicle
ULS-S	Ultra-large-scale systems
W2T	Wisdom web of things
WisTech	Wisdom technology
Y2K	Year 2000 problem

List of Figures

List of Tables

Part I
Introduction: Research Motivations, Approaches, Challenges, and Overview of Results

Chapter 1
Research Motivations

> *Some say biology is the science of the 21st century, but information science will provide the unity to all of the sciences. [...] Information science, the understanding of what constitutes information, how it is transmitted, encoded, and retrieved, is in the throes of a revolution.*
>
> – David Baltimore, Nobel Prize Winner in biology, professor emeritus at Caltech
> www.caltech.edu/content/caltech-launches-new-information-initiative

1.1 "Understanding Intelligence" as the Greatest Problem of the 21st Century

Many scientists consider understanding intelligence as the greatest contemporary problem in science and engineering. This postulate was formulated by Tomaso Poggio and Stephen Smale in [400] as follows:

> The problem of understanding intelligence is said to be the greatest problem in science today and greatest problem for this century - as deciphering the genetic code was for the second half of the last one. Arguably, the problem of learning represents a gateway to understanding intelligence in brains and machines, to discovering how the human brain works and to making intelligent machines that learn from experience and improve their competences as children do. In engineering, learning techniques would make it possible to develop software that can be quickly customized to deal with the increasing amount of information and the flood of data around us.

The article was published in 2003. It is remarkable that today, many scientists—particularly in IT—still believe that **the problem of understanding intelligence is said to be the greatest problem in science and engineering for this century**.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_1

Faced with the waves of disturbances related to the financial crisis, after the bankruptcy of Lehmann Brothers in 2008,[1] many scientists started to show more respect to holistic attempts at understanding intelligence (in particular, to AI applications in risk management).

The common opinion has begun to dominate that it is worth solving simpler sub-problems, related to the problem of how to understand "intelligence." In other words, it has become increasingly popular to contend that it would be too difficult to perform such a broadly discussed task and that some attempts should be made first to attain simpler objectives and to develop criteria of their acceptance.

In 1950 Alan Turing [510] proposed some criteria for recognition of human type of "intelligent machine." In some sense, he reduced it to the problem of creating "thinking" and natural language "processing" machine. Until today we have many discussions about accuracy and/or modifications of Turing approach to the "intelligent machines." In particular, one can find an interesting proposal of an extension of the "Turing approach" to problems of perception and action. For example, in [375] one can find:

> The Turing test, as originally conceived, focused on language and reasoning; problems of perception and action were conspicuously absent. The proposed tests will provide an opportunity to bring four important areas of AI research (language, reasoning, perception, and action) back into sync after each has regrettably diverged into a fairly independent area of research.

An important example of sub-problems of the problem of "understanding of intelligence" might be the need to understand the mechanisms of **learning of techniques for reasoning about approximations** of *complex vague concepts* [34, 218, 268, 269, 387, 417, 443, 458] involved in the decision-making process. It should be emphasized that we understand concept of *learning of techniques for reasoning about approximations* in a wide sense. In particular, this also concerns **the perception (based on interactions) of situations necessary for decision support and/or for learning.**

The question of learning and employing, by a machine, human's way of interpreting vague concepts and reasoning about vague concepts is an issue, which is frequently analyzed in studies over the phenomena that accompany diagnosis of causes for economic disturbances [176].

A common trait of these diagnosis is considering complex vague concepts as notions that are inexpressible by simple deterministic, measurable, and irrefutable criteria. Instead, one must assume that the criteria related to such concepts are subject to a **systematic enhancement through interactive learning of how to discern objects, which fall under such a concept and those which do not**. A particularly important emphasis is placed here on the systematic enhancement of simulated modeling and stress testing processes.

The concept of risk is important in so far as it concerns all forms of complex human activities, aiming at achieving a certain goal. In such situations, there always exists a risk that the goal may not be attained. Naturally, the risk is not the only

[1] http://en.wikipedia.org/wiki/Lehman_Brothers.

practical vague complex concept that is important. Concepts which appear in natural languages, that lack precise criteria needed to define their scope, are usually vague.

When describing any real life scene and/or its constraints, we often employ vague concepts such as rich, poor, sick, healthy, tall, short, strong, weak, beautiful, ugly, heavy, light, warm, cool etc. If we use them to describe action plans or product/service requirements, we must bear in mind that as a consequence, our descriptions or specifications will also be vague. Junior adepts of software engineering often neglect the fact that the majority of functional specifications of complex software systems remain vague. Thus, during the developmental phase, they do not make adequate efforts to interactively assess the meaning of vague concepts, contained in those specifications. They do not spend enough time on discussions with the future users in order to remove any potential doubts related to the meaning of vague concepts included in specifications. Such a behavior usually leads the failure of a given project.

Certainly, not all concepts display vague features. For instance, in the definitions of many concepts, there are intrinsic unambiguous criteria which clearly indicate whether a given object or phenomenon is satisfying a given concept or not. These are called crisp concepts.

For example, most formal mathematical concepts are crisp. The postulate that **vague concepts require a different treatment from that of crisp concepts** was first put forth by Gottlob Frege, one of the creators of the fundamentals of modern mathematics and logic: "the concept must have a sharp boundary. To the concept without a sharp boundary there would correspond an area that had not a sharp boundary-line all around" [147, 157]. He paved the way for founding modern mathematics on a premise of a crisp concept, namely the concept of a set understood in the Cantor's sense [83, 292].

However, most of the concepts in natural languages are vague concepts [74, 118, 157, 268, 269, 387, 417, 443, 471, 542]. In the opinion of many researchers, phenomenon of dynamic rules of interpretations for "vague concepts" is an essential feature of language. In particular, it is a part of concept of "languages games" developed by Ludwig Wittgenstein [17, 539]. According to this approach, the natural language of human beings **is not a formal calculus with rigid rules, that can provide precise meaning to all possible circumstances**.

The importance of vague concepts led to the development of many approaches to models of computation, which help handle and manage these concepts. Perhaps the first well-known and widely used approach was fuzzy logic by Lotfi Zadeh [547]. One of the later approaches for computing on vague concepts was the rough set approach by Zdzisław Pawlak [384]. He understood vague concepts in the Frege's sense, by using a boundary area of a concept i.e. *an area that had not a sharp boundary-line all around*.

The above introduction is just a very brief preliminary to describing the degree of difficulty when it comes to understanding the process of learning complex vague concepts (such as the risk of a global economic crisis) and the process of reasoning about these concepts. It also illustrates the importance of learning vague concepts and presents a more general understanding of the human intelligence directly related to the attempts at comprehending and modelling human thinking, communicating, and thought-refinement processes.

It is therefore hardly a surprise that many aspects of the research as well as related achievements in this field were the central themes of the latest Turing Awards (sometimes the Turing Award is called the *Nobel Prize for IT*).

If we are to agree that the Turing Award is a reliable index of the most important scientific and technological issues in modern IT, then, the ones awarded, e.g., for 2010–2012 clearly display a common feature—they all touch upon the questions on how to understand the mechanisms involved in the treatment and acquisition of complex knowledge as well as the models of intelligent thinking. The awarding verdicts of ACM[2] can also be viewed as a distinction of people, who in the ACM's assessment, captured the most important research directions and thus can shape the priorities of the future IT research. To illustrate those verdicts, below, we present selected fragments of the Turing Award citations[3]:

> 2010 – ACM Turing Award Goes to Innovator in Machine Learning. Valiant Opened New Frontiers that Transformed Learning Theory, Computational Complexity, and Parallel and Distributed Computing. [...] Valiant has contributed to computational neuroscience, offering a concrete mathematical model of the brain and relating its architecture to complex cognitive functions. In his 1994 book Circuits of the Mind, he details a promising new computational approach to studying the intricate workings of the human brain. It focuses on the brain's ability to quickly access a massive store of accumulated information during reasoning processes despite the extreme constraints imposed by the brain's finite number of neurons, their limited speed of communication, and their restricted interconnectivity.
>
> 2011 – Judea Pearl developed Novel Framework for Reasoning under Uncertainty that Changed How Scientists Approach Real World Problems. [...] His contributions to causal reasoning have had a major impact on the way causality is understood and measured in many scientific disciplines, most notably philosophy, psychology, statistics, econometrics, epidemiology and social science.
>
> 2012 – Goldwasser, Micali Receive ACM Turing Award for Advances in Cryptography. MIT Researchers' Innovations Became Gold Standard for Enabling Secure Internet Transactions. [...] One of the most significant contributions of Goldwasser and Micali is their 1985 paper with Charles Rackoff, titled "The Knowledge Complexity of Interactive Proof Systems." It introduced knowledge complexity, a concept that deals with hiding information from an adversary, and is a quantifiable measure of how much useful information could be extracted.

The Turing Awards indicate that currently, learning of approximations of complex vague concepts, aimed at building and securing useful complex knowledge, is of particular importance. It would be best to elicit and securely store the knowledge that is used to design the architectures of human-level intelligence for non-trivial problem solving and learning through understanding languages. At this moment, we are still far from attaining a solution for this problem in practical applications.

Obviously, there are IT practitioners who deem that they are faced with completely different problems in their professional careers and may not share the view of ACM on the most important achievements and research problems in IT.

---

[2] https://en.wikipedia.org/wiki/Association_for_Computing_Machinery.

[3] http://amturing.acm.org/alphabetical.cfm.

For example, they may not agree with the suggestions made by the winners of these three Turing Awards with regard to the future research directions. Nevertheless, the decision of ACM and the suggestions on the future research directions, associated with it are to be taken seriously into account. These suggestions clearly show the importance of understanding and modeling human thinking, communicating, and thought-refining processes, especially when it comes to modeling computational models that facilitate learning, discovery, and processing of knowledge, based on a proper management and treatment of complex vague concepts.

A wide range of areas in which this particular topic can be applied means that the results of the research in this field will have a great impact on economy, medicine, agriculture, and many other domains.

1.2 Personal Research Motivations Based on the Author's Technical CSE Experiences

1.2.1 Author's Technical Experience: Development of AI Applications and CSE Project Management

For several decades, the author of the book has had a unique opportunity to acquire practical experience through his active engagement in numerous projects, involving the application of Information Technology (IT) and development of AI technology applications. Moreover, for more than two decades, he stood in charge (mainly as project manager) of the development, implementation, planning, verification, and deployment of many highly complex and diverse projects.

These projects were devoted to the application of IT in areas such as: banking, logistics, storage of petroleum products, algorithmic trading, international electronic financial markets, governmental administration, tax administration, research and development centers, as well as many educational and academic enterprises (cf. Part IV). These projects were conducted by people with different work cultures, mainly from Poland, the United States, Japan, France, Germany, and the UK at different stages of their organizational and technological development.

In relation to the above-mentioned experience the universal nature and significance of a gap between theory and practice (cf. Part III, in particular Chap. 11) about the Theory-Practice Gap Problem (TPGP)) (Part III) become more apparent.

The problem is particularly visible in the case of projects related to CSE, implemented in a partially unknown and dynamically changing environment. Let us note that TPGP is directly related to the concept of the CSE crisis (cf. Chap. 9).

It should be strongly emphasized that, in the context of the book, by *CSE projects*, **we understand those projects that are carried out in a dynamic and unpredictably changing environment with imperfect (imprecise, uncertain, incomplete, and unreliable) data and knowledge. This means that re-implementation of similar projects, carried out in predictably changing environments about**

which all relevant information is available, do not constitute, in our sense, good examples of CSE projects.

He also had occasion (as a contractor or assessor of some of the tasks) to observe and participate in discussions related to numerous scientific initiatives, in which he did not directly play an active role. He was particularly interested in the projects related to the developments within the field of Artificial Intelligence (AI) technology application. In the context of the conclusions presented in this book, the UAV project is particularly interesting. Chapter 20 presents some important conclusions regarding hierarchical classifiers, which played a significant role in the development of AI technologies in the above-listed projects, including AlgoTradix, Merix, and Excavio (cf. Chaps. 16–20). The scope of the projects, presented in Part IV of the book, can be described as follows:

1. POLTAX (cf. Chap. 16) is one of the largest IT systems deployed in Central and Eastern Europe in the 20th century. Its main task was to introduce a reform of the Polish tax system in Poland, following a deep transformation in the country, which occurred at the beginning of the 1990s, from a centrally-planned socialist economy system to a free market economy that conforms with the EU standards. The project is the result of a conceptual study, which was planned in 1990 under the guidance of the author of this book. He developed and, to a large extent, implemented the foundations of the system, related with its shape, functioning, and development, in a form of CSE principles of the POLTAX system (cf. Sects. 14.3 and 16.3).
   Thanks to the implementation (supported by the BULL company) of the initial versions and proper modules of the POLTAX system, it was possible to properly launch and implement the PIT and CIT taxation in 1992, as well as the VAT taxation initiative in the mid-1993. Currently (i.e., in 2017) POLTAX, is still functioning. It successfully supports the tax system in Poland based on PIT/CIT/VAT. The POLTAX system architecture consists of the following functional modules (which constitute **co-operating subsystems of POLTAX**):
   a. registration, which includes the Central National Taxpayer Register (CRP KEP),
   b. declarations,
   c. taxpayer assessment,
   d. accounting,
   e. fines,
   f. monitoring,
   g. debt collection,
   h. automatic recommendations for tax audit,
   i. dedicated data warehouses,
   j. reporting,
   k. ontological foundations.

It should be emphasized that the tax system reform, which took place at the beginning of the 90s in Poland, played a crucial role in the modernization of the Polish economy, based on the so-called Balcerowicz Plan.[4] Owing to the implementation of POLTAX, it was possible to reform the Polish tax system, so that it meets the requirements of the Polish-UE integration, and tailor it to the country's market economy. Consequently, the new tax system (based on POLTAX) paved the way for a significant economic growth of Poland, as it is illustrated in Fig. 16.3 (based on the data from the World Bank).[5]

For about a quarter of the century, POLTAX has been effectively supporting the collection of taxes for the Polish budget. According to the Central Statistical Office in Poland, in 2015, about 75% of taxable revenues collected for the Polish budget by the Ministry of Finance were supported by POLTAX from PIT, CIT, and VAT taxes. The growth rate of the tax collection in Poland is illustrated in Fig. 16.2, prepared on the basis of the data from the World Bank.[6]

2. AlgoTradix (cf. Chap. 17) is a system of algorithmic trading by an "intelligent," autonomously operating "society of software robots," designed for trading on international financial markets. The system was used, among others, on the OANDA platform in New York City, where it performed about 25 thousand transactions on real money. About 80% of these transactions were winning transactions. Some other performance indicators (cf. Fig. 17.13), are the following:

    a. annualized return of investment: 18%,
    b. annualized daily volatility: 10.2%,
    c. maximum drawdown of closed positions: 4.5%,
    d. maximum drawdown of all positions (including open positions): 10.6%.

    More performance indicators one may find in Chap. 17. In particular, a summary of AlgoTradix results is presented on Figs. 17.11–17.21.
3. Merix (cf. Chap. 18) is a system for the automatic fraud detection on on-line credit card transaction streams. The pilot versions of this system were presented to the Bank of America, whereas its modifications to PKN ORLEN. The project and the system under the same name are presented in Chap. 18 of this monograph.
4. Excavio (cf. Chap. 19) is one of the first systems in the world to support automatic "intelligent" dialogues between a user and a Web search engine. Some of the ideas developed as a part of the Excavio system were patented. According to Google Scholar,[7] the patents, of which the author of the book is a co-author (together with Zbigniew Michalewicz), are cited by the leading research centers in the world (about 260 citations). The project and the system under the same name are described in Chap. 19 of this monograph.

---

[4] https://en.wikipedia.org/wiki/Balcerowicz_Plan.

[5] http://data.worldbank.org/indicator/NY.GDP.MKTP.CD?end=2015\&locations=PL-UA-SE\&start=1990\&view=chart.

[6] http://data.worldbank.org/indicator/NY.TAX.NIND.CD?end=2015\&locations=PL-UA-SE\&start=1991.

[7] http://scholar.google.pl/citations?user=zVpMZBkAAAAJ\&hl=pl.

5. UAV (cf. Chap. 20) is an abbreviation for the Unmanned Aerial Vehicles. This project is presented in [117]. The main goal of the research, undertaken as a part of the project (presented in Chap. 20 of this book), was to develop and test a system that could, e.g., recognize potentially dangerous situations on a highway. The whole process was to be performed by an autonomous drone, that is an unmanned rotary-winged-aircraft. The author had a chance to observe the development of the project and discuss various aspects of AI technologies, implemented as a part of the project. Chapter 20 presents some important conclusions regarding the hierarchical classifiers, which played a significant role in the development of AI technologies in the above-listed projects, including AlgoTradix, Merix, and Excavio.

In addition, the author's experience includes many other complex projects for global market players (GM, Ford, BoA), and projects for numerous Polish organizations (e.g., The Ministry of Finance, PKN ORLEN, RUCH, BGŻ, BGK, PWPW, PERN/OLPP).

Selected projects are presented in Chaps. 16–20 of the book. Each such chapter consists of the following thematic sections:

1. Background, genesis and goals.
2. Examples of CSE principles.
3. Brief history.
4. Main results.
5. Conclusions.

Along with initiating and managing these projects, the author usually actively participated in their implementation. In this way, he implemented and verified many personal ideas related to the design and technological solutions. At this point, a question of which of particular experiences enabled him to achieve the most significant results when considering individual projects, may be asked.

Moreover, the author (being a member of the Polish Prime Minister's IT Council (Polish PM's IT Council)), had a chance to observe, and, often asses the development and emergence of virtually the largest initiatives undertaken by the Polish government, during the 1990s.

Besides the projects mentioned above (in particular projects described in Part IV of the book), the author also took part in several other CSE projects of high complexity. For example, the ICRA[8] [286] project project (cf. Sects. 35.3 and 33.4) is one of such initiatives, which deserves particular attention.

One of the objectives of the ICRA project was to enable the commander of a fire and rescue action to receive potentially important messages from a system with questions and/or suggestions related to whether the commander did not overlook anything important in a complex situation which is currently perceived by him/her. The ability of the system to conduct a "dialogue" with fire-fighters (with different responsibilities in a fire and rescue action) is an important, functional aspect.

[8] http://icra-project.org/?node=10.

This dialogue should be held, on one hand, in a language that is comprehensible to fire-fighters (that is possibly similar to the natural language based on vague, complex concepts) and, on the other hand, in a language that is comprehensible to systems for automatic processing of knowledge. In modern systems, this knowledge is often expressed in some formal language that is based on approximations of complex, vague concepts. More details regarding the author's experiences in developing proposals of algorithms that could potentially support such dialogues can be found in Chap. 19 devoted to the Excavio project. Section 35.3 presents a brief proposal of pilot version of the ICRA project, which is an attempt to use the aforesaid approach to fire and rescue actions. The ICRA project has got the 2016 INPEX International Invention Award.[9, 10]

As part of this initiative, some aspects of risk management in commanding fire and rescue operations have been verified and tested. The implementations of projects was based on the simplified versions of the concepts presented in this book (cf. Sect. 33.4), using the approaches towards the IGrC models, based on WisTech.

1.2.2 PC-TPGP: A Summary of Conclusions Related to Causes of TPGP

Having analyzed an extensive literature and drawing conclusions from practical experience, the author has undertaken scientific research with the aim of diagnosing the *Causes of TPGP* and proposing some tools, that could reduce the gap between theory and practice in CSE projects. In this monograph, we present the results of his work.

To simplify, the summary of the most important **conclusions, drawn from the above-mentioned experiences and analyses, come down to the following observations**, called **PC-TPGP**:

1. The principal cause of the TPGP is the lack of appropriate techniques for monitoring and/or controlling interactive computations of Complex Adaptive Systems (CAS). This especially is related to **lack of effective and efficient techniques for monitoring and/or controlling the CAS phenomena relevant for the CSE** project implementation through **interactions of available resources (including networks of cooperating and/or competing agents) in the open environment**. This kind of CAS monitoring and/or controlling encompasses the **processes of monitoring and/or controlling of interactive computations related to CAS phenomena as well as to reasoning** processes about important properties of these computations.

[9] https://www.sgsp.edu.pl/aktualnosci-sgsp/sukces-projektu-icra-podczas-targow-inpex-2016-733.html.

[10] https://www.inpex.com/.

2. From the point of view of TPGP, the **Main Aim of the CAS Monitoring and/or Controlling (MACM)** is to achieve—by the networks of cooperating and/or competing agents—sufficient level of understanding of information processing mechanisms for **achieving goals (identified using continuous aggregations of the agent's needs from networks of agents) through learning the effective and efficient principles of wise cooperation and/or competition among human beings and/or the societies of artificial agents**, especially in rapidly changing and only partially known monitored environments.
3. WisTech proposes a methodological approach to **MACM** by **construction of appropriate models of interactive granular computations related to discovery, analysis, and/or aggregation of important properties** of CAS processes especially useful for monitoring and/or controlling of CAS phenomena.
4. One of main technical difficulties with constructing models of interactive granular computations *(related to discovery, analysis, and/or aggregation of important properties of CAS processes especially useful for monitoring and/or controlling of CAS phenomena)* is called the **Fundamental Precause of CSE Crisis (FP3C)**.
5. The essence of the **FP3C** concerns the problems with an appropriate (i.e. aimed at achieving **MACM**) modeling, monitoring, and controlling of **interactive aggregations of many local computational models, which interact in an open environment, as well as technical difficulties with the development of techniques for reasoning** of agent network about important properties of such models (especially important for effective and efficient support of achieving MACM by the agent network).
   These *local computational models* are often **aggregated in a hierarchical manner and keep changing in an open environment**. They are used next for generating relevant interactions with the environment using networks of sensors and/or actuators embedded by the control of agents in the physical environment.

Since the first half of the 20th century, numerous attempts have been made to construct computational models that could provide at least a partial solution to the CAS modeling problem. Most of these attempts were the variations of such computational models, based on ideas embedded in Turing machines, cellular automata, Monte Carlo simulation based models, soft computational models, artificial neural networks-based models, as well as models for increasing the accuracy of approximations, evolutionary strategies, and self-organization [59, 110, 130, 136, 161, 162, 184, 287, 311, 343, 365, 405, 420].

Unfortunately, these models did not meet many expectations, due to the fact that being abstract constructions, they were in some sense independent of the physical reality, and their implementation was based on a closed-world semantics. Their hermetic nature derives, among others, from the fact that these models do not take into account interactions with the real physical world. For example, they do not provide effective techniques for learning (non-trivial concepts) through interactions with the real physical world. Moreover, they do it in a too simplistic or improper way (e.g., often, random variations of parameters, inconsistent with the reality, were applied to a given computational model). In the CSE practice, this is one of the major factors that leads to a situation in which theoretical applications, derived from computations carried out by these models, become gradually in-compatible with models obtained by adaptive systems, "open" to interactions with the reality.

As shown by the arguments above, a natural demand for alternative computational models and techniques for reasoning about computations based on these models has appeared. The research on new approaches to computational modeling constitutes one of the objectives of this book. The results of the research include a proposal for the development of models based on Interactive Granular Computing (IGrC) with techniques for reasoning about the properties of computations in IGrC. This approach is based on the paradigms built around the *Wisdom Technology* (WisTech) [244–247], an idea developed in cooperation with Professor Andrzej Skowron which encompasses the metaphorical equation known as the *Wisdom Equation* (cf. Eq. 8.1 in Chaps. 8 and 29).

1.3 Economic Motivations

In many sources one can find estimations concerning the negative impact of Information Technology (IT) failures on the global economy. For example[11]:

> **experts calculated the global impact of IT failure as being** 3 **trillion annually.**
>
> [...] What reward can we expect through better management, operational excellence and governance of IT?
>
> [...] That's a staggering amount of value, 4.7 percent of global GDP, or more than the entire economic output of Germany.

Of course, there are many other calculations[12] as for the global economic impact of IT failures and there are many reports, which explain why these fail[13] (e.g., [54, 153, 336, 368]). Many sources provide different recommendations for preventing failures of IT projects. However, the majority of experts agree that IT failures exert a considerable, detrimental influence on the world's economy. Moreover, most of them agree that we do not have any "silver bullet" to solve this very costly, for our civilization at least problem.

Based on the author's experience, in a country similar to Poland, the resulting loss can be measured in billions of dollars within a decade. In the book, we propose some *engineering principles* which proved contributory in implementation of projects, generating significant cost savings and increasing the chances for success of particular projects.

Some of these principles are general in nature (e.g., ABC book of CSE projects, Basic Principles of CSE Project Development (BPCD), cf. Sect. 14.3). However, it

[11] http://www.zdnet.com/article/worldwide-cost-of-it-failure-revisited-3-trillion/.

[12] http://blog.capterra.com/surprising-project-management-statistics/.

[13] http://www.ukessays.com/essays/computer-science/software-crisis.php, https://insights.sei.cmu.edu/sei_blog/2015/08/.

should be noticed that these principles have roots in principles of civil engineering[14] dating back to the ancient times. They have been known for more than 6 thousand years. For example, some basic principles of engineering can be found in ancient Mesopotamia, especially the basic principles of urban engineering[15] (cf. [288]). In [288] one can read:

> From before 4,000 BCE, over the next ten to fifteen centuries, the people of Eridu and their neighbors laid the foundations for almost everything that we know as civilization. It has been called the Urban Revolution [...]

There are many books about "basic principles" devoted to complex IT projects. Most of them repeat (in a slightly different language) quite similar principles. One of the first ever known books proposing such a family of principles was "The Mythical Man-Month" by Fred Brooks [68]. Brooks' observations are based on his experiences at IBM, while managing the development of OS/360 announced in 1964. The book presents some kind of a tendency of managers to repeat similar errors. It has often been called The Bible of Software Engineering as it covers basic principles and values, which are important for the development of any complex software system. However, as with any bible, the truth is that *almost "everybody" quotes it, some people read it, but unfortunately, only few people follow it.*

Some engineers say that general engineering principles for IT system do not work because each complex project has its own very unique specificity. According to this view, it is impossible to identify any practically useful universal family of principles for complex IT projects. Each type of project has to follow some unique principles, adjusted to its specific nature. This can be done based on practical experiences of project managers.

In Part III, IV, and in Appendixes we present a set of proposals for specific families of principles in two types of projects:

1. Projects for public administration sector, especially for a country with a similar maturity to Poland, in the midst of its economic transformations.
2. Projects aiming at developing new AI-based technologies for applications, especially for the US market and international electronic financial markets (such as algorithmic trading on Forex).

One of the main messages of the book is that engineering principles can be treated as some kind of "software" that govern the processes of engineering of complex systems. On the other hand, these processes can be controlled by computational models, based on IGrC. Thus, the principles of modeling such kinds of computations can play an essential role in Complex Systems Engineering (CSE). In Chap. 32 we propose a family of postulates, which provides some conceptual tools specifying a basis for construction of models of IGrC. The family of postulates in Chap. 32 can be treated as a generalization of the architectural engineering principles, discussed as

[14] https://en.wikipedia.org/wiki/Civil_engineering.

[15] https://en.wikipedia.org/wiki/Architecture_of_Mesopotamia, http://www.ancient.eu/article/678/.

art of case studies included in Part IV. Moreover, in the following chapters, one can find conceptual tools which can be used to improve risk management and efficiency management for such general models of computing and for CSE. Moreover, a set of tools for risk and efficiency management based on IGrC is also provided.

According to the author's observations concerning hundreds of IT projects carried out in the past decades, all complex and publicly founded IT projects (implemented in a country with a similar maturity to Poland) and/or **all AI projects which fail to follow appropriate principles of ABC book, BPCD, and/or other specific principles discussed in the book are very likely to fail**.

On the other hand, any deeper understanding and/or improvement of engineering principles for complex IT systems should entail a higher likelihood of project success. The improvement of such principles is one of the main objectives of modeling in IGrC. A simplified version of these principles is presented as an illustration of an interactive computing model in Chap. 36 (cf. Points igc-1 - igc-18). However, it should be emphasized that the illustration does not encompass numerous important aspects, that have been discussed in this book. The basic catalogue with the most important aspects of the model may be found in the chapter on FWP (cf. Chap. 32). From the point of view of economic motivations, it is worth mentioning the concept known as the Engine of Efficiency Management (EEM), which is described in Chap. 33. The main components of the EEM framework architecture are presented in Fig. 33.5.

1.4 Summary of Research Motivations

The research motivations presented above (cf. Sects. 1.1–14.4) encompass some common research problems. In this section, we present key intuitions behind of an example of such common research problem.

This example concerns a problem which can be formulated using a very simple sentence.

Namely, it is a problem of **construction of efficient techniques for agent's c-granule control aiming at achieving the top agent's needs specified by vague agent's concepts (assuming the agent has only access to imperfect knowledge and data e.g., uncertain, incomplete, noisy) about the environment**.

The problem is expressed by a simple sentence, but according to many researchers, it can be treated as one of the most important and challenging problems of the XXI century (e.g., cf. Sect. 1.1).

In order to explain the intuitions of the problem, let us consider a complex physical **object**. In particular, the **object** can be one of the following examples:

1. **population of viruses** attacking healthy cells of an animal,
2. bacteria **colony** solving critical problems for bacteria colony survival (e.g., learning—by a colony of bacteria—how to overcome the human immune system and/or how to overcome the latest versions of antibiotics),

3. **artificial neural network**[16] solving some complex AI problems,
4. complex multilayer and/or multiplex networks[17] [58],
5. some concrete thoughts and/or emotions of a given person represented in **brain** of the person,
6. falling **snowball** through the mountain slope, which creates an avalanche,
7. some concepts (or parts of ontology) used for communication among agents, creating a **representation of corresponding conditions initiating actions** which can be launch (or should be launch/forbidden) after recognition of some properties in the current agent's environment,
8. stage of a **CSE project**,
9. some concrete thoughts and/or emotions of a given society represented by corresponding **brains and/or media**,
10. any **CAS system** (it could be treated as a generalization of CSE project).

Notice that, each of the above objects, is subject of continuous interactions in an open environment.

In particular, the object can be partially perceived by an agent. Moreover, sometimes the object can be recognized and/or interpreted in relationships to the *hierarchy of needs* of the agent. The agent uses her/his *adaptive judgment* processes for current prioritization of properties and/or behaviour patterns which are currently "really" very important/attractive from the point of view of her/his currently the most important needs. In order to make decisions corresponding to the question: *what is more important?* the agent may select or construct a decision using an aggregation of arguments *for* and arguments *against*.

Summing up, the agent based on the results of her/his interoperation, may use the object to **register, analyze, synthesize, discover, and/or learn** (important in terms of the agent's hierarchy of needs) properties and/or behaviour patterns of interactions among physical objects. After identification of important properties (to satisfactory degrees) of interactions in the environment, the agent may use the object to **launch some interaction plans for achieving desired properties** of the environment.

In the preface we have introduced some intuitions of one of the most important concepts of the book, called *c-granule* (for more detailed presentation of the c-granule concept cf. Chaps. 28–32). Any of the above examples of objects can be treated as an example of **objects perceived by agents using c-granules**. Any agent is aiming to achieve important goals specified by some vague agent's concepts. She/he can use c-granules for implementation of interaction plans. However, in order to do it, first it is necessary to construct interaction plan (i.e., relevant control techniques for agent's c-granules).

Having in mind the above considerations, it should be obvious that the problem of construction of efficient techniques for agent's c-granule control, is really very difficult. Moreover, in the matter of fact, it is an example of common research motivation problems occurring in all Sects. 1.1–1.3 and 14.4.

[16] https://en.wikipedia.org/wiki/Artificial_neural_network.

[17] http://scitation.aip.org/content/aip/journal/chaos/26/6/10.1063/1.4953595.

Chapter 2
Research Objectives and Selected Approaches

I don't demand that a theory correspond to reality because
I don't know what it is.
Reality is not a quality you can test with litmus paper.
All I am concerned with is that
the theory should predict the results of measurements.
Stephen Hawking [196]

2.1 The Main Research Objectives

This book includes a concise summary and analysis of the conclusions, drawn from the completion of many complex projects, both described in literature and experienced by the author himself.

To a great extent, these conclusions concern a scientific exploration of new tools for reducing the discrepancy between the theory and practice of CSE.

An important conclusion drawn from these experiences and their analysis is that it is necessary to consider these new tools as new types of computational models and new techniques for reasoning about these computational models. At the same time, these new computational models and techniques for reasoning about them should ensure a better understanding and a more skillful use of such concepts as "information processing", "learning complex concepts", "intelligence", and "wise interactions among human beings and artificial agents in a dynamic and only partially known environment." In this book we propose to treat **IGrC models for WisTech as such tools.** Essential part of presented approach to models of computation are techniques for monitoring, control of interactive computations, and agents reasoning techniques about interactive computations performed in the physical world.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_2

The main research objectives of the book encompass, in particular, the following objectives:

1. To develop proposals for counteracting the most significant causes of discrepancies between the theory and practice of CSE projects, based on personal experiences and achievements in the design, construction, and technological development of complex IT projects.
2. To present proposals of CAS computational models, with the aim to achieve goals of MACM (cf. Sect. 1.2.2). The models should support CAS monitoring, controlling, and reasoning about the properties of these computations.
   These models should also, in particular, be designed to simulate key functionalities of a wise teamwork (among human beings and artificial agents) in a partially known, dynamically changing environments, with particular focus on monitoring the performance of such activities as:
   2.1. Decision-making and action-taking processes performed by agents, based on the satisfiability (to a certain degree) of concepts that cover the following aspects of these processes:
      2.1.1. Identification of complex and time-changing concepts that are important to agents.
      2.1.2. Adaptive learning approximations of concepts at the level of details best adapted to the needs of agents.
      2.1.3. Discovery of concepts that are currently needed for the efficient fulfillment of the agents' goals.
   2.2. Cooperation and/or competition across the societies of agents (humans and artificial) in such actions as:
      2.2.1. Creating new technological solutions for the application in AI.
      2.2.2. Counteracting substantial risks (e.g., fire and rescue operations, based on the experience from the ICRA project).
3. To propose techniques for reasoning about the properties of computations, carried out within the already proposed computational models, with an emphasis on such applications as CSE project management, creation of practically applied AI techniques, and substantial risk suppression.
4. To apply computational models and models of reasoning about the properties of computations to:
   4.1. Improve the identification and understanding of TPGP causes (especially FP3C).
   4.2. Reduce the gap in TPGP.

It should be noted that **this book in no way aspires to present a complete solution to the problems presented as part of the main research objectives, as well as in the previous chapter, entitled "Research Motivations"** (cf. Chap. 1).

The author of this book agrees with the view of many specialists (e.g., Sect. 1.1), who consider the aforesaid problems to be among the most important and most difficult problems of the 21st century's scientific research. In some sense, they constitute sub-problems of the problem with "understanding the intelligence."

Intuitively speaking, a **satisfactory solution to these problems would most likely ensure an effective automation of a search for the solutions to many core issues in virtually all areas of the modern life** (e.g., Wisdom Web of Things (W2T), cf. [556, 557]). **The consequences of such an effective automation could be comparable to the consequences of the introduction of information processing breakthroughs, such as the introduction of writing, printing, or a global introduction of computers and computer networks.**

For now, however, we have to remember that we are only approaching this next stage of transformation.

Hence, this book is intended merely as a "small nano-step" on the way to solving the aforesaid problems. It is also worth mentioning that the results of the research described in the book include original achievements in the design, construction, and technological development of many CSE projects. In Part IV of the book, we present examples of projects in which the architecture and implementation of solutions were based on the author's personal experience.

2.2 Selected Approaches to the Objectives

In the chapter entitled "Research Motivations" (cf. Chap. 1), we discussed some relationships between the subject matter of the book and certain approaches to CAS (in particular to CSE). In this section, we describe relationships between particular problems with various approaches to understanding and modeling computations that simulate intelligent decision-making processes.

Of course, concepts such as knowledge, logic, wisdom, etc. have their roots in the earliest civilizations. Certainly, gigantic achievements in this respect were made in ancient Greece, as well as in India and China. Today, our daily life, permeated by the technologies of inter-operating computer networks, opens up new horizons for the creation of innovative products, both in terms of quality and quantity. This has a crucial impact on the development of science. As a result, scientists who investigate modern interactive computational models, including models of information processing, are considered the avant-garde of contemporary science. Rozenberg and Kari [265], expressed this in a slightly different way:

> In these times brimming with excitement, our task is nothing less than to discover a new, broader, notion of computation, and to understand the world around us in terms of information processing.

With the progress in scientific research, there emerged new paradigms for the design, construction, and development of new technologies, aimed at the integration of physical, communicational and information processing processes. A classic example is the research on the so-called cyber-physical systems (CPSs).[1] CPSs refer to the previously discussed ideas, e.g., [75]:

> It is necessary to remark that there is an ongoing synthesis of computation and communication into a unified process of information processing. Practical and theoretical advances are aimed at this synthesis and also use it as a tool for further development. Thus, we use the word computation in the sense of information processing as a whole. Better theoretical understanding of computers, networks, and other information processing systems will allow us to develop such systems to a higher level.

Another very important direction, which combines various results of the research on computational models to support AI techniques is undoubtedly Natural Computing. In [115], it is characterized as follows:

> Natural computation is a study of computational systems including the following:
>
> 1. Computing techniques that take inspiration from nature for the development of novel problem-solving techniques (artificial neural networks, swarm intelligence, artificial immune systems, computing on continuous data, membrane computing, artificial life, evolvable hardware, self-organizing systems, emergent behaviors, machine perception).
> 2. Use of computers to simulate natural phenomena; and
> 3. Computing in nature (by natural materials) (e.g., information processing in evolution by natural selection, in the brain, in the immune system, in the selforganized collective behavior of groups of animals such as ant colonies, and particle swarms, quantum computing, molecular computing, DNA computing, biocomputing, neural computation, evolutionary computation, biological computing/organic computing).

Many studies have been undertaken in this direction and published [420]. Their authors all agree that interactions constitute an important common feature of natural computing.

Among others directions, research concerning various trends in natural computing and software engineering led to the creation of the so-called Interactive Computing. It has been characterized, for example, in [169] as follows:

- Computational problem is defined as performing a task, rather than (algorithmically) producing an answer to a question.
- Dynamic input and output modeled by dynamic streams which are interleaved; later values of the input stream may depend on earlier values in the output stream and vice versa.
- The environment of the computation is a part of the model, playing an active role in the computation by dynamically supplying the computational system with the inputs, and consuming the output values from the system.
- Concurrency: the computing system (agent) computes in parallel with her/his environment, and with other agents that may be in it.
- Effective non-computability: the environment cannot be assumed to be static or effectively computable; e.g., it may include humans, or other elements of the real world. We cannot always pre-compute input values or predict the effect of the system's output on the environment.

[1] http://cyberphysicalsystems.org/, http://ercim-news.ercim.eu/images/stories/EN97/EN97-web.pdf.

The search for computational models that meet the above-mentioned criteria of interactive computation led to the creation of many research directions. These directions include, in particular, very interesting models of quantum computing, which stimulated the birth of the so-called quantum information technology. The following observation regarding this trend, presented in [4], is interesting:

> In particular, the quantum informatic endeavor is not just a matter of feeding physical theory into the general field of natural computation, but also one of using high-level techniques developed in Computer Science to improve on the quantum physical formalism itself, and the understanding thereof. We highlight a seemingly contradictory phenomenon: passing to an abstract, categorical quantum informatics formalism leads directly to a simple and elegant graphical formulation of quantum theory itself, which for example makes the design of some important quantum informatic protocols completely transparent. It turns out that essentially all of the quantum informatic machinery can be recovered from this graphical calculus. But in turn, this graphical formalism provides a bridge between techniques of logic and computer science, and some of the most exciting developments in the mathematics of the past two decades.

From the point of view of this book, Granular Computing (GrC) [21, 394, 546, 549, 554] is no less important than the aforesaid research on interactive computing models. The historical roots of contemporary research in this area can be traced back to the roots of algebraic techniques in logic which can be found, among others, in the works of Gottfried Wilhelm von Leibniz, George Boole, Afred Tarski, Adolf Lindenbaum, Helena Rasiowa, Roman Sikorski, William Lawvere, Myles Tierney, and many other prominent scientists. Certainly, modern research on GrC and AI largely reflects the ideas by Lotfi Zadeh on both fuzzy logic and the CWW paradigm. It also refers to the work of Zdzisław Pawlak, who founded one of the main pillars of this scientific book - the theory of rough sets [384, 386, 389–391]. For more information about the research on GrC one can refer to [21, 393, 394].

It is also worth remembering that the approach presented in the book refers to concepts that are central to AI.[2] The field of AI has developed dynamically since the mid-20th century. One of the first computational models for AI was based on the attempts to model the behavior of neural networks. These models are still being developed and refined: in particular, the research on the so-called reinforcement learning is one of the results of this development. An introduction to this interesting work was presented, for instance, in [490]. In this approach, as well as in many other approaches towards the construction of computational models for AI applications, we encounter a fairly fundamental and common barrier to a technological advancement. To simplify, this barrier means that modern AI solutions are good to some extent, depending on the representation of internal local knowledge, which basically constitutes the domain knowledge of experts about some specific cases and knowledge elicited from available data. Modern systems still have a fairly weak adaptability of action schemes for continuously changing environments. Therefore, these systems are intuitively limited by the closeness of their local world. Many researchers have already paid attention to this. The problem was particularly clearly

[2] http://en.wikipedia.org/wiki/Artificial_intelligence.

manifested a few decades ago when developing one of the first large expert systems, called MYCIN [73].

It turned out that in a particular sense, a “perfect” system (at the stage of design and development) would fail in practical applications due to the difficulty with adapting to the changing environmental conditions, in which it is used. In the following years, some attempts were made to solve this problem, using paradigms that were better adjusted to for implementing the mechanisms of adaptation. A particularly interesting solution was proposed by John Holland in his work entitled “Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule- based systems” [209]. Three decades after the publication of Holland’s work, some progress has been made in this direction, not yet a fully satisfactory one [348, 506, 519].

Many researchers believe that overcoming these barriers requires a better understanding of computational models which simulate the key functionality of a wise teamwork (between human beings and artificial agents) in only partially known, dynamically changing environments. A particular role in such processes is played by the techniques of simulating thinking and communication mechanisms, such as dealing with natural languages. In other words, we encounter here quite fundamental technological barriers to the development of our civilization. The subject of the research in the field of understanding the functionality of natural languages has a very long history and resulted in a number of highly complex approaches. In this book, we refer to the approach to natural language, initiated by Ludwig Wittgenstein, who considers the meaning of language as a form of a game, which he calls language game [539].

It should be noted that the above-mentioned approaches to the subject matter of this book are not exhaustive and it is impossible to discuss them all in such a brief summary. More information on this topic can be found in other chapters of this book. However, it is undoubtedly worth mentioning that there is growing need among researchers to depart from a classical model of computations, which dominates in modern mathematics and is based on various modifications of the Turing machine; instead, they lean towards computational models which are based more on the physical laws than on mathematical laws. There are many important studies and scientific books which emphasize the need for acknowledging physical aspects in computational models [44, 112, 299, 438, 505]. In order to better explain the context and motivation of these research directions, let us cite [112] (cf. p. 266):

> The Turing machine was an abstract construct, but thanks to subsequent developments in computer science and engineering, algorithms can now be performed by real automatic computing machines. The natural question now arises: what, precisely, is the set of logical procedures that can be performed by a physical device? The theory of Turing machines cannot, even in principle, answer this question, nor can any approach based on formalizing traditional notions of effective procedures. What we need instead is to extend Turing’s idea of mechanizing procedures, in particular, the procedures involved in the notion of derivability. This would define mathematical proofs as being mechanically reproducible and to that extent effectively verifiable. The universality and reliability of logical procedures would be guaranteed by the mechanical procedures that effectively perform logical operations but by no more than that. But what does it mean to involve real, physical machines in the

> definition of a logical notion? and what might this imply in return about the 'reasonableness' or otherwise of the effectiveness of physics in the mathematical sciences?

After asking these questions, the author of [112] comes to a pretty intriguing conclusion, which is largely coincident with the properties of interactive granular computational models, as presented in this book. This intriguing conclusion, found on page 268, is as follows [112]:

> It seems that we have no choice but to recognize the dependence of our mathematical knowledge (though not, we stress, of mathematical truth itself) on physics, and that being so, it is time to abandon the classical view of computation as a purely logical notion independent of that of computation as a physical process.

Chapter 3
Challenges of WisTech (Based on IGrC) for CAS Modeling, Controlling, and Monitoring

Computer Science has something more to offer to the other sciences than the computer. In particular, in the mathematical and logical understanding of fundamental transdisciplinary scientific concepts such as interaction, concurrency and causality, synchrony and asynchrony, compositional modelling and reasoning, open versus closed systems, qualitative versus quantitative reasoning, operational methodologies, continuous versus discrete, hybrid systems and more, Computer Science is far ahead of many other sciences, due in part to the challenges arising from the amazing rapidity of the technology change and development it is constantly being confronted with. One could claim that computer scientists (maybe without realizing it) constitute an avant-garde for the sciences in terms of providing fresh paradigms and methods.

– S. Abramsky and B. Coeske [4]

Roughly speaking, a complex system can be understood as the one whose the elements are difficult to separate. We may find examples of complex systems all around us [18, 346]: cells are composed of interacting molecules, brains are composed of interacting neurons, societies are composed of interacting individuals, ecosystems are composed of interacting species. We have already emphasized that in order to design complex systems, it is important to understand the concept of interactions. Without interactions, particular elements can become separated. However, when relevant interactions occur, the elements co-determine their future states. Thus, the future state of an element cannot be determined in isolation, as it co-depends on the states of other elements, precisely of those interacting with it.

One of the basic concepts of the WisTech approach [244–247] is the concept of a *complex granule* (*c-granule*, for short). In order to introduce the key intuitions behind it, let us recall the concept of a hunk first. Following the proposal put forward by

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_3

Heller in [201], any physical object and/or phenomena existing in a spatiotemporal space is called a hunk (i.e., portion of matter). To cite Heller:

> [...] physical objects are four-dimensional hunks of matter.

The fundamental intuition behind the concept of a c-granule is the following:

The control of an agent ag uses her/his c-granules for perceiving and/or accessing fragments of the surrounding physical world. Intuitively speaking, each c-granule consists of three architectural layers:

1. *Soft_suit*, i.e., configurations of hunks which represent the properties of the ag's environment of activity (including the properties of the present, past, and expected phenomena, as well as expected properties of interaction plans and/or the results of some interactions, potentially activated by a c-granule).
2. *Link_suit*, i.e., communication channels (links) which transmit the results of interactions among accessible fragments of the agent's environment of activities and the results of interactions among different representations of properties in the soft_suit; according to the weight (significance) of the current ag's needs, links may have assigned priorities, which reflect the results of judgment, performed by ag.
3. *Hard_suit*, i.e., configurations of hunks accessible by links from the link_suit.

C-granules are generated by an agent ag depending on the specific configurations of hunks related to the ag. It should be underlined that any typical active c-granule is a dynamically changing entity. It means that all components of c-granules (i.e., soft_suits, link_suits and hard_suits) are usually subject to continuous changes. To illustrate this intuition one can use a metaphorical representation of a c-granule as a blind person with a white cane, which represents a link to the physical world (cf. Figs. 28.2, 29.1, and 29.2). The intuitions behind the concept of a link are illustrated in Fig. 32.1. A c-granule can represent concept defining criteria which should be satisfied to initiate an appropriate action. An agent perceives the environment through interactions and learns interaction rules (e.g., conditions, concepts, and actions) to interact with it and achieve agent's goals. The intuitions behind such a game are illustrated in Fig. 28.7. Notice that the soft_suit (i.e., the mirror in Fig. 28.2) reflects past and/or current results of interactions transmitted by agent's link. Moreover, in the link_suit may find the expected results of agent's interaction plans. An agent may compare the results of current interactions with the expected ones. She/he may also use them to initiate next actions and generate new interaction plans. In such a process, a given c-granule is continuously changing. The agent's control can use some c-granules to modify other c-granules. This can be treated as some kind of a metaphor of algebraic operations in mathematics.

In general, c-granules particularly support the following activities of an ag:

1. Improving the techniques of representing the agent's hierarchy of needs and perceiving these needs, as well as the relations between them.
2. Interpreting and judging the importance of phenomena that occur in the agent's environment of activity: in particular, judging the causes and consequences of par-

ticular properties of relationships between these phenomena (from the perspective of the agent's hierarchy of needs).
3. Constructing, initiating, implementing, verifying, adapting, and terminating interaction plans for the identification and fulfillment of the agent's top needs.
4. Communicating, cooperating, and competing with other agents.

In order to support the above-mentioned activities performed by c-granules the agent's control has to construct, using m-hunks (cf. FPW-05 in Sect. 32.3.5), various types of models that mirror the environment. For this purpose, she/he can use spatiotemporal maps (cf. FPW-02 in Sect. 32.3.2), agent's interaction models (cf. FPW-08), and/or interaction plans (cf. FPW-08 in Sect. 32.3.8). In other words, the agent's control can use c-granules to model certain parts of the environment. The aforesaid models can be related to some parts of the world from the past, presence, and future (e.g., models for predicting of some properties of the environment). The models should be adaptive in relation to the perceived differences between the expected and actual properties of the environment. The models can be dynamic and four-dimensional (i.e., spatiotemporal) or even multi-dimensional. In general, the models can be classified into the following three types:

1. *Directly accessible models*, i.e., models related to certain parts of the world, which may be (and/or might have been) directly accessible by links of agent's c-granules. In such situations, an agent, using interactions, can build a mereology and compute some properties of the already identified parts of the model.
2. *Indirectly accessible models*, i.e., models related to parts of the world, which are not (and/or were not) directly accessible. This include, e.g., models of distant galaxies in space and/or predictive models (models of the forecasted weather or market situation). In such situations, the construction of models is performed using an appropriate method of reasoning about accessible models and/or with the help of communication (or learning) with other agents.
3. *Mix accessible models*, i.e., models which combine directly accessible models and indirectly accessible models.

Using c-granules, it is possible to deal with interactions (cf. e.g., Chap. 32). When developing IGrC models for CSE [19, 52, 107, 190, 191, 210, 279, 377, 424, 530, 531], **it was necessary to decide how to deal with several other important, long lasting, and challenging issues that appear in domains different from CSE**, such as:

- complex networks [14, 20, 23, 58, 78, 79, 84, 85, 89, 95, 97, 104, 148, 149, 179, 208, 212, 271, 272, 303, 314, 315, 332, 334, 335, 345, 354, 357, 380, 432, 439, 478, 485, 503],
- computational intelligence [262], including soft computing: in particular, fuzzy sets [262, 431, 495, 547], rough sets [262, 384, 386, 389–391, 449, 450, 463],
- cognitive science [80, 81, 86, 126, 131, 133, 146, 151, 154, 159, 258, 270, 295, 325, 327, 364, 366, 396, 407, 436, 478, 480, 487, 488, 502, 537],

- data mining, machine learning, and pattern recognition (in particular big data analytics) [5, 7, 27, 36, 134, 193, 224, 274, 293, 323, 344, 400, 439, 446, 490, 521, 526, 527],
- CAS [18, 87, 109, 158, 210, 311, 338, 346, 543], including self-organization [59, 110, 130, 136, 161, 184, 287, 289, 302, 311, 343, 365, 405, 406, 413, 420, 423, 488, 555],
- multiagent systems (MAS) and intelligent agents [87, 124, 161, 199, 217, 302, 311, 332, 340, 364, 367, 435, 491, 532]
- GrC [21, 393, 394, 546, 548–554],
- natural computing [420, 444], including: neural networks [8, 50, 262], evolutionary computation and swarm intelligence [262],
- decision making under uncertainty, decision support systems [262],
- human-computer interaction [66, 319],
- adaptive control [12, 18, 56, 98, 109, 130, 158, 178, 213, 214, 298, 338, 346, 364, 543],
- internet and the semantic web, cyber-physical systems, internet of things, wisdom web of things, and ultra-large-scale systems [1, 211, 368, 369, 379, 404, 556, 557],
- complex biomedical systems and signals [14, 64, 95, 103, 138, 271, 430],
- social networks [199, 302, 338, 432, 488], in particular, computational social choice [67],
- linguistic systems [103, 127, 137, 195, 275, 367],
- risk management systems [15, 57, 69, 92, 176, 200, 253, 433, 545],
- the origin and evolution of communication languages [137, 195, 197, 198, 222, 275, 313, 367, 398, 481],
- nonconventional models of computing (in particular computing models based on interaction, ecorithms) [42, 59, 126, 130, 133, 169, 170, 210, 265, 270, 295, 333, 337, 341, 350, 396, 400, 420, 436, 437, 521, 544, 551, 554].

One can observe that the aforesaid challenging issues that appear in mentioned above many domains especially concern many fundamental problems of Artificial Intelligence (AI) [422] as well as Big Data [134, 224, 446], Ultra Large Systems (including CPSs), or Web of Things (W2T)) [1, 404, 445, 446]. They include:

1. Understanding CAS.
2. Symbol grounding problem (Harnad, 1990 [189]) and semantic pointers [126].
3. Computability by c-granules.
4. Asymptotical correctness of learning and real-life problems.
5. Concepts, agents, their ontologies and languages, societies of agents.
6. Modeling compound c-granules as networks of c-granules, e.g., actions or niches.

In following sections we introduce general descriptions of the above six problems.

3.1 Understanding CAS

There exists a large portion of literature on CAS. Here, we refer only to two selected books [158, 210] in order to illustrate the importance of IGrC methods in designing CAS. We will start with quoting some fragments from [158]:

> The common feature of these processes is that in each one a complex adaptive system acquires information about its' environment and its own interaction with that environment, identifying regularities in that information, condensing those regularities into a kind of "schema" or model, and acting in the real world on the basis of that schema. In each case. there are various competing schemata. and the results of the action in the real world feed back to influence the competition among those schemata. (p. 20)

> The laws of elementary particle physics are thought to be exact, universal, and immutable (apart from possible cosmological considerations), even though we scientists may approach them by successive approximations. By contrast, subjects like archaeology, linguistics, and natural history are concerned with individual empires, languages, and species, and at a more detailed level with individual artifacts, words, and organisms, including human beings like ourselves. In these subjects the laws are approximate; moreover, they deal with history and with the kind of evolution undergone by biological species or human languages or cultures. (p. 8)

> In much of today's research on complex adaptive systems, mathematics plays a very significant role, but in most cases it is not the kind of mathematics that has traditionally predominated in scientific theory. (p. 320)

> Discrete mathematics of the kind we have been discussing is often called rule- based. It is a natural kind of mathematics for digital computers and it is often applied to the simulation of complex adaptive systems composed of many individual adaptive agents, each of which is itself a complex adaptive system. Typically. the agents-such as organisms in an ecological community or individuals and businesses in an economy-are evolving schemata describing the behavior of other agents and how to react to it. In such cases. rule-based mathematics becomes agent-based mathematics. (p. 321)

Hence, Murray Gell-Mann, who in 1969 received the Nobel Prize in physics for his findings related to the theory of elementary particles, explicitly expressed in his book [158] that there is a need for new tools (based on, in the opinion of the author, on agent-based mathematics) to deal with CAS, such as the "jaguar" as referred to in the book's title. In this book, we propose to search for such an approach, based on IGrC.

John Holland, in [210] investigated CAS as systems exhibiting obvious internal boundaries, which divide each such system into a diverse array of semi-autonomous subsystems called agents. Each agent has a "program" that supervises interactions with other agents, and other parts of the agent's environment. CAS are treated as signal/boundary systems [210]. The CAS control is expressed by modifying signal/boundary hierarchies. In general, interactions are one of the basic concepts of this approach. We may distinguish four categories of interactions in signal/boundary systems: diversity, recirculation, niche, and co-evolution. A niche is a diverse array of agents that regularly exchange resources and depend on this exchange for their continued existence.

3.2 Symbol Grounding Problem and Semantic Pointers

The greatest concern when constructing models of cognitive systems has always been how to characterize the relationship between the states inside a given system and the objects in the external world which are supposed to be represented (cf. Harnad, 1990 [189]). In the book, we propose to deal with this problem by introducing c-granules. In our approach, links in c-granules are used to serve the role of semantic pointers, as introduced in [126]:

> A semantic pointer is a neural representation that carries partial semantic content and is composable into the representational structures needed to support complex cognition.
>
> [...] semantic pointers bear systematic relations to what they point to. For example, a neural representation at the top of the visual hierarchy would be a semantic pointer that points to the many lower-level visual properties of an image encoded in visual cortex. It points to these precisely because it does not explicitly contain representations of those properties, but it can be used to re-activate those lower-level representations through feedback connections.
>
> [...] The relation between a semantic pointer and what it points to can be usefully thought of as a kind of compression. Just as digital images can be compressed into small JPEG files, lower-level visual information can be compressed into semantic pointers. To identify the visual features captured by a JPEG, it must be decompressed. Nevertheless, we can manipulate the JPEG in some ways without fully decompressing it-by giving it to our friends, copying it, or perhaps flipping it upside down. However, to get back the visual details we must decompress or, in pointer parlance, "dereference" it. Dereferencing is thus part of the function of feedback connections in my earlier vision example. Crucially, if the same process is always used for compression, then similar uncompressed input will result in similar compressed representations.
>
> This is the sense in which semantic pointers are semantic. They can be directly compared to get an approximate sense of the similarity of the states from which they were compressed. So, they carry with them what we might call a "surface" semantics. To get at the "deep" semantics, we need to more directly compare the uncompressed states. Of course, much of the information lost through compression resides in the process used to produce a compressed representation. For this reason, to get at deep semantics we must be able to effectively run the system "backward" to flesh out what properties likely gave rise to that pointer. This is just another description of dereferencing. (p. 19).

Moreover, operations of aggregations performed on c-granules, which lead to the formation of new c-granules on the basis of the existing ones, enable us to deal with structural representations [126]:

> The second question above addresses what I have already noted is identified nearly universally as a hallmark of cognitive systems: the ability to manipulate structured representations. Whether or not we think internal representations themselves are structured (or, for that matter, if they even exist), we must face the undeniable ubiquity of behavior that looks like the manipulation of language-like representations. Consequently, any characterization of cognition will have to tell a story about how syntactic structure is encoded and manipulated by a cognitive system.
> [...] I tackle this question by showing how semantic pointers can be composed in a symbol-like manner to create complex representational structures. Specifically, I show how different semantic pointers can be bound together.

(p. 20) In IGrC, we assume that there are some physical objects inside agents, thanks to which their states are "measurable" (may be encoded), by means of information, e.g., represented by descriptors or their combinations, words, or expressions from a natural language. In a more general setting, one can treat the results of the encoding process, carried out by an agent, as special c-granules. The physical states of these c-granules are consciously perceived by an agent.

Partial information about the states of physical objects, which are found outside the agent's environment, can be perceived by an agent only implicitly, using the properties of interactions among physical objects. These interactions are transmitted to objects located inside an agent.

C-granules enable us to transmit the properties of interactions in the physical world to certain parts of an agent (other physical objects found inside an agent) and encode them by states, which can be interpreted by means of information, e.g., represented by descriptors or their boolean combinations, words, or expressions of a natural language. A further aggregation of such parts leads to the creation of expressions in the agent's (private or social) language.

3.3 Computability by C-granules

It is worth mentioning that the IGrC-based computational model, proposed in this book differs from the Turing computational model. The summary of the discussion devoted to this issue can be found at the end of the book (cf. Chap. 36). Here, we would like to present an outline of the comparison between computational models of IGrC, as proposed in this book, and computational models based on (partial) recursive functions, as proposed by Stephen Kleene [273]. In his definition, Kleene distinguishes some elementary computable functions and operations under which a set of partial recursive functions is closed [273]. In our case, the decision of what would be the notion of an elementary "computable" function depends on the agent's ability to construct such a function in the physical world.

In our approach the concept of computability is close to concept of ecorithm (cf. [521] and Sect. 14.4).

As another example, one can consider the process of computing a value of a complex function, using the quantum computing paradigm presented by Andrew Chi-Chih Yao, the Turing Award winner, during the 2014 Web Intelligence Congress (WIC 2014) panel discussion in Warsaw (cf. Fig. 3.1).

One can take another point of view and consider, e.g., real valued signals or videos as elementary functions. Another example is related to the project of a relational machine by Stanisław Ulam [514], in which he proposed using some relations as elementary entities, on which computations should be performed. He tried to implement the idea, using optical computing. The operations of aggregations on c-granules related to such complex objects are computationally admissible, provided that we can perform them in the physical world. This situation is illustrated in Fig. 3.2, which shows the computation of operation $\otimes$ by c-granules embedded in the physical world.

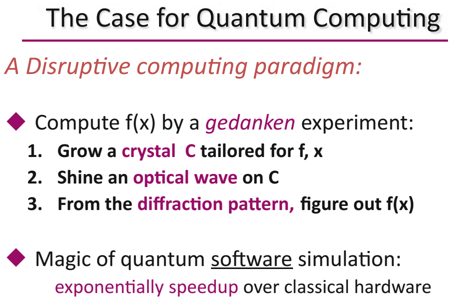


Fig. 3.1 The case of quantum computing

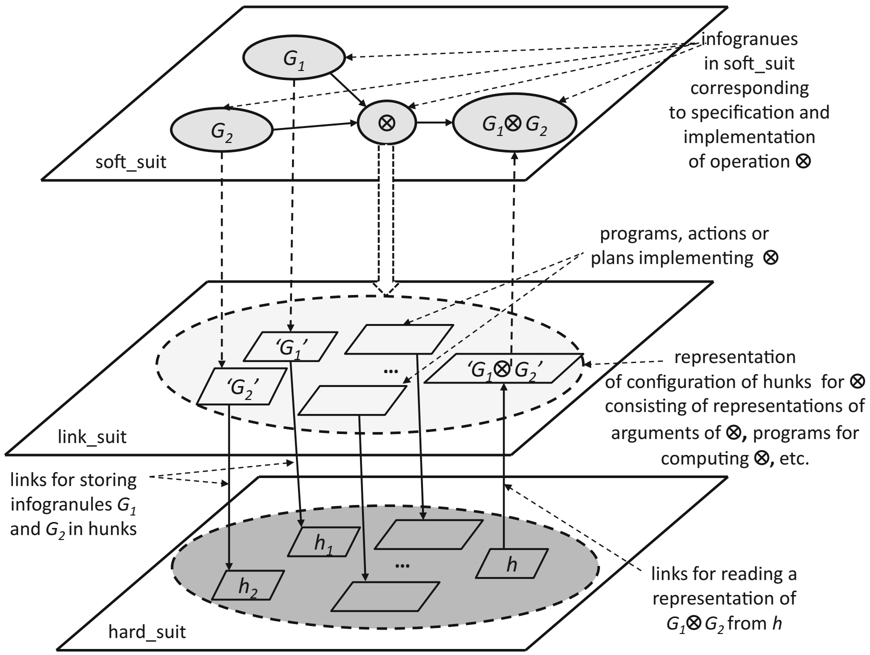


Fig. 3.2 Implementing aggregation in the physical world - "computability" of ⊗ function

In Fig. 3.2, there are three parts of a c-granule: soft_suit, link_suit, and hard_suit. The "computability" of the value of $\otimes(G_1, G_2)$ requires (at given moments of the agent's time) the following:

a. Gaining an access to a relevant configuration of physical objects (from the hard_suit) through interactions between particular physical objects from the hard_suit and particular objects from the soft_suit (through transmissions, using link_suit). These relevant configurations of the objects encode G_1, G_2.

b. Initiating, by means of links, relevant interactions in the hard_suit. These interactions are responsible for computing the value of the function $\otimes$ on representations of arguments G_1, G_2.
c. Encoding the value of $\otimes$ on arguments G_1, G_2 in the soft_suit. This is done by transmitting relevant properties of interactions from certain parts of the hard_suit to the soft_suit by means of links.

Hence, in our case, the "computability" depends on the possibility of performing computations in the physical world and on the cognitive faculty of agents. It is worth mentioning that when aggregating c-granules which correspond to the internal parts of an agent, a great role, with constraints on data tables (information systems [456]), is played by join operations. This is discussed later in the monograph. These operations enable us to develop searching methods for relevant c-granules on different levels of a c-granular hierarchy and interactions. Note that when initiating aggregating operations, relevant from the point of view of a given problem, the role of self-organization strategies [59, 110, 130, 136, 161, 184, 287, 311, 343, 365, 405, 420], based on c-granules or on (societies of) agents, are also very important.

3.4 Asymptotic Correctness of Learning and Real-Life Problems

At this point, it is worth citing some sentences from [521] by Valiant:

> The algorithms I discuss in this book are special. Unlike most algorithms, they can be run in environments unknown to the designer, and they learn by interacting with the environment how to act effectively in it. After sufficient interaction they will have expertise not provided by the designer, but extracted from the environment. I call these algorithms ecorithm. [...] ecorithms [...] derive their power by learning from whatever environment they inhabit, so as to be able to behave effectively in it. To understand these we need to understand computations in the Turing sense. But we also need to refine his definitions to capture the more particular phenomena of learning, adaptation, and evolution. [...] The phenomenon of generalization has been widely discussed by philosophers for millennia. It has been called the problem of induction. Turing did not attempt to capture all the connotations that the word computing may have had in his day. He sought only to uncover a phenomenon associated with that word that had fundamental reality independent of any word usage.
>
> [...] the right goal must be to find a mathematical definition of learning of a nature similar to Turing's notion of computation, rather than an informal notion like the Turing Test. [...] At the heart of my thesis here is a mathematical definition of learning. It is called the PAC or the probably approximately correct model of learning, and its main features are the following: The learning process is carried out by a concrete computation that takes a limited number of steps. Organisms cannot spend so long computing that they have no time for anything else or die before they finish. Also, the computation requires only a similarly limited number of interactions with the world during learning. Learning should enable organisms to categorize new information with at most a small error rate. Also, the definition has to acknowledge that induction is not logically fail-safe: If the world suddenly changes, then one should not expect or require good generalization into the future. [...] the idea of an ecorithm goes well beyond the idea of machine learning in its current, general usage. Within the study of ecorithms

> several additional notions beyond the learning algorithms themselves are included. First, there is the notion that it is important to specify what we expect a learning algorithm to be able to do before we can declare it to be successful. Second, using such a specification, we can then discuss problems that are not learnable-some environments will be so complex that it is impossible for any entity to cope. Third, there is the question of how broad a functionality one wants to have beyond generalization in the machine learning sense. To have intelligent behavior, for example, one needs at least a reasoning capability on top of learning. Finally, biological evolution must fit somehow into the study of coping mechanisms, but it is not clear exactly how, since traditional views of evolution do not exactly fit the machine learning paradigm. In studying ecorithms, we want to embrace all of these issues, and more. [...] In late 2008 Queen Elizabeth II asked a group of academics why the world financial crisis had not been predicted. She was not the only one asking this question. Was the crisis inherently unpredictable in some sense, or was the failure due to some gross negligence? After the crisis a substantial amount of public discussion pertained to this question. Is there a rational way of predicting rare events? Why do humans have so many intellectual frailties and behave as irrationally as they do? Why are humans subject so easily to deception and self-deception? Why do humans systematically delude themselves into thinking that they are good predictors of future events even if they are not? The general disappointment that the world financial crisis had not been better predicted was not based entirely on naive illusion. It was based on the well-justified high regard we have for our predictive abilities, and so it would be clearly to our advantage to identify why they failed.
>
> It may be that the world was changing in such a random fashion that the past did not even implicitly contain reliable information about the future. Or perhaps the past did indeed contain this information, but that it was somehow so complex that it was not practically feasible to dig it out. A third case is that prediction was indeed feasible, but the wrong algorithm or the wrong data had been used. [...] How can entities cope with what they do not fully understand? The simplest living organisms have had to face this problem from the beginnings of life. With limited mechanisms they had to survive in a complex world and to reproduce. Every evolving species has faced a similar problem, as do individual humans going through their daily lives. I shall argue that solutions to these problems have to be sought within the framework of learning algorithms, since this is the mechanism by which life extracts information from its environment.

The brilliant idea of Probably Approximately Correct (PAC) algorithms, introduced by Valiant [521], enables us to better understand the nature of learning. The approach ensures (under the assumption that the so-called set of hypotheses has a finite combinatorial Vapnik-Chervonenkis dimension [7, 527]) the asymptotic approximate correctness, which means that intuitively speaking, there will be a very low chance that an induced data model, consistent with a sufficiently large number of examples, will be of not acceptable quality when used for unknown so far cases is very low. However, in real-life, any organism or agent should be able to make right decisions without having a chance to wait for such a “sufficiently large” samples of examples. In the actual “technological reality” and in a given real-life situation, they should be able to continuously judge which tasks should be selected and performed to achieve their prioritized goals. Note also that in many cases the lack of any decision, is worse than even a poor decision.

A discovery and/or construction of new sensors and actuators (for interacting with the physical world external to agents) as well as aggregating operations performed on c-granules (computational building blocks for cognition), which lead to the perception (comprehension) of the physical world, become crucial elements of the process of judgment and action selection. Based on the perception of the physical world, agents select and execute relevant actions.

3.5 Concepts, Agents, Ontologies, Languages, and Societies of Agents

Let us note that agents as well as societies of agents can be treated as c-granules too. In our model, the ability to aggregate the existing c-granules to more compound ones depends on the physical ability of an agent. Ontologies of agents or their societies are complex. This will be discussed later in the book (cf. e.g., Chap. 33). One can better understand the complexity of such a task, which concerns the research on cognitive systems, by means of an analysis of systems such as SOAR or ACT-R [6, 295]. They aim at constructing a general cognitive architecture for the development of systems, which exhibit intelligent behavior. The research on the development of such systems, initiated many years ago, is still very active and engages different groups of researchers [2, 3].

It is worth mentioning that the process of modeling cognitive architecture and/or intelligent behavior is a highly non-trivial task. In particular, this requires the ability to design very complex ontologies for many domains, with thousands of concepts and relationships among them. An analogous situation occurs during IGrC modeling based on WisTech. The approach presented in this book is based on FPW (cf. Chap. 32) postulates, which in itself are very complex. They are expressed in a natural language, using many different domains of knowledge (e.g., psychology, phenomenology, logic, management, sociology, economy, and computer science). Most of the concepts (and relationships among these concepts) from the FPW ontology are vague and complex. Moreover, they are used in an open and dynamic environment. Hence, empirical interactive experiments often provide the basis for specifying the meaning of particular concepts which evolves with time.

3.6 Modeling Compound C-granules as Networks of C-granules

In the approach proposed in this book, actions or plans are modeled by c-granules. The results of actions performed by c-granules in the physical world may not lead to the expected results due to unpredictable interactions in the environment. Hence, one can design only a scheme presented in Fig. 3.3, where the properties γ of the action ac are only expected.

However, the condition α may guarantee that in "typical situations," one can expect the results γ of an action ac. However, the specification α can define the features of

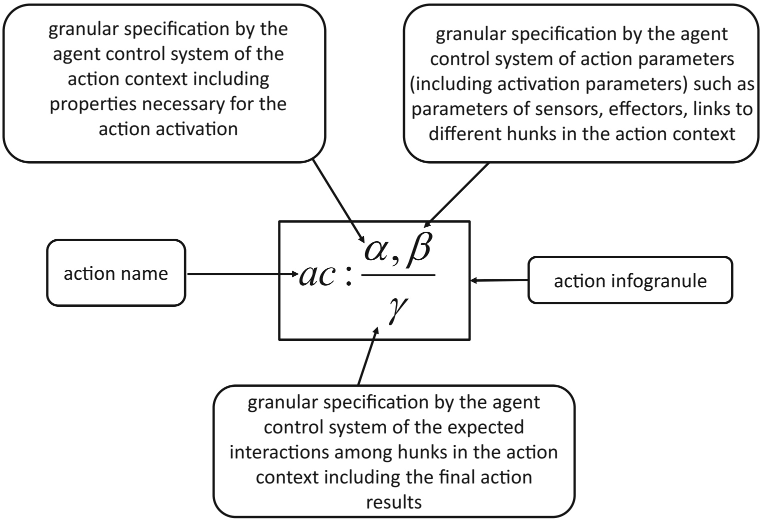


Fig. 3.3 The expected results of the action *ac*

resources that are necessary for initiating and running the action ac. Analogously, the features displayed by the new states of resources, expected, after performing of the action ac, can be described by the specification γ. Let us also note that due to some degree of uncertainty, usually only some arguments *for* and *against* the satisfiability of conditions α, γ can be received. Hence, the techniques for resolving conflicts between such arguments for different actions should be developed. This issue will be discussed in more details further in the book.

Let us also note that a general scheme of interaction rules can be represented in an analogous way to actions (cf. Fig. 3.4).

One may refer here to the concept of niches from [210], which has been already mentioned in the book:

> The niche, then, is made up of physical and virtual boundaries that determine the limits of these interactions. [...]. The invisible boundaries that define niches are a complex topic, still only partly understood. (p. 2)
>
> The most interesting signal/boundary systems are composed of agents – bounded subsystems capable of interior processing. (p. 8)
>
> [...] signal/boundary systems evolve continually. So the "rules" must provide for origins of new signals and boundaries and their selective modification. (p. 10)
>
> The diversity of highly specialized species plays an important part in the conservation of resources by recirculation. (p. 13)
>
> As with diversity, recirculation is a common feature of systems exhibiting signal/boundary interactions. (p. 15)
>
> How do the signals that make finely tuned niches possible come into being? [...] Because there are niches within niches, web-like hierarchies result. [...] The resulting niches offer new possibilities for interaction and increasing diversity. [...] Niches and hierarchies are common to all signal/boundary systems. [...] there is even less study of mechanisms for the formation

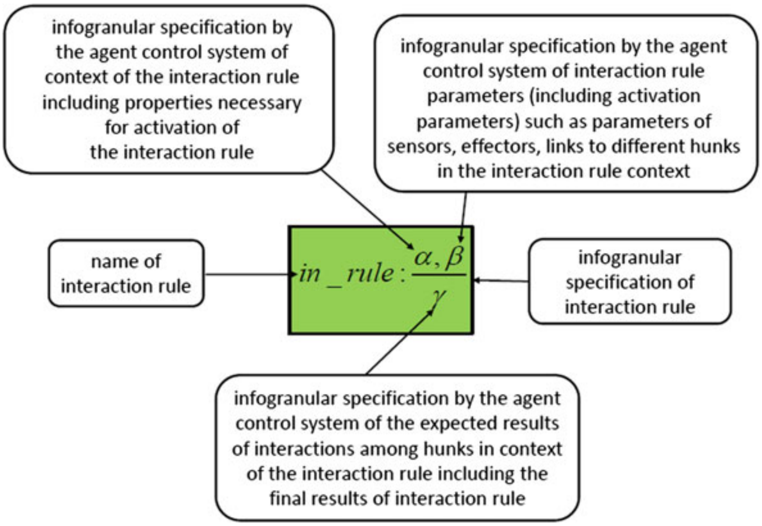


Fig. 3.4 Interaction rule scheme

> of hierarchies—mechanisms that would explain the pervasiveness of hierarchies in natural systems. That is due in part to the extreme difficulty of the mathematics of such processes. (p. 16)
>
> studying the formation of boundaries within networks offers a broad approach to all signal/boundary interactions. (p. 17)
>
> In coevolution, each new species offers new possibilities for interaction. These interactions can become increasingly specialized [...] Combining this progressive specialization with the "multiplier effect" mentioned above in the subsection on recirculation gives a tantalizing comparison between the richness of ecosystems and the richness of factory systems. Again boundaries and enclosures are crucial. [...] As has already been mentioned, the progressive specialization of interactions has an analog in economics. (p. 19)
>
> Every time we examine a signal/boundary system closely, we see coevolution as a pervasive feature. [...] The mechanisms and interactions falling under the broad categories diversity, recirculation, niche and hierarchy, and coevolution are central to understanding a wide range of large-scale and small-scale signal/boundary systems. (p. 20).

One of the challenges is to develop methods based on c-granules in order to induce relevant hierarchies of c-granules with hierarchical niches [210], as well as reasoning schemes about these hierarchies. IGrC can be also used as a basis for modeling different categories of interactions in signal/boundary systems, such as those related to diversity, recirculation, niche, and co-evolution [210]. One of the approach which could be used as an initial step for dealing with niches is related to decomposition of data tables used in discover concurrent models (see e.g., [28, 361, 452, 461, 462]).

3.7 Adaptive Judgment

For centuries, philosophers, psychologists, logicians, and recently also computer scientists, have been interested in different aspects of judgment [37, 76, 116, 181, 188, 276, 328, 333, 397, 427, 504]. One can ask about the definition of judgment. For example, Per Martin-Löf presents the following understanding of the concept of judgment [427] (p. 4):

> [...] I think that the most natural explanation is to say that the meaning of a judgement is fixed by laying down what it is that you must know in order to have the right to make the judgement in question. Or, in another formulation, which is the same in substance, though, a judgement is defined simply by what knowledge it embodies: a judgement is a piece of knowledge, and you have to clarify what knowledge.

In [35, 183], the following major features of practical judgment are distinguished:

1. **Holistic nature of judgment**. Different human attributes related to cognition, ethics, morality, emotion, practice, etc. are all integrated in judgments and the "right" judgment constitutes a balance between these attributes.
2. **Contextual nature of judgment**. For example, the final judgment is an aggregation of judgments obtained on the basis of some specific contexts, extracted from data or domain knowledge.
3. **Denotational nature of judgment**. Judgments ensure the connection of an agent with the outer world by determining actions or plans and predicting their results (due to the level of uncertainty, these predictions may not be correct); hence, practical judgment can only be learnt from experience and actual practice.
4. **Defeasible (or fallible) nature of judgment**. Practical judgments are used for such purposes as solving a crisis, meeting the needs of a customer, satisfying a deadline; they can be changed if things are not working out as predicted; they will not be effective every time, but they will be more or less satisfactory. Moreover, deeper understanding or more detailed information might lead to the adaptation of judgment.
5. **"Problem-identification" nature of judgment**. Here, several tasks arise from judgment (especially for an autonomous practitioner):

    i. for real-life projects, a major task of a practitioner is to identify what might constitute a problem or problems in a given situation,
    ii. the problem may not have a unique solution,
    iii. only partial data may be available and some procedures may be unclear.

6. **Social nature of judgment**. Judgments are socially shaped in line with the norms of a society, in which they are made; practical judgment should obey, e.g., social, cultural, and political norms of a given society, which evolve with time.

It is worthwhile mentioning that recently we observe a growing interest in research on judgment and interactions on interdisciplinary level, including computer science, AI, cognitive science[1] as well as logic and philosophy [45, 47, 67, 114, 180–182, 207, 418].

In this book, we focus on the concept of adaptive judgment. In this way, we emphasize the role of adaptation in judgment. The role of adaptive judgment is crucial in controlling computations, thus enabling (societies of) agents to achieve their goals according to their adaptive hierarchy of needs. Adaptive judgment is also fundamental in searching for relevant hierarchical c-granules. This corresponds, e.g., to adaptive strategies for inducing hierarchies of satisfiability relations, used in searching for the computational building blocks of cognition (e.g., patterns for inducing approximations of complex vagues concepts, used to initiate actions (plans)). The reasoning about c-granules and computations performed on them is not only based on deduction, but also on induction and abduction and sometimes, it may even go beyond these paradigms of reasoning. This is well characterized in [320]:

> In the formal act of judgment we can distinguish several elements or stages, though it would not be possible to separate all of them: (1) The apprehension of the thing or object about which the judgment is made; (2) the separation or separate grasp of the two terms—the two aspects or phases of the thing which are to be compared; (3) their juxtaposition; (4) the perception of the agreement or disagreement of the juxtaposed concepts; and (5) the concomitant awareness that the mental juxtaposition of ideas corresponds to the objective reality.

Hence, adaptive judgment is also responsible for reasoning about the agent's understanding of the perceived situation in the physical world, i.e., in a sense, it is connected with the "correctness" of the perception process, which is relative to the agent's goals. This involves, in particular, controlling the process of judgment and shifting the agent's attention to issues that are important for her/him based on the selection of (spatiotemporal) "windows," through which it is possible to perceive situations in the physical world. The results of the perception process are used by an agent for the selection of right actions or plans to achieve his/her goals. Note that this judgment concerns two different worlds: the external physical world of an agent and her/his mental world. Moreover, due to a certain level of uncertainty, an agent should take into consideration the risk in accepting the already constructed (induced) views of a given situation. This leads to the cost/benefit analysis, discussed later in the book.

At this point, it is worth referring to a discussion from [245] concerning the Aristotle's understanding of perception. Aristotle, the father of formal logic and its basic concepts, like deduction and induction, emphasized the importance of attention in perception. He investigated relationships between various concepts, such as thinking, imagination, and psyche. In [475], for example, one can find the following statement:

> Thinking is different from perceiving and is held to be in part imagination, in part judgment: we must therefore first mark off the sphere of imagination and then speak of judgment.

If we select an interpreter (i.e., soul (psyche)) as the most important concept, we can illustrate this statement as it is shown in Fig. 3.5. Let us recall that Aristotle

[1] http://plato.stanford.edu/archives/fall2008/entries/cognitive-science/#Bib.

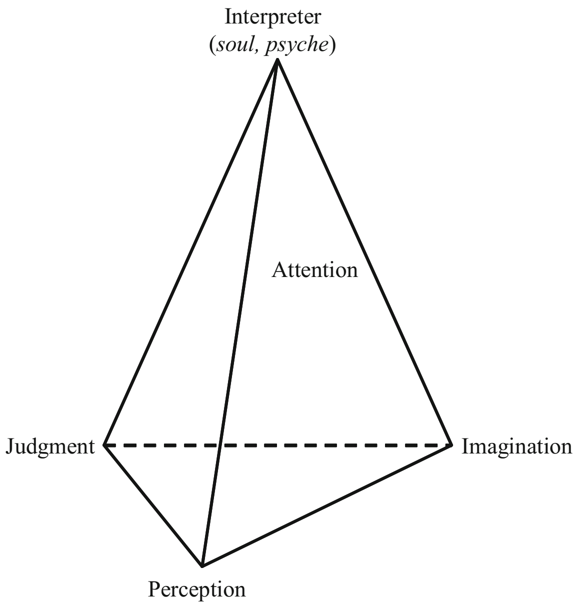


Fig. 3.5 Relationships between imagination, judgment, perception and psyche

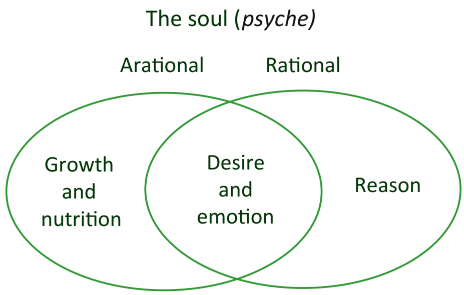


Fig. 3.6 Parts of the soul (psyche) [294] (p. 78)

provides also an analysis of the concept of soul (psyche). In the simplest approach he divides it into an arational part, and a rational part. For mode details. see Fig. 3.6 from [294] (p. 78).

Aristotle emphasized the importance of links between the symbols of spoken words (which are the basis for judgment and syllogistics), mental experiences represented in our imagination, and things of which our experiences are the images.

In [122], one can find the following fragment:

> Spoken words are the symbols (symbola) of mental experience (pathemata) and written words are the symbols of spoken words. Just as all men have not the same writing, so all men have not the same speech sounds, but the mental experiences, which these directly symbolize, are the same for all, as also are those things (pragmata) of which our experiences are the images (homoiomata).

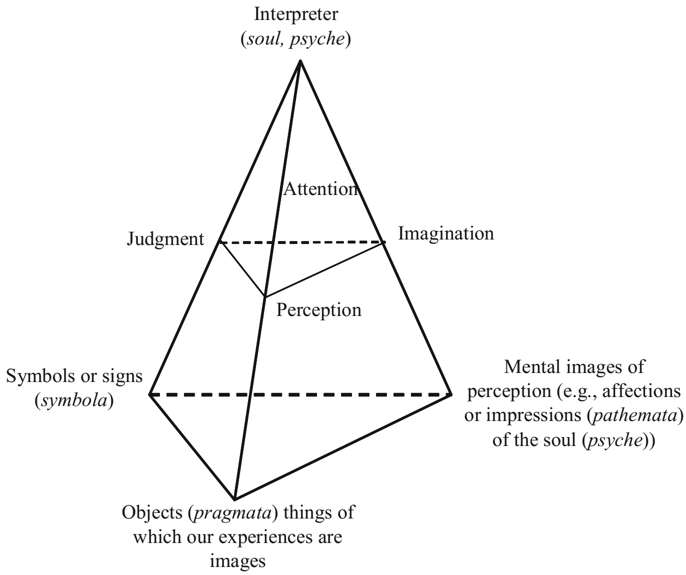


Fig. 3.7 Aristotle tetrahedron

In some sense, this is an extension of the idea presented in Fig. 3.5 which can now be represented in the *Aristotle Tetrahedron* (cf. Fig. 3.7), [245].

The Aristotle Tetrahedron can be treated as a metaphorical starting point for the constructions of computational models for WisTech based on IGrC. However, the Aristotle Tetrahedron is a very complex concept even its sides and walls represent very complex relationships in themselves. For example, one of the greatest challenges is to learn how to connect words, phrases, and sentences with the perception of objects and events in the world [350]. Let us also notice that modern linguistics is mainly based on the exploration of relationships, represented by the projections of the tetrahedron. This is the main relation studied in a well known book, [371]. The corresponding illustration is presented in Fig. 3.8.

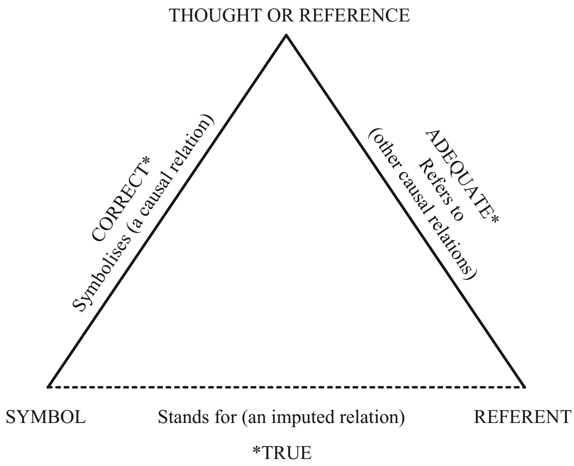
At this point, we should also cite adequate fragments by Valiant[2]:

> A fundamental question for artificial intelligence is to characterize the computational building blocks that are necessary for cognition.

Complex networks of c-granules are used for modeling higher-order c-granules. These network-based c-granules help perform adaptive judgment on higher levels of c-granular hierarchies. C-granules also play a crucial role in the research on evolving communication languages of agents or their societies [137, 195, 197, 198, 222, 275, 313, 367, 398, 481]. Agents may contribute to the formation of societies of agents. This makes it possible to produce relevant emergent patterns of social behavior, which enables the whole society, as well its individual members, to fulfill their needs.

[2] http://people.seas.harvard.edu/~valiant/researchinterests.htm.

Fig. 3.8 Semiotic triangle



Communication languages of the societies provide tools for developing cooperation strategies, e.g., strategies based on self-organization [59, 110, 130, 136, 161, 184, 287, 311, 343, 365, 405, 420]. Communication languages are continuously evolving compound c-granules which enable complex systems composed of the societies of agents, to construct steering tools, based on the cost/benefit analysis, so as to controll the behavior of the whole societies and their members. The role of c-granules in defining agent's languages is very well-defined in [171] (p. 201):

> The domain of language, like those of vision and attention, are a collection of networks specialized for the execution of quite specific functions.

The crucial role of judgment, related to the cooperation among and within the societies of agents, is to improve, as much as possible, the fulfillment of the needs, shared by the whole societies of agents, as well as the needs of their individual members. At the same time, the process of judgment should guarantee the achievement of goals, keeping the risk of failure as low as possible.

3.8 Wisdom

This book is, in a sense, a step towards the implementation of the WisTech program. First, we would like to present some comments on the concept of wisdom. In [183] (pp. 217–219), wisdom is characterized as something more than just knowledge and understanding. The extra dimensions that wisdom is supposed to add to cognitive knowledge and understanding are usually non-cognitive qualities associated with experience and practice, such as judgment grounded in experience. For instance, "The Concise Oxford Dictionary" views wisdom as

> [...] possession of experience and knowledge together with the power of applying them critically or practically.

Webster's "New World College Dictionary" defines it as the:

> [...] power of judging rightly and following the soundest course of action, based on knowledge, experience, understanding, etc. [...] the wisdom of practice takes on diverse moral features reflected in the different levels of practice.

Another view from [484] (p. 152) indicates that:

> [...] wisdom is not just about maximizing one's own or someone else's self interest, but about balancing various self-interests (intrapersonal) with the interests of others (interpersonal) and of other aspects of the context in which one lives (extrapersonal), such as one's city or country or environment or even God. Wisdom also involves creativity, in that the wise solution to a problem may be far from obvious.

Finally, let us note that practical wisdom is understood as the ability to make informed, rational judgments without recourse to a formal decision procedure [294] (p. 99).[3]

One may find the following explanation of practical wisdom by Aristotle[4]:

> Practical wisdom (Greek: phronesis; sometimes translated 'prudence'), says Aristotle, is 'a true and reasoned state of capacity to act with regard to the things that are good or bad for man' Nicomachean Ethics VI.5). So while practical wisdom involves knowledge of what is good or bad, it is not merely theoretical knowledge, but a capacity to act on such knowledge as well. This capacity requires
>
> 1. *a general conception of what is good or bad, which Aristotle relates to the conditions for human flourishing;*
> 2. *the ability to perceive, in light of that general conception, what is required in terms of feeling, choice, and action in a particular situation;*
> 3. *the ability to deliberate well; and*
> 4. *the ability to act on that deliberation.*

Aristotle's theory makes practical wisdom very demanding. The type of insight into the good that is needed and the relation between practical wisdom and virtues of character are both complex. Practical wisdom cannot be taught, but requires experience of life and virtue. According to Aristotle, only a person who is good knows what is good.

---

[3] http://cw.routledge.com/textbooks/alevelphilosophy/data/A2/Moral/PracticalWisdom.pdf.

[4] http://www.alevelphilosophy.co.uk/.

Chapter 4
Main Overview of Results

I think the next [21st] century will be the ***century of complexity.***
We have already discovered the basic laws that govern matter and understand all the normal situations.
We don't know ***how the laws fit together,***
and ***what happens under extreme conditions.***
But I expect we will find a complete unified theory sometime this century.
There is no limit to the complexity that we can build using those basic laws.

Stephen Hawking (Stephen Hawking quoted in an interview on January 23, 2000 in the San Jose Mercury News; cf. http://www.mercurycenter.com/resources/search, https://arxiv.org/pdf/1303.3855).

[Answer to question: Some say that while the twentieth century was the century of physics, we are now entering the century of biology. What do you think of this?]

A deeper analysis shows that the motivation behind the research, outlined in Chap. 1 is related to conducting a search for the guidelines, which will ensure a better understanding of the answers to the following questions:

1. How to *comprehend*, describe, represent, model, and test such phenomena as:

   i. Thoughts of *intelligent beings* (including *human thoughts*)?
   ii. *Thought process* that processes the thoughts of intelligent beings (including the thought-communicating processes) that and are used by them to control (at a level of efficiency satisfactory to these beings) interactions between these beings and the surrounding environment?
   iii. *Thought process mechanisms* help to solve difficult problems, particularly mechanisms of effective cooperation, including mechanisms of effective

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_4

communication between the societies of intelligent beings, which enable them to cooperate when handling difficult problems?

2. What are the main reasons behind the great degree of faultiness of processes, the aim of which is to help people effectively solve practical problems, with a particular emphasis placed on a wide gap between the theory and practice in engineering processes related to complex systems?
3. What computational models should be proposed when it comes to the simulation of behaviors, displayed by intelligent beings in interaction with the surrounding environment?
4. What changes should be implemented in prevalent computational models, based on the paradigms established by the Turing machines, in order to increase the resemblance of these computations to thought processes that could simulate the behaviors of intelligent beings in interaction with the surrounding environment?

The aim of the book is to indicate potential directions for the future research, that could bring us closer to understanding the above-listed questions as well as highlight potential "road-signs," which could help in the search for relevant answers.

The nature of the above-listed proposals is limited to the description of IGrC, based on WisTech. A c-granule is one of the key terms of WisTech. **In our approach, they can represent thoughts of an intelligent being.**

The **interaction-related granular computations, carried out on the basis of WisTech, constitute the essence of the approach presented in this book, which should be adopted in order to better understand thought processes**. In our approach, these processes are conducted by the **societies of cooperating/competing c-granules and by the societies of agents who have control over these c-granules.**

In simplified terms, the nature of WisTech may be limited to a metaphorical Wisdom Equation, which is described more thoroughly in Chap. 29. Intuitively speaking, the equation assumes that wisdom (a society of c-granules and societies of agents, who have control over these c-granules) consists of three components, which are aggregated together: **interaction, adaptive judgment, and knowledge**. In some sense, this constitutes an analogy to a political system, based on the principle of separation and aggregation of powers: executive power (interactions), judicial power (*adaptive judgment*), and legislative power (knowledge).[1]

The WisTech concept approximation may be (approximately) presented using this metaphorical equation. For example, the intuitions behind the structural architecture of a c-granule may be—approximately—perceived as a concept consisting of three components: hard_suit (interactions), link_suit (adaptive judgment), and soft_suit (knowledge) (cf. Chap. 29).

Metaphorically speaking, the Wisdom Equation ensures the creation of the first, very rough, approximation of a general WisTech concept. Second approximation of this type which explains the meaning of basic WisTech concepts in a more accurate way, is seen in FPW (*Framework Postulates for WisTech*). These postulates are presented in Chap. 32. The following parts of the book, Chaps. 33–35, try to explain the

[1] http://en.wikipedia.org/wiki/Separation_of_powers.

FPW postulates in more details, particularly in reference to AI, efficiency management, and engineering related to complex systems.

A more accurate clarification of FPW shows a peculiar role of the so-called *granulations*, performed as part of WisTech, which are the processes that deal with the construction, aggregation, decomposition, processing, communication, and archiving of c-granules.

In the model of interactive computations proposed in this book, c-granules are used by the agent's control and lead to a systematic development and adaptation of the hierarchy of needs, which are also contained in these c-granules. The hierarchy of needs makes it possible to prioritize the needs of an agent. Other c-granules of an agent deal with the construction and adjustment of interaction plans. An agents is driven towards the construction and fulfillment of interaction plans that are as effective as possible. The book proposes an outline of a **computational model, the purpose of which is to manage the effectiveness of the agent's interaction plans**. The model in question is presented in Chap. 33. The nature of the concept of this model is thoroughly explained in Sect. 33.5, entitled "Engine for Efficiency Management" (EEM).

The computational models proposed in this book constitute a generalization of methodologies, used in the field of AI. Within this context, language may be used as a natural environment for the hybridization and comparison of a variety of AI technologies.

It should be underlined that the main outcome of the research conducted as part of this book are conclusions based on the analysis of many CSE projects. The results are motivated by the PC-TPGP thesis. A more detailed summary of the main conclusions (related to CSE) from the research can be found in Sect. 36.1.

Below, we list the main research results with links to appropriate chapters of the book:

1. Key common causes of discrepancies between the theory and practice of CSE projects. (cf. Chaps. 12 and 13).
2. Proposals of computational models (cf. Chap. 33), based on interactive granular-IGrC within the WisTech framework, designed to simulate the key functionalities of a wise teamwork (human beings and artificial agents) in only partially known, dynamically changing environments, with a particular focus on the performance control with regard to such activities as:

    2.1. Decision-making/action-taking processes carried out by agents, based on the satisfiability of concepts, and encompassing the following aspects of these processes, leading to discovery of interaction rules:

        2.1.1. Identification of complex and time-changing concepts important to agents (cf. Chap. 33).

        2.1.2. Adaptive learning of approximations of concepts at the level of details best adapted to the needs of agents (cf. Chap. 33).

        2.1.3. Discovery of concepts that are currently needed for an agent (cf. Chap. 33).

2.2. Cooperation or competition across societies of agents (human beings and artificial agents) in such actions as:
   2.2.1. Creating new technological solutions for AI application (cf. Sect. 35.2).
   2.2.2. Proposing applications of risk/efficiency management for life-critical systems (e.g., fire and rescue operations—on the basis of experience from the ICRA project (cf. Sect. 35.3)).

3. Proposals of techniques for reasoning (adaptive judgment) about the properties of computations carried out within computational models proposed in the book, with a particular emphasis on applications such as CSE project management (cf. Chap. 35), creation of applied AI techniques, and substantial risk suppression (cf. Chap. 35).
4. Some proposals for the use of the previously developed computational models and models of reasoning about the properties of these computations to:
   4.1. Improve the identification and explanation of the TPGP causes (cf. Chap. 35).
   4.2. Reduce the gap in TPGP (cf. Chap. 35).

From a technical point of view, the main results are focused on the development of a relevant computational model for WisTech, based on IGrC.

It is worth noting that the nature and ontology of computational models presented in this book essentially differs from the nature and ontology of computation models expressed in the language of modern mathematics at the turn of the century (like Turing computations expressed in the language of a set theory). In the book, we discuss many aspect of the differences between the models of computations based on a Turing machine and models of computations based on IGrC (cf. Chap. 33). The summary of the main differences is presented in tables, included in Sect. 36.1.

Chapter 5
Guide to the Contents of the Book

The present book consists of the following parts:

Part I—Introduction: Research Motivations, Approaches, Challenges, and Overview of Results

Chapters 1–5 introduce the motivation for the research. It also contains a summary of the most important results and may be treated as a guide to the remaining parts of the book.

Part II—CSE: Rudiments

Chapters 6–8 contain the unified terminology that is used in the book to refer to all complex systems and CSE. These chapters emphasize the variety of complex systems and their possible applications, as described in Part IV– CSE: Case Studies.

Part III—The Theory-Practice Gap Problem (TPGP)

Chapters 9–15 contain the clarification of a problem, related to the negative consequences of a gap between the theory and practice of CSE. The starting point, adopted here, is the explanation of the above-mentioned phenomenon on the basis of software engineering. Next, it is shown that similar phenomena may also be spotted in other areas of CSE.

We refer to this situation as the "CSE Crisis." In this part of the book, the most significant causes of the phenomena related to the "CSE Crisis" are thoroughly discussed. A particular emphasis is put on the causes that seem particularly important from the point of view of practical experiences, gained by the author of the book. At this point, the author also makes an attempt to identify the common precause of these problems. This precause is presented in Chap. 13

Part IV—CSE: Case Studies

Chapters 16–21 present (in a very condensed way) practical conclusions drawn from the implementation of selected projects, carried out under the direction of the author of this book. They include the following initiatives: POLTAX (first decade

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_5

of the development), Merix, Excavio, and AlgoTradix. Moreover, the book also presents the conclusions drawn from the UAV project (especially important for the results included in the book), the implementation of which was closely observed by the author. The author also had a chance to participate in substantial discussions with the key project participants, involved in the scientific part of the initiative (particularly with regard to the applications of the AI technology within the project).

Chapter 21 covers some important conclusions of the case studies from the point of view of *science-friendly languages* for interaction rules and/or CSE principles.

Part V—CSE as a Metaphor of Mind "Computations"

Chapters 22–26 are related to the metaphor, according to which the human mind can be treated as a society of agents. The thought process—in this metaphor—is a result of the cooperation and/or competition among such groups of agents. The metaphor was first used by Ervin Minsky in his book "The Society of Mind" [340]. On this basis, we proposed, in Part V, to treat CSE projects as complex models with computations performed by a metaphorical mind. However, we should note that this is not a solution to the problem of modeling thought processes, but only a guideline, which indicates that more work should be undertaken in order to reach the goal.

In this part of the book, we also discuss the concept of the language of communication used between agents, which is one of the key conditions for an effective cooperation.

At the same time, we lean towards the perception of the meaning of language as a language game, as understood by Wittgenstein.

The role of the language is exposed in this part of the book. Language is interpreted as a tool that is used to communicate and represent thoughts. However, the linguistic expressions are not necessarily treated as thoughts. At this point, we refer to the thesis formulated by one of the co-creators of intuitionism, Brouwer, and many other famous mathematicians. They claimed that the process of thinking, i.e., creating a thought (particularly, thoughts of mathematicians), is an activity which does not involve any language, that is, it is languageless. When discussing intuitionism, we should recall the importance of the concept of "judgment" within this philosophy.

Part VI—WisTech Approach to Models of Mind: Preliminaries

Chapters 27–30 present some preliminaries of IGrC in the context of agent architecture (cf. Chap. 27), selected distinguishing properties of WisTech IGrC models (cf. Chap. 28), and an interpretation of the Wisdom Equation as a component of a c-granule (cf. Chap. 29).

Part VI covers a discussion on relationships between concept of c-granules and concepts of Socratic dialogues, Tarski truth, and semantic games of Wittgenstein (cf. Chap. 30).

Part VII—Framework Postulates for Ontology of WisTech Models

Chapters 31–32 present an approach towards modeling interaction-related granular computations, based on WisTech. In order to perform the above-mentioned task, the Framework Postulates for WisTech (FPW) have to be implemented. These postulates can be treated as an elaboration on a simplified understanding of WisTech, expressed in the metaphorical Wisdom Equation. At this point, the author clarifies the intuitions behind such basic WisTech concepts as c-granules, interactions, agent's and c-granule's perception, adaptive judgment, and interaction plans.

Possible clues related to the physical implementation of the previously proposed interactive computational models are being considered. The high complexity of FPW is emphasized, along with the complexity of modeling interaction-related granular computations, based on WisTech.

Part VIII—WisTech Introduction to Efficient Acting, Learning, and CSE Project Implementation

Chapters 33–36 contain an outline of the proposed computational models, based on the FPW postulates, all of which are formulated in the previous part of the book. In these chapters, the IGrC model is elaborated on, especially within the scope of its potential use in enhancing the effectiveness management (cf. Chap. 33), as understood in WisTech.

At the same time, effectiveness management is treated as a critical activity, required in CSE and AI projects.

Effectiveness management in CSE and AI projects is an area which offers a wide range of possible applications for interactive granular models (including efficiency/effectiveness management, as understood in WisTech) for the purpose of implementing CSE and AI projects.

On the basis of the above-mentioned observation, the author proposes an approach towards machine learning, along with cooperation mechanisms used by a society of agents, in order to tackle the aforesaid problems.

These proposals are presented within the scope of the potential use of WisTech, in order to get a deeper understanding of TPGP and processes that occur as part of the CSE project implementation. This particularly concerns the projects, which may ultimately lead to the creation of some innovative AI technologies (cf. Chap. 35). In some sense, the message included in the approach proposed by the author may be reduced to a statement that CSE projects constitute a particular case of IGrC, the aim of which is to fulfill FPW postulates. Thus, this leads to an acquisition of both a more in-depth knowledge, related to the control of such processes, and better conclusion-forming tools, related to the properties of such computations. These elements should help diminish the negative consequences of TPGP for CSE projects.

The conclusions presented in the book are illustrated with potential practical applications with regard to the previously presented computational models, within the scope of risk management in rescue operations (on the basis of the ICRA[1] [286] project). Additionally, the author presents various options of using the proposed efficiency management model (as understood in WisTech) in order to gain a deeper understanding of the causes which lead to negative consequences of TPGP and to reduce the risk of these consequences.

[1] http://icra-project.org/?node=10.

The final chapter (cf. Chap. 36) contains a summary of the book, along with the prospects for the future research. Moreover, substantial differences between the Turing computational models and the WisTech computational models are thoroughly presented in a number of tables.

Part II
CSE: Rudiments

Chapter 6
The Concept of Complex System

6.1 Complex System

Going beyond mythology as a way to explaining the world the ancient Greek philosophers introduced science based on reason, evidence, and ontology linked to verifiable experiments.

They used many intellectual tools, among which one of the most important was the concept of a “system.” The concept was used in numerous contexts, e.g., Plato’s “Philebus”, Aristotle’s “Politics”, Euclid’s “Elements.” Generally, it appeared in the context of other concepts, such as unity, totality, whole, parts, relationships, and principles of change.

Undoubtedly, the concept of a system is important for all branches of science. Its meaning evolved over the course of time. In many cases, the history of science can be treated as a history of development of some systems.

There are many great scientists who added essential value to our contemporary understanding of the concept of a system, including René Descartes, Charles Robert Darwin, Nicolas Carnot, Robert Brown, Rudolf Clausius, Henri Poincaré, Ludwig von Bertalanffy, Marian Smoluchowski, Albert Einstein, Andrey Markov, George Birkhoff, Norbert Wiener, Ross Ashby, John von Neumann, Claude E. Shannon, Andrey Kolmogorov, Edward Lorenz, Murray Gell-Mann.

After more than two thousand years, the concept of a system is still one of the key intellectual tools used for the development of our civilization. Today, many top scientists believe that the understanding of the concept of complex system constitutes the most critical challenge for the further development of science and engineering.

To illustrate this point of view, let us consider the mission statement for the very well-known Santa Fe Institute. The mission is following[1]:

[1]https://www.santafe.edu/about.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_6

> **Our Mission**
> Searching for Order in the Complexity of Evolving Worlds
> Our researchers endeavor to understand and unify the underlying, shared patterns in complex physical, biological, social, cultural, technological, and even possible astrobiological worlds. Our global research network of scholars spans borders, departments, and disciplines, unifying curious minds steeped in rigorous logical, mathematical, and computational reasoning. As we reveal the unseen mechanisms and processes that shape these evolving worlds, we seek to use this understanding to promote the well-being of humankind and of life on earth.
>
> **What Is Complexity?**
> Complexity arises in any system in which many agents interact and adapt to one another and their environments. These interactions and adaptations result in evolutionary processes and often surprising "emergent" behaviors at the macro level. Complexity science attempts to find common mechanisms that lead to complexity in nominally distinct physical, biological, social, and technological systems.

There have been many attempts to formalize the modern notion of a *complex system*. For example, Herbert A. Simon provided the following definition (cf. [440], pp. 467–468):

> Roughly, by a complex system I mean one made up of a large number of parts that interact in a nonsimple way. In such systems, the whole is more than the sum of the parts, not in an ultimate, metaphysical sense, but in the important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole.

Herbert Simon has used interactions in complex systems. Moreover he underlined the importance of the idea of hierarchy (and therefore, of architecture) of complex systems, defined as follows (cf. [440], p. 468):

> By a hierarchic system, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem.

The concept of a system and the meaning of its scope are the subject of consideration in many important standardization processes. For example, there is an international standard for the system and software architecture descriptions (ISO/IEC/IEEE 42010 Systems and software engineering—Architecture description). The concept of a system as used in this International Standard is intended to encompass, but not to be limited to, entities within the following domains:

1. **Systems** as described in the Standards ISO/IEC 15288:2015[2]: *systems that are man-made and may be configured with one or more of the following tools: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials and naturally occurring entities*;
2. **Software products and services** as described in the Standards ISO/IEC 12207: 2008[3];

---

[2]http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63711.

[3]http://www.iso.org/iso/catalogue_detail?csnumber=43447.

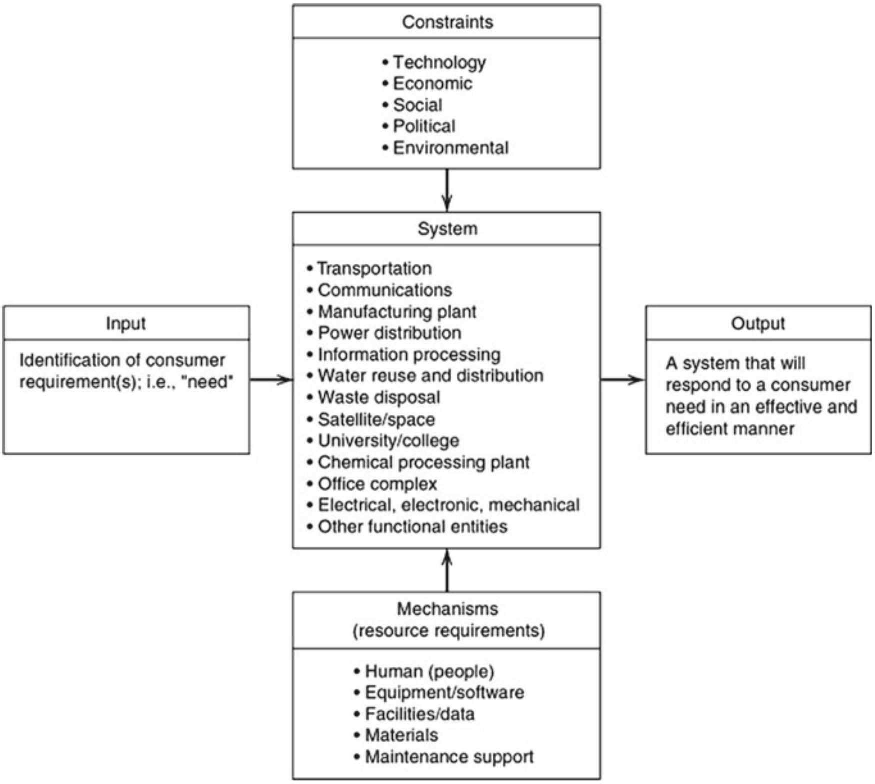


Fig. 6.1 The man-made system [52]

3. **Software-intensive systems** as described in IEEE Recommended Practice for Architectural Description for Software-Intensive Systems 1471–2000 (IEEE Std 1471:2000)[4] *any system where software handsomely influences the design, construction, deployment, and evolution of the system as a whole* to encompass *individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest*.

However, the ISO/IEC/IEEE 42010 standard does not define what constitutes a system within these domains—or elsewhere. As a matter of fact, the nature of a system is not defined by the standard. The ISO/IEC/IEEE 42010 standard is intended to use within the domains of the above-listed systems. However, there is a clear message in it—*nothing herein precludes the use of a system outside these domains for the sake of architecture descriptions of entities of interest (*e.g., ***natural systems*** *and* ***conceptual systems****)*.

This book especially focuses on man-made systems. This intuitive concept is illustrated in Fig. 6.1 [52].

[4]https://standards.ieee.org/findstds/standard/1471-2000.html.

6.2 Software Intensive Systems

By a *software intensive system*, we understand complex systems in which software essentially influences the design, construction, deployment and evolution of the system as a whole. Such systems have significantly changed the modern world and the experience gained through their implementation has led to the formation of a standardized summary and systematization of particularly important concepts and their relationships. In this context, the IEEE 1471 standard is especially noteworthy as on its basis, the ISO/IEC 42010 standard was developed.[5] All case studies discussed in Part IV, are devoted to systems using software intensively, therefore they are software intensive systems.

6.3 Cyber-Physical System (CPS)

The concept of *Cyber-Physical System* (CPS) has been defined in numerous ways. Probably, the most concise definition is the one that can be found on Wikipedia[6]:

> CPS is a system of collaborating computational elements controlling physical entities. [...] Ongoing advances in science and engineering will improve the link between computational and physical elements by means of intelligent mechanisms, dramatically increasing the adaptability, autonomy, efficiency, functionality, reliability, safety, and usability of CPS.

A more precise definition is provided by "The TerraSwarm Research Center"[7]:

> *Cyber-Physical Systems (CPSs) are integrations of computation, networking, and physical processes.*
> *Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. The economic and societal potential of such systems is vastly greater than what has been realized, and major investments are being made worldwide to develop the technology.*
> *The technology builds on the older (but still very young) discipline of embedded systems, computers and software embedded in devices, whose principle mission is not computation, such as cars, toys, medical devices, and scientific instruments. CPSs integrates the dynamics of the physical processes with those of the software and networking, providing abstractions and modeling, design, and analysis techniques for the integrated whole.*

Figure 6.2 presents an accurate visual representation of a high complexity CPSs, using a concept map.[8] The starting point is a taxonomy provided by S. Shyam Sunder

[5]http://www.iso-architecture.org/42010/.

[6]http://en.wikipedia.org/wiki/Cyber-physical_system.

[7]http://www.terraswarm.org/.

[8]http://www.terraswarm.org/.

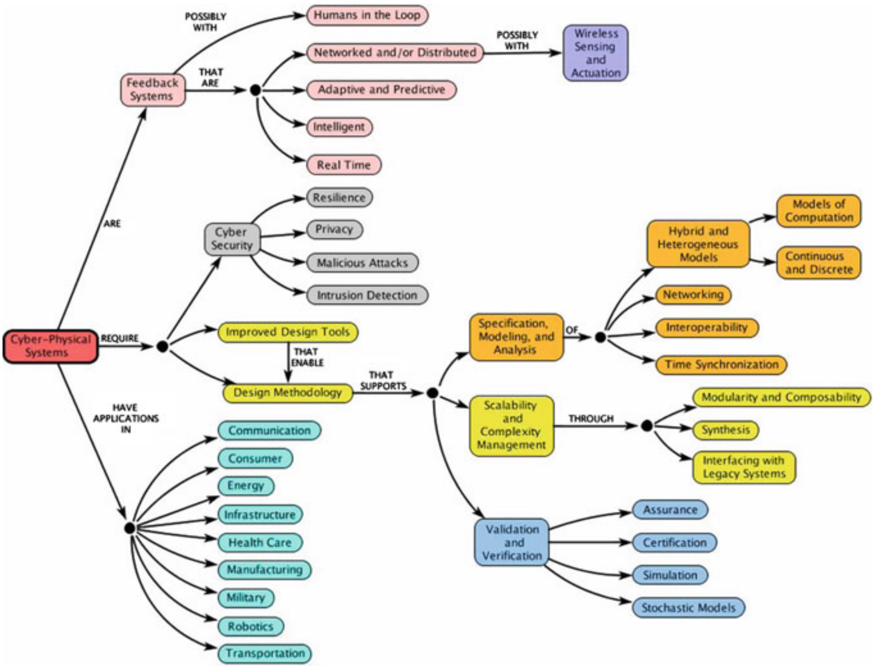


Fig. 6.2 CPS-Concept-Map (cf. http://www.terraswarm.org/)

from NIST at the NIST CPS Workshop that took place on March 13, 2012 in Chicago. Edward A. Lee from UC Berkeley converted the taxonomy into a picture and then, started to develop it with input from various people. So far, a direct contribution to the current version have been made by: Philip Asare, University of Virginia, David Broman, UC Berkeley, Edward A. Lee, UC Berkeley, Martin Torngren, KTH and S. Shyam Sunder, NIST.

6.4 Internet of Things and Wisdom Web of Things

Having in mind possible applications and the next generation technology for the future versions of the Internet (treated as a network of cooperating and/or competing intelligent autonomous agents), the concepts of the Internet of Things (IoT) and the Wisdom Web of Things (W2T) are of particular interest. In Fig. 6.3 ([211], p. 37) one can find key features of IoT relative to Machine-to-Machine approach.

It is the next generation of networks, which should provide ubiquitous wisdom services in a ubiquitous network in the "hyper world." A general idea of the concept of hyper-world [557] is nicely presented in the Figs. 6.4 and 6.5.

Aspect	M2M	IoT
Applications and services	Point problem driven	Innovation driven
	Single application - single device	Multiple applications - multiple devices
	Communication and device centric	Information and service centric
	Asset management driven	Data and information driven
Business	Closed business operations	Open market place
	Business objective driven	Participatory community driven
	B2B	B2B, B2C
	Established value chains	Emerging ecosystems
	Consultancy and Systems Integration enabled	Open Web and as-a-Service enabled
	In-house deployment	Cloud deployment
Technology	Vertical system solution approach	Horizontal enabler approach
	Specialized device solutions	Generic commodity devices
	De facto and proprietary	Standards and open source
	Specific closed data formats and service descriptions	Open APIs and data specifications
	Closed specialized software development	Open software development
	SOA enterprise integration	Open APIs and web development

Fig. 6.3 Comparison of the main characteristics of M2M and IoT ([211], p. 37)

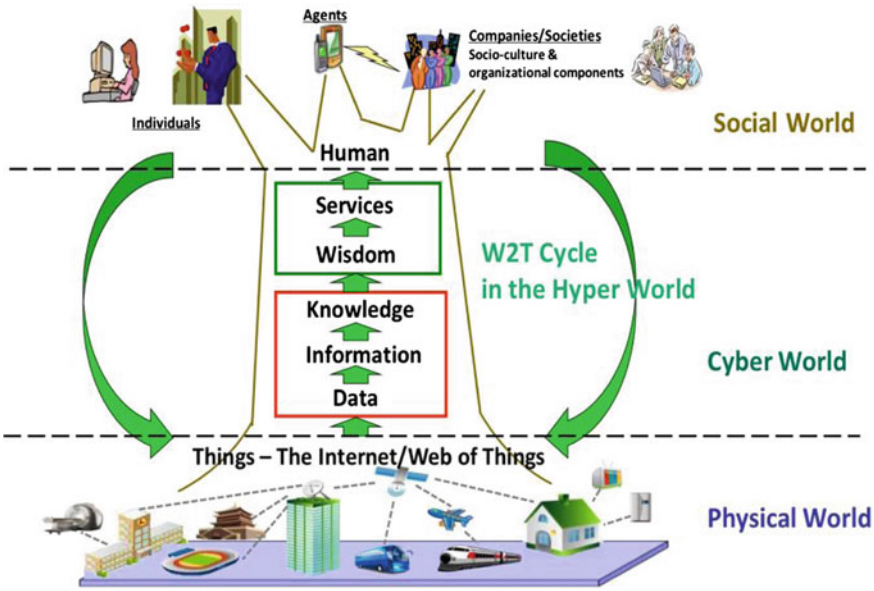


Fig. 6.4 A W2T cycle in the hyper-world ([557], p. 30)

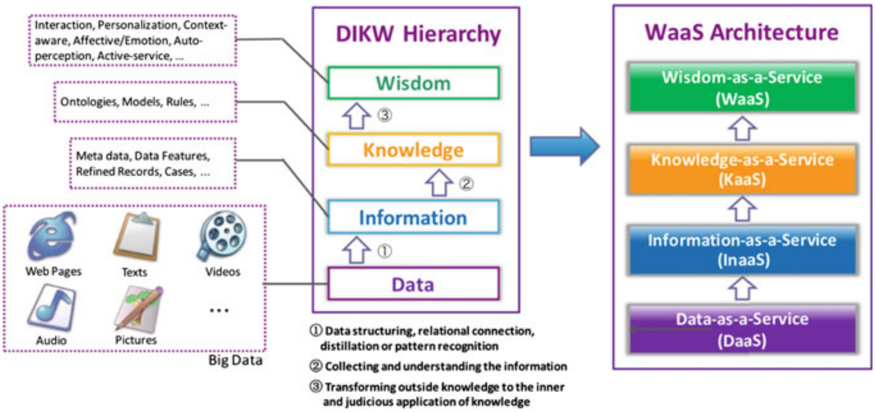


Fig. 6.5 From the DIKW hierarchy to WaaS ([557], p. 32)

The following definitions present more detailed descriptions of what IoT [211] and W2T [556, 557] really are:

> The IoT is a widely used term for a set of technologies, systems, and design principles associated with the emerging wave of Internet-connected things that are based on the physical environment. [...] In the longer term, it is envisaged that an IoT ecosystem will emerge [...] allowing things and real world objects to connect, communicate, and interact with one another in the same way humans do via the web today.

> [Hyper world] consists of the cyber, social, and physical worlds, [...] [Wisdom Web of Things] focuses on the data cycle, namely from things to data, information, knowledge, wisdom, services, humans, and then back to things. A W2T data cycle system is designed to implement such a cycle, which is, technologically speaking, a practical way to realize the harmonious symbiosis of humans, computers, and things in the emerging hyper world.

6.5 Ultra-Large-Scale Systems (ULS-S)

In this section, the concept of *Ultra-Large-Scale Systems* (ULS-S) is introduced[9] [141, 369].

According to [369], the concept of ULS-S could be defined as follows:

> *Ultra-Large-Scale (ULS) System*: A system at least one of whose dimensions is of such a large scale that constructing the system using development processes and techniques prevailing at the start of the 21st century is problematic. ULS systems exhibit the following characteristics: decentralization; conflicting, unknowable, and diverse requirements; continuous evolution and deployment; heterogeneous and changing elements; erosion of the people/system boundary; and normal failures of parts of the system.

[9] http://2013.icse-conferences.org/documents/publicity/ICSE_Keynote-Northrop.pdf.

Research portfolio for ULS-S covers the following domains [369]: (i) Human Interaction, (ii) Computational Emergence, (iii) Design, (iv) Computational Engineering, (v) Adaptive System Infrastructure, (vi) Adaptable and Predictable System Quality, (vii) Policy, Acquisition, and Management.

The concept of ULS-S dates back to around 2006. The importance of ULS-S applications is growing very quickly. Unfortunately, it is very difficult to manage the complexity of engineering problems related to ULS-S. According to the first motto of the book the progress in this area is very slow.[10]

[10] http://2013.icse-conferences.org/documents/publicity/ICSE_Keynote-Northrop.pdf.

Chapter 7
Examples of Complex Systems

In everyday life, one can find many mutually interactive systems that facilitate or hinder our actions. Of course, only some of them are of significance to us, whereas many others do not have any influence on our life.

However, in order to improve the quality of our life, we must try to understand the mechanisms that may enhance those systems that are important for us. We have already discussed the construction and development of such systems on the basis of illustrative examples, as given in the case studies in Part IV. On one hand, these examples clearly highlight the importance of such complex systems in our everyday lives, and on the other they indicate a serious weakness of contemporary approaches towards their effective development and control

To a large extent, this weakness stems from the limitations of contemporary paradigms and the commonly used approaches to the inference of cause—effect relationships, and from the shortcomings of approaches towards knowledge representation and wisdom acting.

In the following chapters, we present an approach towards both of these issues on the basis of WisTech, implemented in the form of interactive (rough) granular computations.

7.1 Natural Computing Systems

By a *natural computing system*, we understand one of the following types of systems [265, 420]:

- computing systems, which perform computations taking inspiration from nature and are used for the development of novel problem-solving techniques

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_7

(e.g., cellular automata, evolutionary algorithms, swarm intelligence, artificial immune systems, fractal geometry, artificial immune systems, membrane computing, amorphous computing);
- computing systems based on artificial synthesizing of natural phenomena, by means of computing processes (e.g., “artificial life”);
- computing systems that employ natural materials, e.g., molecules, DNA computing, quantum computing, as the physical support of computation.

Over the years, many interesting results concerning natural computing [420] have been published and they have proven to be important for the development of intelligent systems. Natural computing is also essential for the development of many approaches presented in this book. The contemporary approach to IGrC based on WisTech is, in a sense, a generalization of different approaches to natural computing. We will, however, focus on analyzing the context on which this research field emerged, rather than on other approaches to natural computing, which can be found in other relevant publications [420].

Ever since the first projects of electronic computers appeared, people have used simulations of various phenomena occurring in nature. In the forties of the twentieth century, the following two concepts, developed by computing architectures, were used:

- The Classical von Neumann Architecture, presented as a modification of the computational model for commutable functions, developed by Alan Turing and known as the Turing machine model.
- Von Neumann—Ulam Architecture, presented as a distributed processing architecture, based on some biological insights, proposed by John von Neumann and Stanisław Ulam in the form of a cellular automata.

It is significant that the concepts of these two architectures were proposed roughly at the same time. This demonstrates the major dilemmas that lie ahead of the architects of the first electronic computing systems.

In the first half of the twentieth century, many mathematicians were convinced that the rapid development in mathematics will stimulate the rapid development of computer applications.

In the 50s, this belief led, among other things, to the creation of a number of ambitious research programs that were aimed at a rapid development of AI. These programs were based on calculations made by the von Neumann architecture computers and, to a large extent, on the “improvement” of the Turing computing model for recursive functions, implemented, with some modifications, in the form of the von Neumann architecture and Boolean algebra.

It is worth noting that the first implementations of computer architecture was strongly influenced by the so-called Church—Turing thesis, which posited that any function is algorithmically computable in an intuitive sense if and only if it is computable by a Turing machine [508, 509].

In those days, scientific views neglecting the Church—Turing thesis were often strongly criticized. The prevailing opinion was that effective calculations lead to recursive functions.

However, there were some warnings that the essence of many practical problem solving procedures cannot be reduced to feasible computable functions. There are many practical problems which do not have any efficiently computable solutions.[1] Luckily, in practice, it is very often sufficient to find an approximate solution that satisfies certain constraints instead of a very precise solution, but in a non-effective way. Intuitively speaking, we do not need overly precise road maps with unnecessary details. To the contrary, we only need relevant information (e.g., related to crossings of roads).

Independently, there were many important philosophical questions regarding the relationship between effective problem solving and recursive functions in terms of the Church-Turing thesis. Among them some are as follows.

1. Undecidability of arithmetic, presented by Kurt Gödel in 1931 [168] which posits that the problem whether a given sentence of arithmetic is its theorem or not is undecidable, i.e., there does not exist a decision procedure checking if a given sentence of arithmetic is its theorem or not.
2. The thesis of Alonzo Church, formulated in 1936 [88], has proved that the set of first-order tautologies with binary predicate or function symbols is not recursive ("not decidable"),
3. The thesis by Ludwig Wittgenstein who in the 30-ties and 40-ties of the XX century summarized his book entitled "Philosophical Investigations" [539], pointed out that the natural language has relatively little to do with artificial formal languages, and natural languages should be treated as dynamic complex objects, resulting from interactions between people, in terms of Wittgenstein they are called as "language games".[2]
4. On the basis of Turing test, introduced by Alan Turing in 1950 [510], one may decide whether a computational process is a manifestation of "intelligent behavior".[3]

In spite of the above-mentioned arguments put forward by the opponents of the Church—Turing thesis, many research centers undertook ambitious projects aimed at constructing sophisticated mathematical models for implementing complex projects, using computational processes executed on the basis of the von Neumann architecture.

The growing belief in the von Neumann architecture was grounded on the claim that advanced mathematical methods, aided by computers, will allow us to effectively deal with complex problems that occur in nature, especially in biology. This belief was also growing on the basis of various projects aimed at making different mathematical representations of recursive functions closer to hardware.

For example, in the 70s of the 20th century, Robert Kowalski initiated a very interesting trend of declarative programming, based on the principle of resolution [282]. This idea provided the foundation for the great projects of computers

[1] http://en.wikipedia.org/wiki/NP-complete.

[2] http://en.wikipedia.org/wiki/Language-game_philosophy.

[3] http://en.wikipedia.org/wiki/Turing_test.

which attempted to employ, as the basis, Boolean inferences rather than arithmetic operations.

When it comes to the domination of the Church—Turing thesis followers, a lot of consternation was caused by Stanisław Ulam's famous statement, which questioned the superiority of the traditional mathematical models in problem-solving over the possibilities that emerged from a better understanding of the processes occurring in nature, especially in biology [91]:

> Ask not what mathematics can do for biology, rather ask what biology can do for mathematics.

This statement emphasized that a better understanding of the phenomena in nature can be helpful in a better understanding of both mathematics and computational processes carried out in the natural world.

Some authors even went so far as to suggest the treatment of perception of various aspects of nature as interactive computational processes. For example, the Zuse-Fredkin thesis [558], dating back to the 1960s, states that the entire universe is a huge cellular automaton which continuously updates its rules. Of course, the model based on the cellular automata is one of the many approaches to the idea of treating natural phenomena as interactive computations. Another one is based on quantum mechanics. It has been suggested by Seth Lloyd, the professor of mechanical engineering at MIT [312], that the whole universe is a quantum computer, computing its own behavior.

The von Neumann—Ulam architecture initiated research on the development of alternative solutions to computing models based on the classical von Neumann architecture. The research in this area, particularly intensively developed in recent decades, has led to the emergence of *natural computing*.

7.2 Political Systems

By the *political system*, in our context, we mean the very first system that selects a range of entities with powers, privileges, responsibilities and relationships to support the management in a society of humans and/or robots. As an example of a political system development, one may consider political changes that occurred in Central and Eastern Europe during 1980–1989.

Such a broad context of political systems can be observed throughout the history.

However, in the context of our investigation, the artificial political systems of societies of automata (e.g., software robots) should be considered. Such systems could be used in many applications, e.g., in algorithmic trading (cf. the AlgoTradix case study in Chap. 17), where the society of software robots is being organized in the evolving of local political systems. Then, the best individual, in terms of the profit-to-risk (ROI/VOL) ratio, could be selected from the local "political system" of software trades to decide whether to buy or sell something on electronic financial markets.

7.3 Economical Systems

Analogously to political systems, *economical system* is also a very broad term. In simple words, one may say that in the context of the worsening economic crisis, there exists an urgent need for the development of effective tools that would enable us to understand the principles behind the design and development of complex economic systems that could satisfy great needs of both national and international societies in a better way.

Surely, the access to such tools helps to establish scientific directions for the further development of economic systems in different countries. It should be noted that with the technological advances, there appears a new, ever more, powerful economic system based on electronic platforms that create the entirely new opportunities for the economy.

Moreover, in this type of an economic system, the human involvement in the old style economics (direct trading) is diminishing. This role—is gradually assumed by the societies of autonomous (software) robots—especially on electronic platforms. As a result, the role of a man has shifted to the proper supervision of risk management behavior displayed by robotic societies.

As far as the functioning of modern electronic financial markets is concerned, the human role becomes especially important when it comes to the design, construction and implementation of new technologies for the societies of robots, operating on modern electronic markets. This issue is addressed in the AlgoTradix case study (cf. Chap. 17).

7.4 Risk Management Systems

It is safe to say that the global financial crisis at the beginning of the twenty-first century is the best proof of a total lack of ability to build and develop *risk management systems* for complex systems.

To illustrate the scale of this phenomenon (especially for people unfamiliar with the world of banking and finance), let us remind that the Basel Committee on Banking Supervision[4] has a significant role in the risk management system of finance and banking. However, some analysts and market observers argue that the recommendations known as the Basel II document contributed to the crisis in the real estate market.[5] These recommendations were revised by the Basel Committee on Banking Supervision, which resulted in the creation of the Basel III document. At the same time, in 2011, the OECD published a report indicating that the implementation of Basel III would lead to a decrease in the economic growth by 0.5–1.5 per mille.[6]

---

[4] http://www.bis.org/bcbs/.

[5] http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1330417.

[6] http://www.oecd-ilibrary.org/economics/macroeconomic-impact-ofbasel-iii_5kghwnhkkjs8-en.

On the basis of the gigantic experience gained during the construction of modern risk management systems, the ISO 31000 standard has been developed. The standard summarizes knowledge, collected and structured on the basis of this experience.

Basically, each case study is based on its particularly important component of managing risk in individual systems. In particular, this applies to the following projects presented in Part IV: POLTAX, Merix, AlgoTradix, and UAV. As will be explained later in Chap. 33 risk management (as a part of efficiency management) could be used for interaction control.

7.5 Quality Assurance Systems

Quality assurance management system (QA) can be an integral part of every significant complex system. Currently, many approaches towards the quality assurance management are rapidly developing in all major economies.

The majority of these approaches are based on ideas formulated about one hundred years ago by Shewhart and Deming. Nowadays, they are often embedded in the so called *Plan-Do-Check-Act cycle* (PDCA cycle).[7] It is worth noting that the essence of quality management systems, similarly to other systems discussed here, is largely based on the ability to establish the cause—effect relationship. In this regard, such systems are naturally associated with the already suggested approach to the design and development of complex systems based on WisTech.

The ISO 9000[8] standard summarizes collected and structured knowledge of QA.

In the context of our further case studies, quality assurance is particularly important for POLTAX, Merix, and AlgoTradix (cf. Part IV).

7.6 Information Security Management Systems

Analogous to the previously mentioned systems, *information security management system* is an integral subsystem of every major man-made complex system. Systems of this type are particularly related to the above mentioned risk management systems. The essential difference lies in the methods and ways of control used in such systems. Generally, such systems also significantly depend on cause and effect inferences and on how they represent knowledge and wisdom. These issues will be covered later in the book.

The ISO standards (e.g., ISO 27001) are expressed in natural language, using a natural variety of complex and vague concepts. Depending on the requirements, the concepts could undergo refinement through a dialogue between the participants of a given project and/or experiments. This belongs to an essence of the postulates regarding the models for Interactive Granular Computing (IGrC) (cf. Chap. 32).

[7] http://en.wikipedia.org/wiki/PDCA.

[8] https://en.wikipedia.org/wiki/ISO_9000.

7.7 Other Examples of Complex Systems

Of course there are many other examples of Complex Systems in all domain of complex applications. Some of them we can find in Wikipedia.[9] An interesting example of organizational map of complex systems broken into six sub-groups can be found in Fig. 7.1[10] created by Hiroki Sayama.

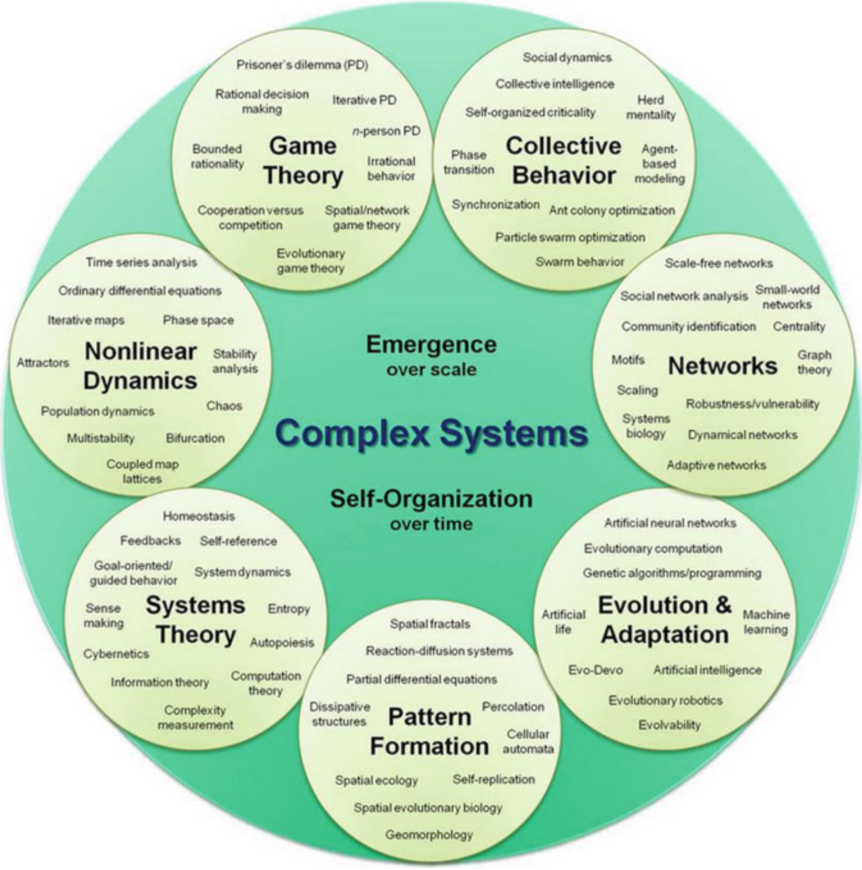


Fig. 7.1 Organizational map of complex systems broken into six sub-groups of aspects of complex systems models [*source* created by Hiroki Sayama, Collective Dynamics of Complex Systems (CoCo) Research Group at Binghamton University, UK]

---

[9]https://en.wikipedia.org/wiki/Complex_system.

[10]https://commons.wikimedia.org/wikiFile:Complex_systems_organizational_map.jpg.

Chapter 8
Concept of Complex Systems Engineering (CSE)

> *Learning how to be a successful systems engineer is entirely different from learning how to excel at a traditional engineering discipline. It requires developing the ability to think in a special way, to acquire the "systems engineering viewpoint", and to make the central objective the system as a whole and the success of its mission. The systems engineer faces three directions: the system user's needs and concerns, the project manager's financial and schedule constraints, and the capabilities and ambitions of the engineering specialists who have to develop and build the elements of the system* [279].

Typically, by *engineering*, we mean [279] the following concept:

> the application of scientific principles to practical ends; as the design, construction and operation of efficient and economical structures, equipment, and systems.

There are many definitions of the concept of *systems engineering*. In general, it is an interdisciplinary field of engineering, which focuses on the engineering of complex systems. According to the book [279]:

> The function of systems engineering is to guide the engineering of complex systems. ... To guide is defined as "to lead, manage, or direct, usually based on the superior experience in pursuing a given course" and "to show the way."

The history of mankind is full of various activities which resulted in the engineering of new complex systems. Unfortunately, this history also shows that very often, these systems do not meet the expectations or, on the contrary, they lead to wasting of time and resources. As a classical example, one may consider numerous efforts made over the past decades that were aimed at the systematization of knowledge and

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_8

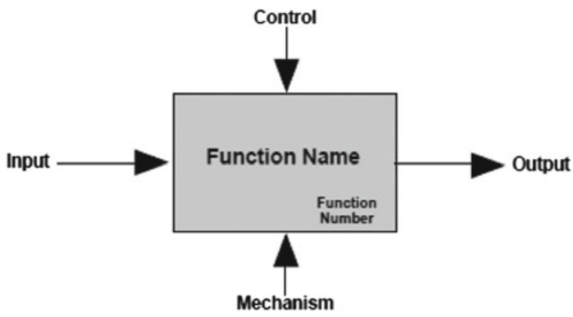


Fig. 8.1 Integration definition for function modeling [107]

experience to decrease risk related to the development of new software-intensive systems, i.e., systems in which software interacts with other software, systems, devices, sensors or people.

Having in mind a very large scope for applications of complex systems, it is easy to imagine how important the factor of successful system engineering is. In particular, it is important to understand and learn some general mechanisms behind system engineering and develop those, whose implementation could increase the chances of achieving the best possible targets in the best possible way, using limited resources.

The successful implementation of systems engineering projects is a very difficult task. It is especially difficult to achieve success when it comes to engineering projects, which have the following objectives: the development and implementation of complex systems based on new complex technology (e.g. complex software for new AI technology) and new appropriate culture of work (e.g., new decision procedures based on new technologies).

Currently, we have access to extensive literature devoted to the field of CSE, which provides all the basic knowledge. Any experienced practitioner possessing considerable experience in the implementation of CSE projects, has her/his own subjective opinion concerning various approaches towards CSE.

According to the subjective experiences of the author, a proper introduction to the traditional approach towards CSE can be found in such publication as [52, 107, 141, 192, 279]. In [107] particularly good example of a well condensed introduction to systems engineering, has been focused.

The document [107] provides recommendations for several methods of describing processes, developed on the basis of a relatively old standard—IDEF0.[1] According to this standard, the process is visualized in Fig. 8.1. In this book, the concept of c-granule will be presented in an analogous manner.

Figure 8.2 presents basic relationships between main activities that are recommended as the components of systems engineering management [107]. Figure 8.3, however, is a schematic representation of the traditional approach towards systems engineering process.

[1] http://en.wikipedia.org/wiki/IDEF.

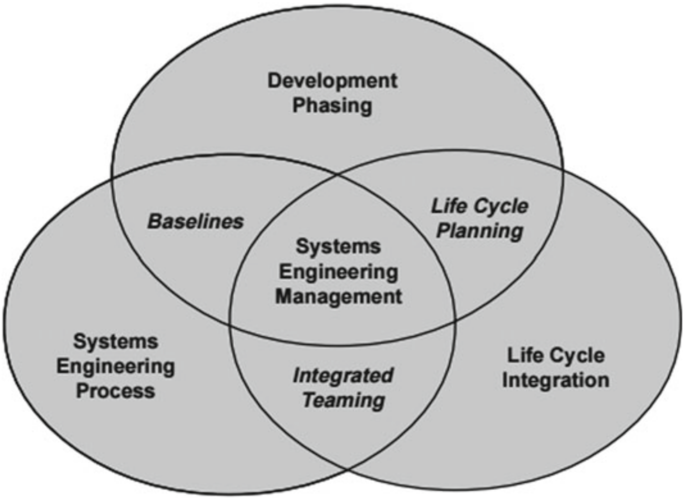


Fig. 8.2 Three activities of systems engineering management [107]

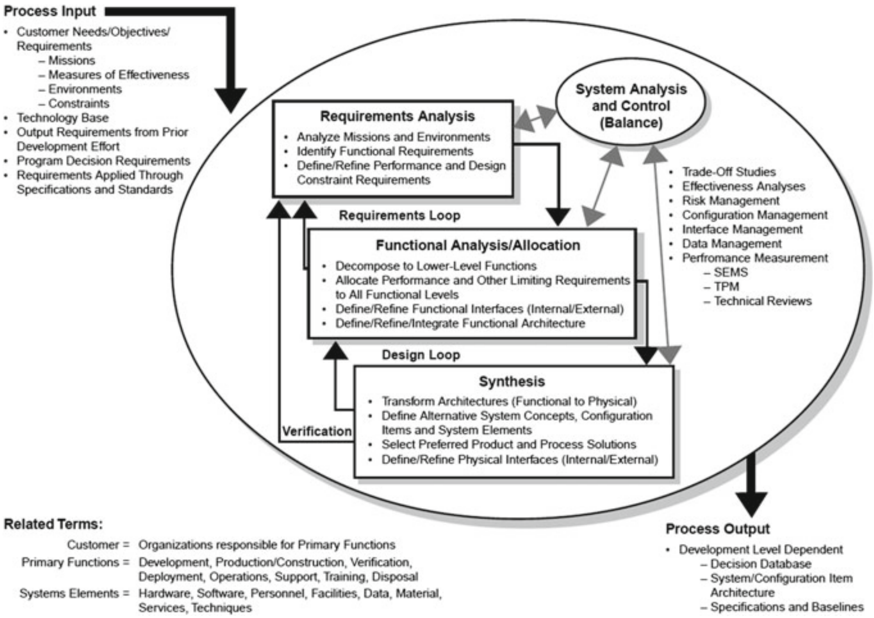


Fig. 8.3 The systems engineering process [107]

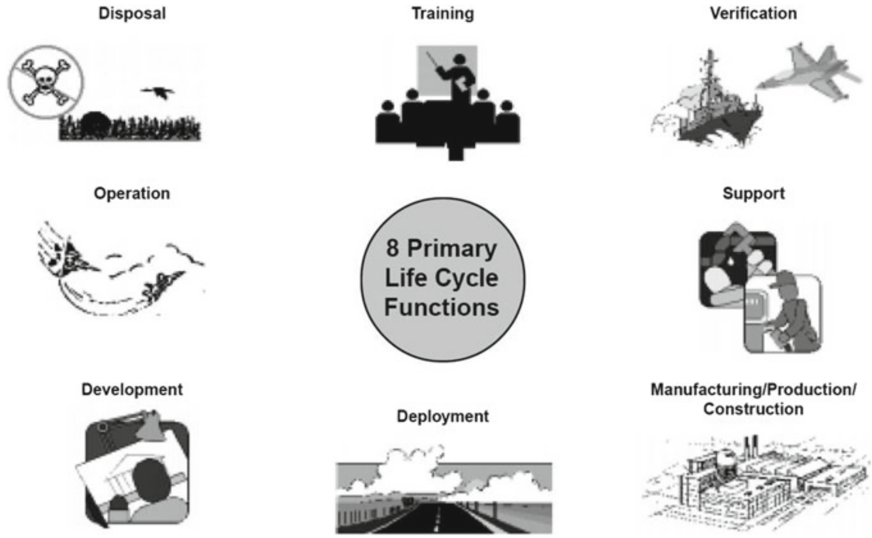


Fig. 8.4 Primary life cycle functions [107]

These processes are conducted within the so-called life cycles. Figure 8.4 presents primary typical life cycles, commonly known in CSE.

However, it is not our aim to present here the whole concept of systems engineering. This chapter is intended to prepare the ground by introducing basic terminologies and results from relevant literature in this field. An important issue concerning the CSE projects is to emphasize the great discrepancy between CSE projects in theory and practice as soon as the concept of systems engineering is introduced.

One of the related observations can be described using the following paradox: the more knowledge we have about the projects that failed and the more sophisticated our methods against unsuccessful ventures are, the more painful and more unpredictable the losses are in case the software does not fulfill its goals.

Of course, many projects are successful and one may observe the great progress in the discovery of knowledge concerning the methods of project management and quality assurance mechanisms.

However, all practicing engineers and managers are well aware of the fact that possessing only a theoretical knowledge is highly inadequate when quick and sensible decision should be made during the implementation of complex projects. These decisions depend, to a large extent—on compound information about the context of the situation and, above all, on the experience and the sense of responsibility of a person implementing the project.

To emphasize this important difference between theoreticians and practitioners involved in the development of complex systems, let us consider the following metaphorical example:

Imagine that Bob has never driven a car, but he has a great memory and read all the important books on—traffic code and driving methods. John, in turn, has never read a single book on car driving. However, he used to play with other children using small toy cars, and his parents would let him, from time to time, drive their real cars. One may therefore say that John has a very substantial experience in driving. Thus, the question of who is a better driver, Bob or John, is clearly a rhetorical one.

Any driver who has ever experienced a sudden dangerous situation (especially when tired) knows that it is not possible to recall a relevant paragraph from the huge volumes of traffic rules or/and driving car rules, when fraction of a second matters. In such a case, the driver should be guided by her/his own intuition and act instinctively to avoid dangerous situations.

To a large extent, the same situation appears in complex systems engineering. When it comes to the quality of the systems, *it is very important to employ people who can "instinctively" combine a vast practical knowledge (*e.g., *"behavior rules on the road") with a specific project (*e.g., *"identifying unexpected situations on the road").*

Having in mind the above-mentioned remarks related to the importance of practical skills, systems engineering could be treated as the aggregation of the following three practical factors:

1. Physical and social **INTERACTIONS** based on appropriate practical skills and abilities that are necessary for the implementation of complex systems,
2. Engineering management based on appropriate **ADAPTIVE JUDGMENT** of "what to do next" and "how to identify the most important next activities" to ensure the success of the project,
3. Technical **KNOWLEDGE** of the domain in which a systems engineer operates.

In other words, the quality of candidates for getting employed in the construction and development of complex systems is contingent with their abilities to deal with these three factors. Both well-rounded education and experience are needed to provide such highly-qualified specialists. Practice shows that in the case of non-trivial projects, the key factor is the experience gained over the course of many years (or even decades). The only exception is when the key concepts of the domain knowledge and the representation of these concepts can be detected and included in the educational materials. They should be designed to illustrate phenomena particularly important during the practice of a beginner adept, a candidate for working as an engineer of complex systems.

Figure 8.5 presents an illustration of the "T" model for the career development of a systems engineer. According to the "T" model, typically it requires more than 20 years of practice in systems engineering to become a good leader of Systems Engineering.

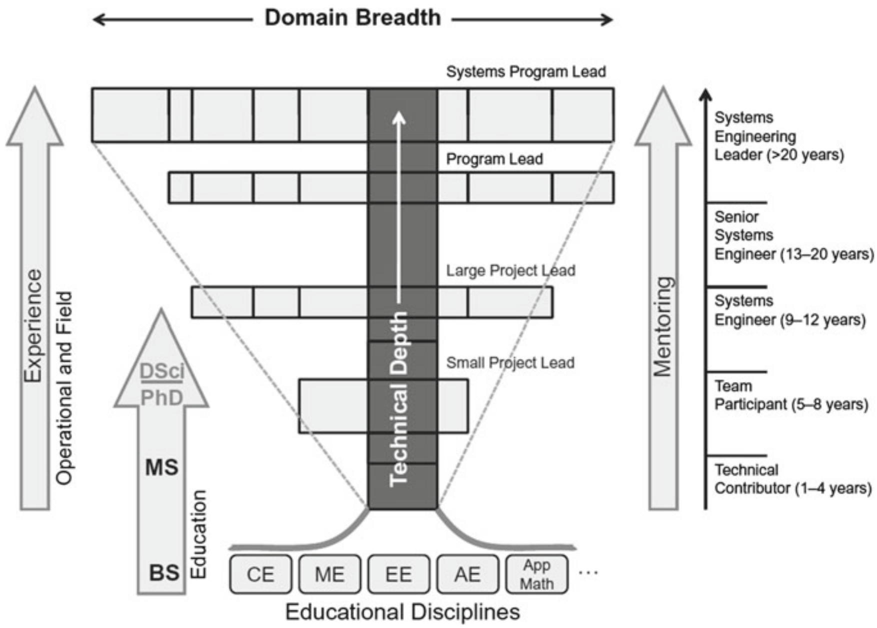


Fig. 8.5 "T"—model for systems engineer career development. CE—chemical engineering; ME—mechanical engineering; EE—electrical engineering; AE—aeronautical engineering; App Math—applied mathematics [279]

Let us summarize, all professionals for CSE, should be able to manage at least the following three issues:

1. **INTERACTIONS**—ability to perform important tasks and observe important phenomena in the environment, combined with the ability to act/react, based on appropriate interactions with the environment (for example, to complete the planned tasks, similar to the ability of "driving").
2. **ADAPTIVE JUDGMENT**—based on hierarchy of needs, reasoning, intuition and adaptive instinctive reactions that are used for quick and adaptive assessment of changes, according to one's subjective perception of a given situation. In particular, adaptive judgment is related to habits, assessment of situations, and decision making skills. The results of judgment are then implemented and used to establish and continuously adapt priorities of needs.
3. **KNOWLEDGE**—having the right knowledge to interpret perceived phenomena as well as draw conclusions.

These three conditions constitute the pillars of WisTech program [244, 245, 247, 253, 448], concisely—represented by the following metaphorical *Wisdom Equation* (cf. Chap. 29):

$$\text{WISDOM} = \text{INTERACTIONS} + \text{ADAPTIVE JUDGMENT} + \text{KNOWLEDGE}\,. \tag{8.1}$$

One of the main goals of this book is to propose mechanisms for the investigation of the key terms associated with a given domain knowledge related to applications, and to introduce a computational model that could simulate the behavior of social groups of "intelligent agents", equipped with the basic components of the Wisdom Equation (cf. Chaps. 29 and 33).

The goals established in this book are also meant to stimulate a further development of knowledge on CSE and, what is most important, put theory into practice. For instance, it would be a huge success if the development of knowledge on CSE could play some role in teaching new generations of engineers and shorten the length of their education.

Part III
The Theory-Practice Gap Problem (TPGP)

Chapter 9
CSE Practice: CSE Crisis

Reality is the murder of a beautiful theory by a gang of ugly facts.

Robert Glass [164]

9.1 CSE Crisis: Introduction

There are many books dedicated to CSE (e.g., see [107, 141, 192, 279, 356, 424]) which encompass essential knowledge, based on considerable experience in many domains of CSE applications. The history of innovative CSE[1] shows that it is not easy to achieve success in the implementation of such projects. According to [19]:

> The traditional approach to large engineering projects follows the paradigm established by the Manhattan project and the Space program. There are several assumptions inherent to this paradigm.
>
> First, that substantially new technology will be used.
>
> Second, the new technology to be used is based upon a clear understanding of the basic principles or equations that govern the system (i.e. the relationship between energy and mass, $E = mc^2$, for the Manhattan project, or Newton's laws of mechanics and gravitation F=-G* Mm/r^2 for the space program).
>
> Third, that the goal of the project and its more specific objectives and specifications are clearly understood.
>
> Fourth, that based upon these specifications, a design will be created essentially from scratch and this design will be implemented and, consequently the mission will be accomplished.

[1] http://en.wikipedia.org/wiki/Systems_engineering.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_9

Unfortunately for many CSE projects, the second and the third of the above assumptions are not true. The chances for the success of such projects are not very high (cf. Chap. 1, especially Sect. 1.3).

It turns out that in the case of many projects devoted to the construction and development of an innovative complex system, there exists **a large discrepancy between the "beautiful" theoretical assumptions concerning the implementation of the project and an "ugly" practice of its actual implementation**.

This discrepancy often demands great "acrobatic skills" and experience from the participants of a project in order to avoid its failure. Nevertheless, in many cases, the "acrobatics" and experience of the participants cannot guarantee the success. Thus, every modern, big country and/or company have their own long lists of large engineering projects that ended up as failures. Often, the cost of such failures incur loss in billions USD. A sample of such a list can be found in [19]. As a consequence, each practitioner should understand the great wisdom hidden in the quotations that have been used as the mottos of this chapter.

On the basis of the above-mentioned experience, we can define the concept of the *CSE crisis*. Roughly speaking, the concept refers to the **difficulty of engineering useful and efficient innovative complex systems within the pre-imposed constraints related to time, resources, cost, functionality, scope, and quality**.

The CSE crisis affects all areas where innovative complex projects (cf. 1.3), aimed at the construction and application of new systems, are used. It particularly encompasses such types of systems, which are used to illustrate the concept of a complex system in Chaps. 6 and 7.

It should be noted that from a theoretical point of view, one type of man-made complex systems stands out from the rest to some extent. Namely, it is easy to notice that if we were able to develop and engineer efficient risk management systems in any area of application, it would be also possible to engineer risk management systems for other CSE projects. This shows that the concepts of risk and risk management are very complex. On the other hand, it proves the interdisciplinary nature of risk management and shows the extent to which the ability to apply a domain knowledge influences the process of CSE project engineering in practice.

Later in the chapter, the concept of CSE crisis will be illustrated by means of the following examples of its application:

i. *Software crisis*.
ii. *Global Financial crisis*.[2]
iii. *Risk management systems*[3] illustrated by applications to *financial crisis forecasting*.

It should be emphasized that the above-mentioned examples illustrate only selected mechanisms of CSE crisis. By choosing these examples, we wanted to foreground the mechanisms which contribute to CSE and show their universality, illustrated by the design of complex systems which are not limited only to software

[2] http://en.wikipedia.org/wiki/Financial_crisis_of_2007-08.

[3] http://en.wikipedia.org/wiki/Risk_management_for_financialmarkets.

development. For this purpose, we have provided the examples of global and financial management systems.

These universal mechanisms are applicable in virtually every area in which innovative complex projects are engineered, tested, implemented and developed. The mechanisms and their causes are presented in more details later in the book, in Chaps. 12, 13, and 15.

9.2 CSE Crisis Case Study: Software Crisis

CSE crisis mainly applies to complex systems that are based on the *intensive use of software*. Serious difficulties related to the implementation of complex computer systems have been already noted in the 50s of the 20th century. In the 60s and 70s, these difficulties escalated to such an extent that the phenomenon was given the name of the *software crisis*.

Software crisis is a special case of the CSE crisis. Thus, roughly speaking, by a *software crisis*, we mean the **difficulty of writing useful and efficient computer programs within the pre-imposed constraints of: time, resources, cost, functionality, scope, and quality**. The term "software crisis" was coined by several attendees of the first NATO Software Engineering Conference, which took place in 1968 at Garmisch, Germany.

Given that the software is becoming an essential part of almost any complex man-made system, the software crisis has become an essential part of the CSE crisis. It should be underlined that the software crisis in not the only factor contributing to the CSE crisis. There are many other factors involved. However, from our point of view, it is a good idea to conduct the analysis of the CSE crisis factors starting from the analysis of the key factors that lead to the software crisis.

In the book [68], are described fundamental mechanisms determining the success or failure of complex computer system engineering. What is particularly important is that Brooks [68] raised the awareness of the following quite obvious fact among a broad society of software engineers and computer scientists:

> Complex programming projects **cannot be perfectly partitioned into discrete tasks that can be worked on without communication between the workers and without establishing a set of complex interrelationships between tasks**, the workers performing them and project decisions based on assessment of ongoing changes, which could have impact for the project.

In a very simple and convincing manner, Brooks [68] explains how problems arise out of the **complexity of various interactions (especially communication)**. The situation becomes especially problematic when we add new people to projects. Namely, **assigning more programmers to a project, which is running behind schedule, might slow it down** even further. This happens because the time required for the new programmers to familiarize themselves with the project and the increased data transfer overhead will consume an ever increasing amount of available time.

When n people have to communicate among themselves, then the number of communication channels and interactions (between any two persons) is:

$$\frac{n(n-1)}{2}, \tag{9.1}$$

and the number of all possible sub-teams for internal project interactions (for any kind of meeting, cooperation, competition in particular) is 2^n.

For example, when we consider a small team consisting of 30 members (e.g., engineers, developers, testers), then we have 870 channels of communication and interactions between any two persons from the team. The number of all possible sub-teams for the internal project interactions is 2^{30}, i.e. around one billion (10^9).

For a project team that is ten times larger, with around 300 members, the amount of all possible sub-teams for the internal project interactions is greater than the total number of atoms in the observable universe.[4]

It means that one of the critical factors determining the success of any complex development project is an appropriate approach to the integration of:

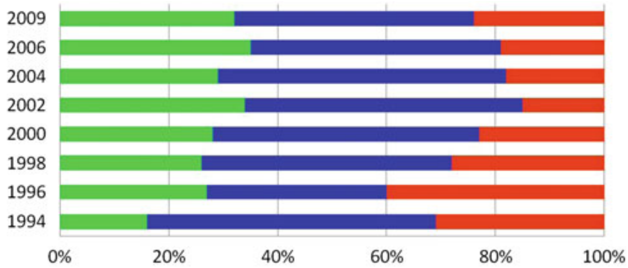
1. The **interactions (especially communication)** among sub-teams and/or other components of the project environment.
2. **Adaptive judgment/assessment and prioritization** of pre-identified project-related issues and/or changes—raised by the sub-teams.
3. Development of knowledge/skills, of the project members, based on **appropriate selection/update of know-how**, needed for the successful implementation of the project.

Understanding the techniques for the integration of the above-mentioned components is the main idea of WisTech, expressed by the Wisdom Equation (cf. Eq. 8.1).

The Standish Group Report from 2010 Fig. 9.1 [132] shows that there was a small progress in the quality of the projects that had been implemented so far. A summary of these results is shown in Fig. 9.4. It should be noted that not all R&D centers agreed with the methodology adopted by the Standish Group. An example of some critical remarks can be found in the following opinion from [132]:

> For many years, researchers and practitioners have analyzed how to successfully manage IT projects. Among them is the Standish Group, which regularly publishes its findings in its Chaos reports.
>
> In 1994, Standish reported a shocking 16% project success rate, another 53% of the projects were challenged, and 31% failed outright. In subsequent reports Standish updated its findings, yet the figures remained troublesome.
>
> These reports, derived from the Standish Group's longitudinal data, suggest that many efforts and best practices to improve project management hardly help increase project success. Over the years, their figures have attracted tremendous attention.
>
> However, we question the validity of their figures. Robert Glass, and Magne Jørgensen and his colleagues indicated that the only way to assess the Chaos results' credibility is to use Standish's data and reiterate their analyses.

[4] http://en.wikipedia.org/wiki/Observable_universe.



	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%
Failed	31%	40%	28%	23%	15%	18%	19%	24%

Fig. 9.1 Report by the Standish Group from 2010 [132]

Fig. 9.2 2015 Chaos Report by Standish Group [257] and https://www.infoq.com/articles/standish-chaos-2015

	2011	2012	2013	2014	2015
Successful	29%	27%	31%	28%	29%
Challenged	49%	56%	50%	55%	52%
Failed	22%	17%	19%	17%	19%

> But there's another way: obtain your own data and reproduce Standish's research to assess its validity. We applied the Standish definitions to our extensive data consisting of 5,457 forecasts of 1,211 real-world projects totaling hundreds of millions of Euros.
>
> Our research shows that the Standish definitions of successful and challenged projects have four major problems: they're misleading, one-sided, perverted in the estimation practice, and resulting in meaningless figures[. . .].
>
> **These case studies show that the Standish figures for individual organizations don't reflect reality and are highly influenced by forecasting biases. Because the underlying data has an unknown bias, any aggregation of that data is unreliable and meaningless.**

It should be underlined that Standish Group improved the methodology. In particular they have introduced an approach based on so called "modern resolution for projects." Results of the research are presented in [257] and in the latest versions of the Standish Group Chaos Report[5] (e.g., Figs. 9.2 and 9.3).

To illustrate the results obtained by different companies, let us look at some results published in 2012, reported by Gartner and McKinsey & Company. First, let us start with the Gartner report which is summarized, e.g., in the post entitled "Gartner Survey Shows Why Projects Fail," written by Lars Mieritz and published on June 1, 2012 (Fig. 9.5).[6]

There, one may find a very interesting summary of the Gartner Survey [336], which is based on the analysis of the collective responses from 150 participants of

---

[5] https://www.infoq.com/articles/standish-chaos-2015.

[6] http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/.

Size	Method	Successful	Challenged	Failed
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Fig. 9.3 2015 Chaos Report by Standish Group [257] and https://www.infoq.com/articles/standish-chaos-2015. Chaos resolution by agile versus waterfall methods

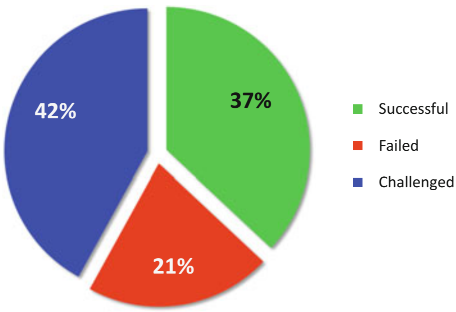


Fig. 9.4 Summary of the results from the Standish Group Report 2010: Project resolution from 2010 CHAOS Research https://cs.calvin.edu/courses/cs/262/kvlinden/resources/CHAOSManifesto2012.pdf

the Gartner survey, conducted in 2011 in five countries. The failure rate of IT projects with Budgets exceeding $ 1 million was found to be almost 50% higher than for the projects with Budgets below $350,000. Moreover, in this post, we may find very important key findings and recommendations [336]:

Key Findings

- Runaway budget costs are behind one-quarter of project failures for projects with budgets greater than$ 350,000.
- Small is beautiful—or at least small projects are easier to manage and execute. The failure rate of large IT projects with budgets exceeding $ 1 million was found to be almost 50% higher than for projects with budgets below$ 350,000.

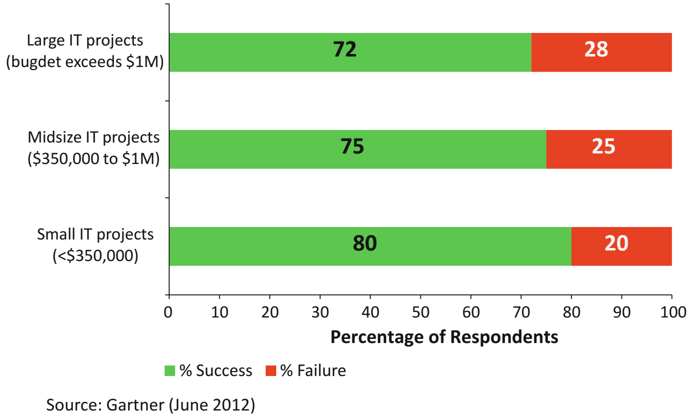


Fig. 9.5 Distribution of success and failure across projects of different sizes, according to Gartner [336]

Recommendations

- To optimize success, look for ways to limit the size, complexity and duration of individual projects, and ensure funding has been committed.
- Stay on top of costs, especially for the largest projects. Ensure that there are the appropriate mechanisms in place to identify budget variances and/or overruns early. Regularly review how cost estimation is done to understand how accurate and effective your approaches are, and pursue improvement opportunities.
- Keep the schedule realistic. Many large projects fail because business conditions keep changing after the project scope has been set, leaving a significant difference between the agreed-on scope and budget versus what the business will require and pay for by the time the project is delivered.
- Invest in truly capturing and understanding the business expectations and functionality sought from the project, and ensure that there is initial, adequate allocated funding, as well as good processes in place for revisiting the expectations and required funding at multiple points during the project.
- Increase the frequency of project status and review meetings, as well as ongoing confirmation of the project's alignment with business strategy—with an eye toward identifying and cancelling projects at the earliest possible stage that no longer meet company needs.

Figure 9.5 shows the distribution of success across projects of different sizes. Equally interesting are the conclusions contained in the report by McKinsey & Company entitled "Delivering large-scale IT projects on time, on budget, and on value", published in October 2012. The research was conducted on a quite large sample. In the report, we find that:

> Our research, conducted in collaboration with the University of Oxford, suggests that half of all large IT projects-defined as those with initial price tags exceeding$ 15 million-massively blow their budgets. On average, large IT projects run 45% over budget and 7% over time, while delivering 56% less value than predicted. Software projects run the highest risk of cost and schedule overruns (cf. Fig. 9.6).

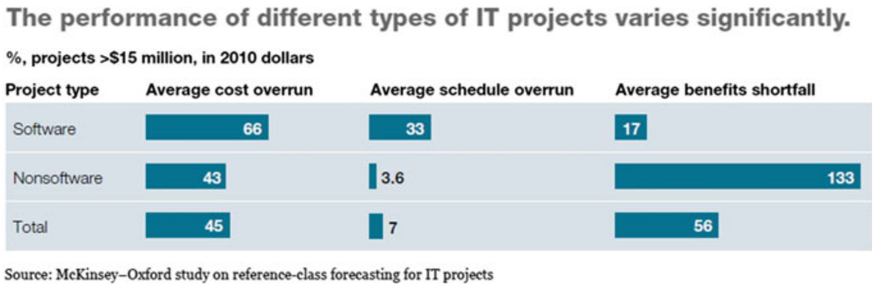


Fig. 9.6 The performance of different types of IT projects [54]

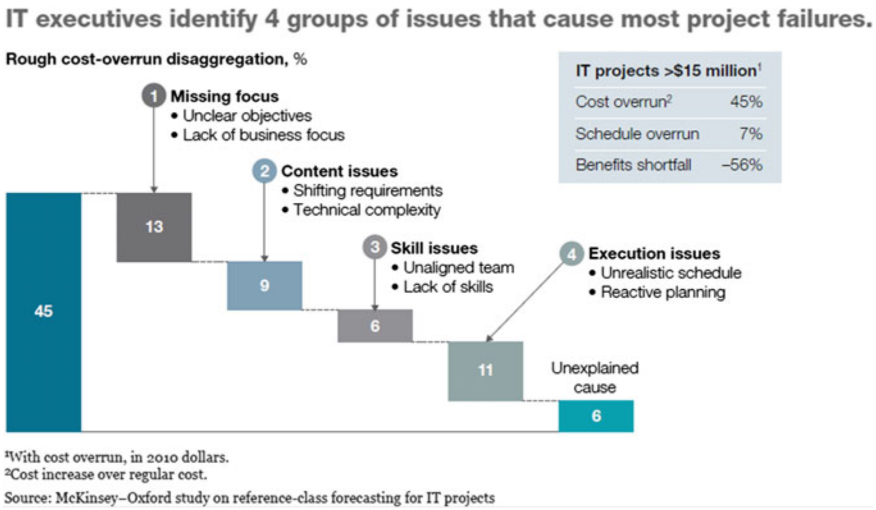


Fig. 9.7 Four groups of issues that cause most project failures [54]

In their study, McKinsey & Company suggested that there are four groups of factors determining the failure of a project. Weight and distribution of these factors in a sample studied by McKinsey & Company is shown in Fig. 9.7. They attempted to translate the causes of project failures into advices on CSF that tell project managers what to focus their attention on in order to avoid a defeat. These attempts are visualized in Fig. 9.8.

The essence of McKinsey's recommendations for achieving IT success is "helping IT and the business to join forces," which means the need for *effective collaboration*, especially between business and IT participants of the project.

As a result, the following question arises—what are the most common mechanisms that constitute fundamental barriers to effective collaboration? Any experienced project manager has his own perception of the mechanisms that may hinder or facilitate effective collaboration in a complex project. The experiences of the author of this book are consistent with the view expressed in a blog entry entitled: "IT Project

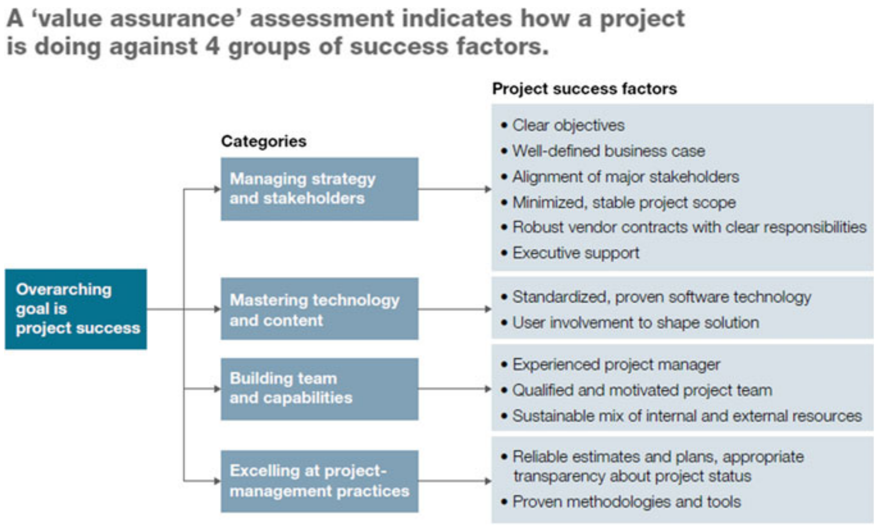


Fig. 9.8 How a project is doing against 4 groups of success factors [54]

Failure: How Did We End Up Here?" [153] by Mike Gammage from 29 October, 2012[7]:

> In my experience, the two biggest barriers to effective collaboration between IT and the business are that:
>
> 1. **They don't speak the same language**. Theoretically, their lingua franca is 'business process'. But, most often, the IT program team adopts a technical language and systems mindset. So the business is forced to adopt swimlanes and other IT constructs. Without the business properly engaged in discussions of the current operational reality and the target operating model, the project is immediately vulnerable to what McKinsey identifies as a common pitfall: 'teams focus disproportionately on technology issues and targets'.
> 2. **Holistic perspectives are lost or ignored**. Most often, the IT systems mindset comes to dominate: if it's not automated, it's secondary. And so the business stakeholders come to focus only on the process fragments linked to system transactions, which limits creative thinking about transformation possibilities and hampers change management. Far too few of those involved, if any, can see the whole. What starts out as a business transformation project enabled by IT slowly degrades to become an IT project with business consequences.

In other words, we came to the obvious theoretical conclusion that **effective collaboration and appropriate application of holistic perspectives** are essential to achieve success in the implementation of complex projects. The above statement sounds obvious. However, in practice, it is very difficult to establish such an effective cooperation and adopt holistic perspectives.

[7]http://sourcing-shangri-la.typepad.com/blog/2012/10/so-who-benefits-from-it-project-failure.html.

9.3 CSE Crisis Case Study: Global Financial Crisis and Risk Management Systems

The vast majority of the population of the Earth have been affected by negative consequences of the Global Financial Crisis.[8]

To better comprehend the perception of emergency of the Global Financial crisis, let us cite the considerations of Alan Greenspan who has a unique experience in the design, implementation, management and development of large-scale financial risk management systems before the Global Financial crisis. When analyzing his opinion, we should take into account the fact that he had the access to very sensitive information.

As a reminder, Alan Greenspan was involved in the management of one of the most complex financial systems in the 20th century for about 20 years. From 1987 to 2006, he was the head of the Federal Reserve System in the United States.

Greenspan compares the Great Depression[9] to the financial crisis of 2008 [177]:

> It was not just the financial crisis—which may very well have been the greatest financial crisis ever. Clearly, the Great Depression of the 1930s was a far more devastating economic problem.
> But what is hard to recognize is that what happened in the hours and 4 days following the Lehman default was a **virtual breakdown of the global financial system**.

In other words, the leading financiers were sure that they used appropriate risk management models. As a consequence, even a few days before the September 15, 2008, they did not expect that the day is going to be the source of a deep and negative crisis [175–177].

Greenspan suggested (in 2008) that if "the models have been fit more appropriately to historic periods of stress, capital requirements would have been much higher and the financial world would be in far better shape today."

Few years later, he wrote a book entitled "The Map and the Territory: Risk, Human Nature, and the Future of Forecasting" [176]. In the book, he proposes an explanation for the failure of economic forecasting to predict (and, in particular, Greenspan's failure to prevent) the 2008 financial crisis. The principal problem that he addresses is that the economic forecasting failed to foresee the financial crisis in 2008, leading to the almost unanticipated crisis of September 2008. He writes that [176]:

> [...] macro-modeling unequivocally failed when it was needed most. Much to the chagrin of the economics profession, the Federal Reserve's highly sophisticated forecasting system did not foresee a recession until the crisis hit.

According to [176], the failure of financial risk management and economic forecasting is not a consequence of feeding the "modern risk models" with data that match historical periods. Rather, it is a problem related to the necessity of changing risk management paradigm of the "modern risk models."

[8] http://en.wikipedia.org/wiki/Financial_crisis_of_2007-08.

[9] http://en.wikipedia.org/wiki/Great_Depression.

In [176], Greenspan emphasizes the emerging stronger influence of behavioral economics.[10] He particularly underlines the importance of the prospect theory, developed by Kahneman and Tversky[11] [263].

Roughly speaking, the main idea of prospect theory is that people do not make real-life rational and optimal decisions. Instead, they make decisions based on the potential value of losses and gains rather than the final outcome. People evaluate these losses and gains using certain heuristic techniques based on intuitions. The book by Kahnemen [263] is based on the idea of a dichotomy between two modes of thought: "System 1" is fast, effortless, frequent, almost automatic, intuitive, instinctive and emotional. According to [263], the mechanism of the "System 1" does not involve rational sense of intentional control, but it is the "secret author of many of the choices and judgments you make" The "System 2" is slower, effortful, infrequent, more rational, and more logical.

Greenspan, on the basis of his extraordinary experience and the analysis of the main causes of the Global Financial crisis, proposed to consider an approach to the construction of computing models, supporting financial forecasting and financial risk management.

His approach uses the concepts of "fast thinking" and "slow thinking" introduced by Kahneman [176, 177, 263].

As Greenspan explains [176]:

> The economics of animal spirits, broadly speaking, covers a wide range of human actions, and overlaps with much of the relatively new discipline of behavioral economics. The point is to substitute a more realistic version of behavior than the model of the wholly rationality-driven "economic man" so prominent for so long in economics courses taught in our universities. This more realistic view of the way people behave in their day-by-day activities in the marketplace traces a path of economic growth that is somewhat lower than would be the case if people were truly "rational" economic actors.
>
> [...]
>
> **From the perspective of a forecaster, the issue is thus not whether behavior is rational but whether it is sufficiently repetitive and systematic to be numerically measured and predicted.**
>
> Can we better identify and measure those quick-reaction judgments on which we tend to base much, if not all, of our rapid-fire financial market and related decisions—"fast thinking," in the words of Daniel Kahneman, a leading behavioral economist? I think so.
>
> THE LONGER PERSPECTIVE
>
> Consider the insights that brought us the steam engine and the electric motor, the railroad, the telegraph, atomic energy, and the integrated circuit. It was those innovations, and more, that over the past two centuries propelled civilization the highest material standards of living ever achieved. They were all the result of human reasoning. As the seventeenth-century French mathematician Blaise Pascal is said to have put it, "Man's greatness lies in his power of thought." It's Kahneman's "slow thinking."

In his analysis, Greenspan emphasizes the fact that the participants of a model system can assume two different roles:

---

[10] http://en.wikipedia.org/wiki/Behavioral_economics.

[11] http://en.wikipedia.org/wiki/Prospect_theory.

i. Active, whereby they can have an influence on the course of a given action and make independent decisions if they want to. In such a case, they use "fast thinking" rather than "slow thinking."
ii. Passive, whereby they have no influence on the course of a given action and cannot make independent decisions (to be more precise, the decision is in fact imposed on them by the situational context).

Initial models stipulated that the role of the market participants should be passive. However, these models differed from the actual market behaviors. To improve their quality, the active role of participants was introduced. At the same time, it was assumed that the decisions of market participants are rational, except for some random, "marginally" made decisions. Greenspan emphasizes that thanks to the analysis of the factors that led to the financial crisis in 2008, he changed his perception of the randomness of these "marginal" decisions, which are often made under the influence of various emotions. He claims that for six decades, he had based his models on the assumption that emotions lead to random decisions. However, after the analysis of factors that contributed to the crisis of 2008, the role of emotions ceased to be perceived as random and became a consistent dependency, conjugated with the law of the market behaviors (e.g., supply and demand). In more simple terms, when the *media value of the market increases*, people are **driven by the animalistic desire to make a profit**. On the other hand, when the *media value decreases*, people are **driven by the animalistic fear**.

All these conclusions were included in the presentation by Greenspan, which is available on the Internet. The presentation was delivered during a meeting of the American Enterprise Institute on January 14, 2014 [177]. Greenspan stated the following:

> **I assume that, over the past six decades, stock market investing, like all economic decisions, has been the result of people at root acting in their long-term self-interest, but with a significant part of human action being biased by deep-seated spirits.**
>
> If, for example, we wake up in the morning and learn that stock price futures have fallen sharply, we are emotionally driven to relieve the anxiety of observing our net worth evaporate, and often sell at, or close to, a stock market bottom. This is a fear-based systematic bias that, more than any other, creates market behavior that can be shown to be out of sync with our objectives of maximum standards of living. **Fear has biased our rational judgment.**
>
> A second illustrative, though hypothetical, example is the case where human emotions and animal spirits play no role in economic outcomes. I choose an example where a person's daily income depends on the sum of a toss of one hundred numbered six-sided unbiased dice. The outcome of any toss must range between 100 and 1200. As much as the dice-thrower may wish, emotions and wishes cannot affect the outcome.
>
> In the first example, where fear induces selling, animal spirits have a large role in determining economic outcomes. In the second example, they have none.

> Classical economic models, for example, either explicitly or implicitly assume that human actions are firmly rooted in rationality, or at least that, **when humans act irrationally, those actions are random and hence have no lasting impact on economic outcomes. That is a view I held until the 2008 crisis induced me to question its generality. In The Map and the Territory, I demonstrate that animal spirits are not random, but in most instances, systematic.**

This important change in Greenspan's approach towards the role of emotions and "animal-like behaviors" made him to suggest including two types of attributes in his model.

One type of attributes deals with the "psychology of a market participant." For example, these attributes are used to measure fear, optimism, pessimism, greediness, appetite for various types of risks, etc. The values of these attributes depend on the construction of a participant, her/his culture and tradition and many more external factors, especially the current market situation. They determine dominant behaviors and decisions of the market participants.

On the other hand, there are attributes which are used to determine the current market situation. They reflect the general laws of the market, which impose certain constraints on the behaviors of the market participants. Therefore, there is a strong dependency between any changes in the values of these two types of attributes.

At this point, it is worth recalling that from the perspective of the CSE Crisis, models supporting the risk management are especially important. It means that if we want to ensure the success of any CSE project, then, theoretically, it is enough to be able to properly manage the risk of its failure.

In other words, by deepening our understanding of the models that support the process of risk management in complex financial systems, we may expect that it will be possible to use some of their mechanisms in other risk management models. Therefore, it is a good idea to become familiarized with the suggestions put forward by Greenspan, concerning models supporting risk management in financial markets. The most important recommendations include his conclusions from the analysis of the factors that led to the Global Financial Crisis [176]:

> [...]
>
> such a model would include a number of variables reflecting those verities of human nature that reveal long-term economic stabilities. Among them are time preference (and interest rates), equity premiums, corporate earnings-price yields, and, since the nineteenth century, the private savings rate. They reflect the outer limits to fear and euphoria that define the dynamics of the business cycle. For forecasting purposes they can be assumed to continue trendless in the future.
>
> In addition, there are those stabilities that are not inbred, such as the sum of social benefits and gross domestic savings as a percent of GDP. Other forecast stabilities include the size of the workforce—those potentially in the workforce have already been born—and average hours worked.
>
> Owing to the vagaries of human nature, forecasting will always be somewhat of a coin toss. But I believe **if we appropriately integrate some of the aspects of animal spirits' systematic behavior, constrained by market forces reflecting the imperatives of double-entry bookkeeping identities, we should importantly improve our forecasting accuracy.** Euphoria will always periodically produce extended bull markets that feed off herd behavior, followed by rapid fear-induced deflation of the consequent bubbles.

Greenspan's conclusions from the lesson entitled "Global Financial crisis and Risk Management Systems for Financial Markets" have enriched our knowledge on the causes and mechanisms of the CSE crisis, including the familiarity with risk management systems. It particularly deepened our understating of the role played by the "human factor" in the above-mentioned causes and mechanisms of the phenomenon, known as the CSE crisis.

Let us recall that in Chap. 16 on POLTAX, we have already strongly emphasized the gigantic role of the "human factor." In a nutshell, a system like POLTAX could not have been developed in Poland between the 80s and 90s of the 20th century, if there had not been mutual trust and enthusiastic atmosphere in a team that was working for the project.

It should be highlighted that the team was being properly motivated to achieve success and its morale was constantly raised. All these factors play a huge role in the development and construction of a new version of the POLTAX system (or its substitute in MF). The system should supersede the current version of POLTAX within the next quarter of a century since its development, construction and implementation at the beginning of the 90s.

At the same time, it should be noted that mutual trust, high morale, good motivation and favorable atmosphere of cooperation are only some aspects of the "human factor" in relation to the CSE projects (including software engineering). Greenspan's considerations point to many important conclusions that enrich our perception of the "human factor" in CSE projects, especially when it comes to systems supporting risk management.

According to the author of this book, **"human factor" is still poorly understood and used by those who are responsible for launching and implementing CSE projects (especially software engineering projects**).

Unfortunately, the "human factor" is very often reduced to personal intrigues within a project team, which leads straight to the so-called "scapegoat hunting," that is finding a person who can be blamed for the failure of a project.

It should be strongly emphasized that the conclusions included in the reflections published by Greenspan are not the only conclusions to be drawn from his lesson on the Global Financial crisis.

There are many other conclusions to be made. Many distinguished specialists from various parts of the world as well as numerous research teams still conduct some research and analyses. Without a doubt, the main obstacle in the development of research in this field is its interdisciplinary nature.

The research does not only encompasses the knowledge of obstacles that hinder the development of finances, economy and financial modelling.

The scope of the research on the development and construction of modern risk management systems also includes research issues that are of utmost importance in such research areas as systems engineering, AI, data mining, semiotics, psychology, linguistics, sociology, economy, logics and software engineering.

Due to the interdisciplinary character of computational systems modelling in risk management, one should not expect that the competence and experience of one man (even with such a considerable experience as Greenspan possesses) will be enough to analyze the whole subject in a satisfactory manner and propose new solutions, encompassing a whole range of various aspects of the techniques used to support the process of risk management.

The author of this book intends to complement the above-mentioned research with the conclusions drawn from his own experiences. According to the author, while constructing computational models supporting risk management, it is necessary to take into consideration many aspects that are not included in the conclusions presented by Greenspan.

For example, if we assume that the concept of "risk" is a complex vague concept, then, during representing and processing it in computational models, we have to apply adequate approximation processing techniques, and characteristic of hierarchical vague concept processing networks.

However, these are not the only additional aspects which, according to the author, should be taken into consideration while constructing computational models for the concept of "risk." Other aspects are discussed in more details in Chapters devoted to the factors that may cause the CSE crisis (cf. Chaps. 11, 12, 13, and 15).

To finish this section, it is worth emphasizing that the Global Financial crisis affected almost every country. As a result, virtually everywhere, attempts have been made to comprehend the factors that lead to economic crises and to develop effective tools that could minimize the risk and protect us from the negative consequences of such crises.

In Europe, the activity of the Basel Committee on Banking Supervision constitutes is an example of such an attempt.[12] One of its outcomes is the Basel III document, which is a complex regulatory framework on bank capital adequacy, stress testing and market liquidity risk. It primarily focuses on the risk of a bank run by requiring differing levels of reserves for different forms of bank deposits and other borrowings. In 2011, the OECD published a report indicating that the implementation of Basel III would lead to a decrease in the economic growth by 0.5 *permille* to 1.5 *permille*.[13] However, some analysts and market observers argue that the recommendations known as the Basel II document contributed to the crisis in the real estate market.[14]

Apart from the aforesaid examples of activities (aimed at identifying the causes of the Global Financial crisis in 2008 as well as risk protection mechanisms) conducted in the USA and Europe, there are also other initiatives that has been undertaken in the USA, Europe and Asia as well as in other continents. **Potentially, every human being suffers the consequences of such crises, which is even more so when it comes to whole countries**.

---

[12] http://en.wikipedia.org/wiki/Basel_Committee_on_Banking_Supervision.

[13] http://www.oecd-ilibrary.org/economics/macroeconomic-impact-of-basel-iii_5kghwnhkkjs8-en.

[14] http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1330417.

Given the enormous scope of these issues, in this book, we will provide only some suggestions regarding the approach towards the construction of computational models supporting risk management (or, in general, models supporting efficiency management) from the perspective of the paradigms of interactive granular computations based on WisTech.

Chapter 10
CSE Theory: Some Approaches

The greatest danger in times of turbulence is not the turbulence; it is to act with yesterday's logic.
Peter Drucker (1909–2005) (http://amiquote.tumblr.com/post/6762585320/)

There are many textbooks which focus on theoretical conclusions drawn from practical observations of many complex engineering projects (e.g., cf. [52, 106, 107, 128, 140, 141, 164, 192, 279, 322, 356, 424, 530, 531]). However, despite being equipped with the knowledge offered by the best books, many complex engineering projects still end in failures. We do not have any common widely acceptable theory of complex systems engineering (particularly, we do not have any widely acceptable theory of CSE).

On the other hand, we do have widely acceptable conclusions regarding the barriers to a successful implementation of CSE projects.

Following the conclusions by Mike Gammage (cf. Chap. 9 and [153]) concerning many CSE projects, in any theory of CSE, we have to propose some efficient methods and tools for the improvement of communication and an appropriate treatment (by all stockholders) of holistic perspectives. In order to fulfill these particular requirements, we should have a good understanding of the essence of the problem. To explain this essence, let us use the metaphor contained in the Indian parable [416], often used to present various abstract complex theories. For example, the metaphor is used in the book [260] for the theory of topoi.

Topoi can be seen as a generalization of abstract concepts. Such as mathematical theories, semantic models of classical and intuitionistic logic, geometric and topological spaces and many other key concepts of modern mathematics. In other words, these are the objects which are the subject of interest for different mathematicians who are looking at them from different perspectives, employing quite different

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_10

Fig. 10.1 Sketches of an elephant

intuitions and associations. In the case of the Indian parable, the metaphorical elephant is a topos and the parable itself begins with the following words:

> Four men, who had been all born blind, wanted to know what an elephant was like; so they asked an elephant-driver for information. He led them to an elephant, and invited them to examine it; so one man touched the elephant's leg, another its trunk, another its tail and the fourth its ear. Then they all attempted to describe the elephant to one another. The first man said "The elephant is like a tree", "No", said the second, "the elephant is like a snake". "Nonsense!", said the third, "the elephant is like a broom". "You are all wrong", said the fourth, "the elephant is like a fan". And so they went on arguing among themselves, while the elephant stood watching them quietly.

This metaphor, in a simplified way, is visualized in Fig. 10.1.

Figure 10.1 shows different observers of the metaphorical elephant. The elephant is perceived by each of them by means of available senses. Each observer seems to associate the animal with an object that is already known to him. The association to a large extent depends on:

a. range and angle of the visual perception of the elephant,
b. senses available to the observer,
c. attributes which can be used by the observer to express his perception.

Somewhat simplifying the case, we can imagine that each observer has his private information system [382, 384, 386], with its own attributes used to describe a given property of the elephant as perceived by him. Depending on the type of the information system, the concept can be expressed in an approximate manner. From the point of view of the rest of this book, let us assume that the information system of

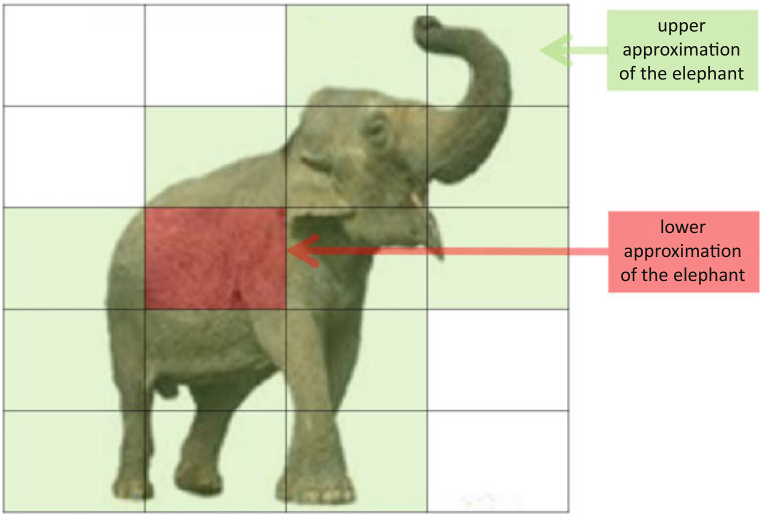


Fig. 10.2 Upper and lower approximation of an elephant

the observer constitutes one or more Pawlak information systems [382, 384, 386], where complex concepts are expressed through the upper and lower approximations and possible relationships between them. Each observer has his own attributes and thus, a unique perception of the upper and lower approximations of the elephant, represented in Fig. 10.2. In Fig. 10.3, we visualize a situation in which each observer has a perception of the elephant based on the upper and lower approximation made by an observer of the information system.

The above-mentioned parable of the elephant gives us a deeper understanding of the role of communication and provides an appropriate perspective of the process of designing and implementing the architecture of a complex system.

The omission of any important aspect in the choice of a perspective or, on the other hand, the imposition of too many perspectives, makes it very difficult or even impossible to communicate and thus, hinders an effective and appropriate collaboration and prevents the project participants from adopting the holistic perspectives of applications.

Taking into account this observation, different centers suggested adopting various perspectives as to the IT systems architecture as well as various methods of organizing and implementing IT projects. One of the first well-known concepts in this respect is the approach published in 1987 by John Zachman, known as the "Zachman Framework" [535] and illustrated in Fig. 10.4.

With the advent of the first architectural framework which defines the recommended perspectives of perception of the IT systems architecture, intensive efforts have been made towards the standardization of the methods used to describe the IT systems architecture, with particular emphasis put on the use of software-intensive systems. Initially, the works were carried out by the IEEE and resulted in the

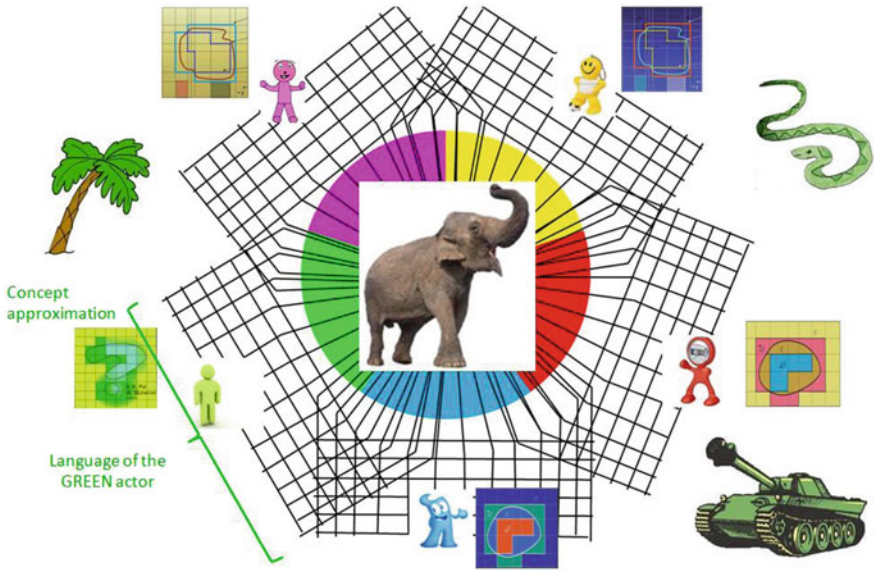


Fig. 10.3 Views of an elephant: each view is based on the Pawlak information system

	What (Data)	**How (Function)**	**Where (Location)**	**Who (People)**	**When (Time)**	**Why (Motivation)**
Scope {contextual} Planner	List of things important to the business	List of the processes that the business performs	List of locations in which the business operates	List of organizations important to the business	List of events/cycles important to the business	List of business goals/ strategies
Enterprise Model {logical} Business Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Fig. 10.4 The Zachman framework for enterprise architecture [535]

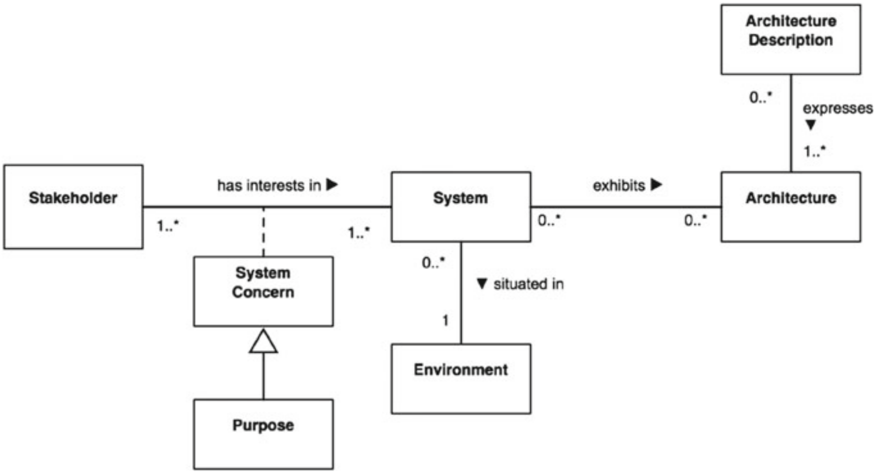


Fig. 10.5 Context of architecture description [221]

creation of the IEEE Std 1471:2000 [221]. The standard became the basis for a broader, international discussion, which gave raise to the ISO/IEC/IEEE 42010:2011 ("Systems and Software Engineering—Architecture Description"). The key concepts of this standard directly refer to the parable of the elephant, that is:

a. **Concern** – interest of many stakeholders in a system,
b. **Architecture View** – work product expressing the architecture of a system from the perspective of specific system concerns,
c. **Architecture Viewpoint** – work product establishing the conventions of the construction, interpretation and use of architecture to encompass specific views for a system in terms of preoccupation and concerns.

These concepts allowed us to specify the ontology language, used to describe the architecture of complex systems. Figure 10.5 shows the context of architecture description, expressed in a language commonly known among IT analysts and used for modeling entity relationship diagrams as well as class diagrams.

Figure 10.5 shows the importance of a relationship between stakeholders, system concerns and system architecture. However, in Fig. 10.6, presenting the conceptual model of an architecture description, one can observe a relationship between architectural concepts and architectural viewpoint.

Over the past decades, a number of architecture frameworks were developed, recommended for various fields of applications and projects of all sizes. In Internet one can find a sample list of approx. 150 such architecture frameworks.[1] For the beginners in computer science, it is very difficult to select a proper architecture framework for a given project. The selection of the architecture framework should be made by an experienced team of people, who truly understand the specificity

[1] http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html.

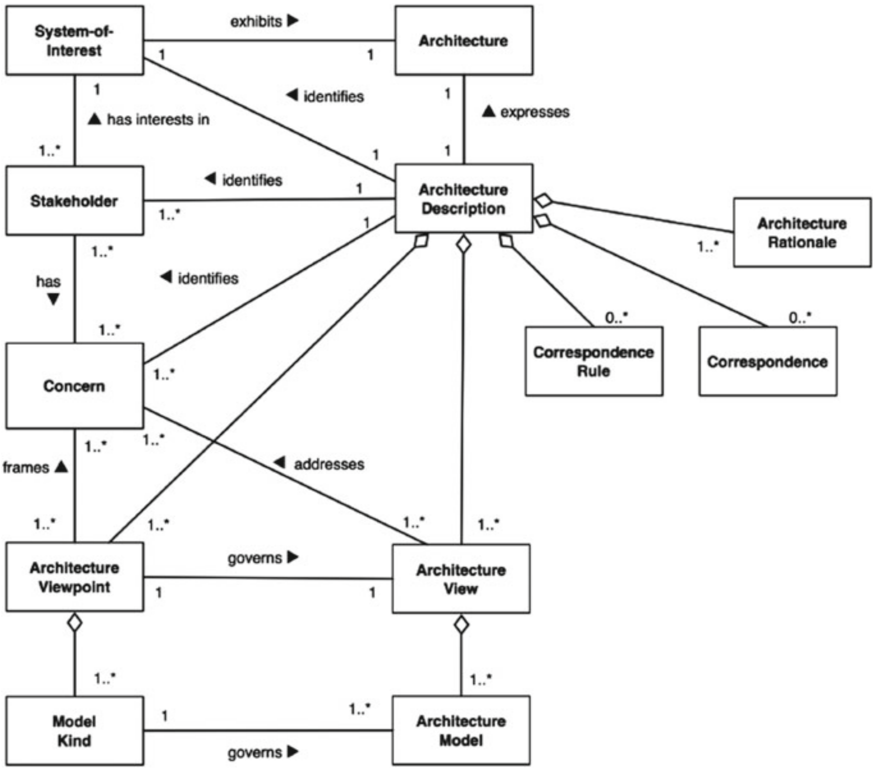


Fig. 10.6 Conceptual model of an architecture description [221]

of the project and the chosen architecture framework. Particular environments are different and of course, promote different architecture frameworks. The proposal by the Open Group, known as TOGAF,[2] is especially popular. Figure 10.7 presents architectural views recommended by TOGAF, while Fig. 10.8 shows an option of a recurring process of development in accordance with the TOGAF architecture, based on the Deming PDCA cycle. As it was mentioned above, there are around 150 architecture frameworks. Of course, the TOGAF architecture is not the only one. Another very interesting framework is the FEAF framework architecture—the U.S. Federal Enterprise Architecture Framework.[3] In this case, we are dealing with a fairly complex architecture framework, which in itself is difficult to control and can be confusing for people who have just begun working with the system. However, it is worthy to note because it concerns highly complex information systems, supporting the functioning of the public administration. Once again, it should be stressed that the solution to the problem of effective collaboration and appropriate applications

[2] http://www.opengroup.org/togaf/.

[3] http://www.whitehouse.gov/omb/e-gov/fea/http://www.whitehouse.gov/omb/e-gov/fea/.

of holistic perspectives significantly depends on the specificity of a given project. In other words, the choice of a too extensive architecture framework can kill the project, overburdening it with administration and formalities. On the other hand, the underestimation of some important perspectives may cause difficulties in the integration of the project into one efficiently and properly working integrated system that is supposed to be flexible enough for predictable and rapidly occurring changes in the environment.

Finding the right trade-off in this respect is the most important task at the first stages of the project's implementation. This should be based on both the extensive experience of those involved in the project and sufficiently flexible selection of the prospects for the perception of a project architecture. Moreover, one should take into account the language used to describe the selected prospects and mechanisms of interaction, including communication, to boost the efficiency of the collaboration.

As shown by the research projects carried out over the last several decades, there is still a very large gap between the theory and practice of implementing complex systems. This means that as of today, there is still no "silver bullet" that could solve the aforesaid problem and, moreover, there is no universal and effective solution to the problem of effective collaboration and appropriate applications of holistic perspectives.

The above architecture frameworks and architecture description standards are criticized by some practitioners due to their high degree of formalization. Moreover, some practitioners even undermine the conclusions of numerous studies that led to the emergence of these standards. It is particularly worth noting that not everyone agrees with the methods of analysis and conclusions used and found in the above-cited reports.

For example, in [204], it is strongly suggested that the major problems associated to the implementation of complex projects are related with the difficulties in measurement and reasoning (in particular in planning) features of CSE projects.

> The Standish data are NOT a good indicator of poor software development performance. However, they ARE an indicator of systemic failure of our planning and measurement processes. [204], (pp. 335–358)

According to this criticism, the key to success in complex engineering lies in the ability to perform measurement and reason about the plans and changes that may occur during the project and create some additional risks that cannot be foreseen at the initial stages of the project.

This ability should result in the creation of a credible plan and the mechanism of change, which will minimize the negative effects of any changes to the project. The difficulty in performing such a reasoning is emphasized by Lotfi Zadeh and Judea Pearl in the motto of the book.

To illustrate this difficulty, let us take an example of the notion used by the head of the Standish Group—Jim Johnson—and presented at the XP 2002 Conference[4] [256]

[4]http://martinfowler.com/articles/xp2002.html.

Fig. 10.7 The TOGAF architecture views [373]

to summarize the statistics and discuss the limited extent to which the functionalities that were once implemented are actually used. A summary of these statistics, performed by Jim Johnson, is included in Fig. 10.9.

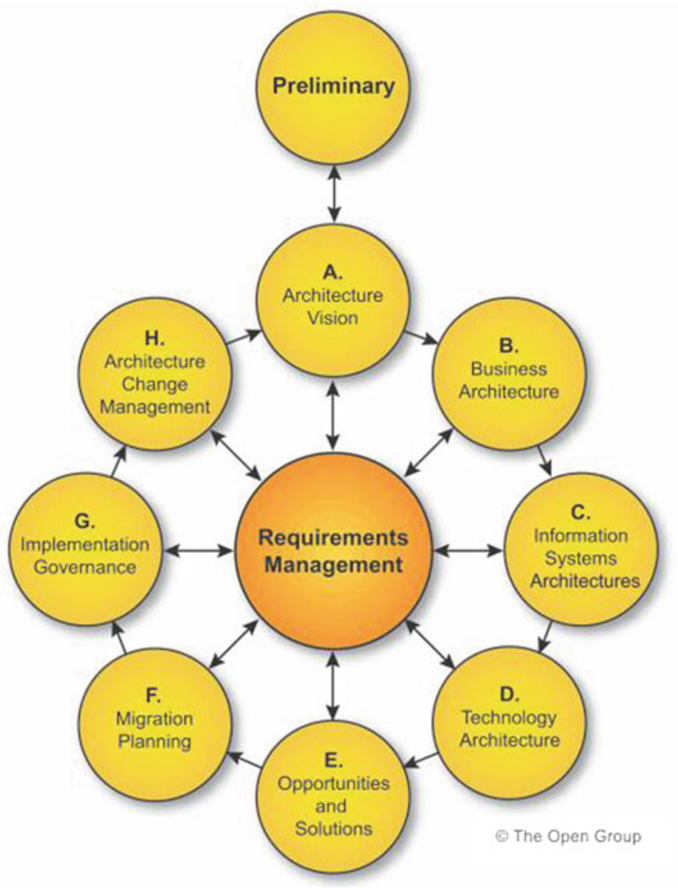


Fig. 10.8 A recurring process of development in accordance with the TOGAF architecture, based on the Deming PDCA cycle [373]

Anyone who has ever used both simple and very complex software is well aware of the fact that regardless of whether he (she) works in MS Office or SAS Enterprise Miner, he (she) effectively uses only a small percentage of the functionalities offered by this environment. Of course, from a psychological and marketing point of view, it is good to have an infinite number of functionalities, even if they are never used. However, the brutal statistics, shown in Fig. 10.9, prove how much you can speed up the implementation of projects through a proper understanding of what is being implemented.

This idea has become one of the main pillars of the paradigm shift in the construction and development of software that initially, in the 80s, was called rapid prototyping. Later experience with rapid prototyping has been expanded to encompass a number of important changes in the classical approach to planning and management

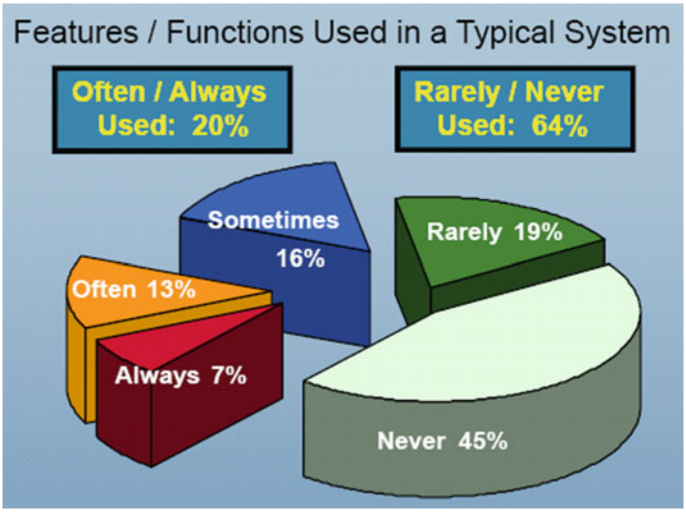


Fig. 10.9 Standish group study reported at XP2002 by Jim Johnson, Chairman [256]

and was named as Agile Software Development. Nowadays, there are two fundamental documents which form the so-called constitution of the agile approach to software development. The first one is "Agile Manifesto"[5]:

> We are uncovering better ways of developing
> software by doing it and helping others do it.
> Through this work we have come to value:
> Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan
> That is, while there is value in the items on
> the right, we value the items on the left more.

The second basic document is called "PM Declaration of Interdependence"[6]:

We [...]

- **increase return on investment** by making a continuous flow of our focus value.
- **deliver reliable results** by engaging customers in frequent interactions and shared ownership.
- **expect uncertainty** and manage for it through iterations, anticipation and adaptation.
- **unleash creativity and innovation** by recognizing that individuals are the ultimate source of value and creating an environment where they can make a difference.

[5] http://agilemanifesto.org/.

[6] http://en.wikipedia.org/wiki/PM_Declaration_of_Interdependence.

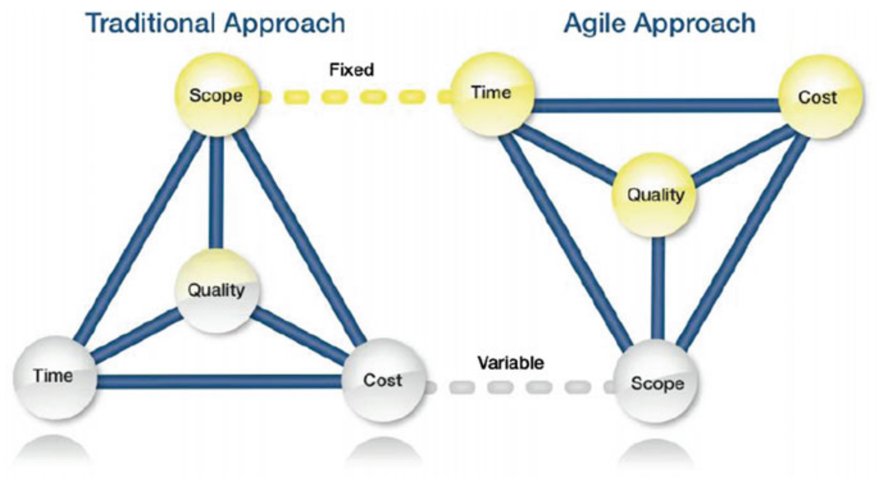


Fig. 10.10 Planning in the Agile Software Development is significantly different from the traditional approach [93]

- **boost performance** through group accountability for results and shared responsibility for team effectiveness.
- **improve effectiveness and reliability** through situationally specific strategies, processes and practices.

The approach to planning based on the Agile Software Development is significantly different from the traditional approach. To explain the main difference, let us use Fig. 10.10, prepared by DSDM Consortium in the paper entitled "The DSDM Agile Project Framework for Scrum White Paper" [93]. According to the description in the paper:

> In the traditional approach to project management (shown on the left of the diagram above Fig. 10.10) the feature content of the solution is fixed whilst time and cost are subject to variation.
> If the project goes off track, more resources are often added (which varies the cost) and/or the delivery date extended (which varies the time). However, adding resources to a late project often makes it even later and a missed deadline can be disastrous from a business perspective and often damages credibility. Under such pressure, quality often becomes a variable, as a result of introducing compromises which have not been thought through, by reducing essential quality control steps or by cutting back on testing.

The difference between Agile and the traditional approach to project planning is outlined from a slightly different perspective (vs. Driven Plan. Value/Vision Driven) in Fig. 10.11[7] [473].

During the first years after the introduction of a new approach to project management, Agile gradually gained more and more popularity. However, in retrospect, one

[7] http://newtechusa.net/wp-content/uploads/2012/06/gathering-agile-requirements-dan-lefebvre-060712.pdf.

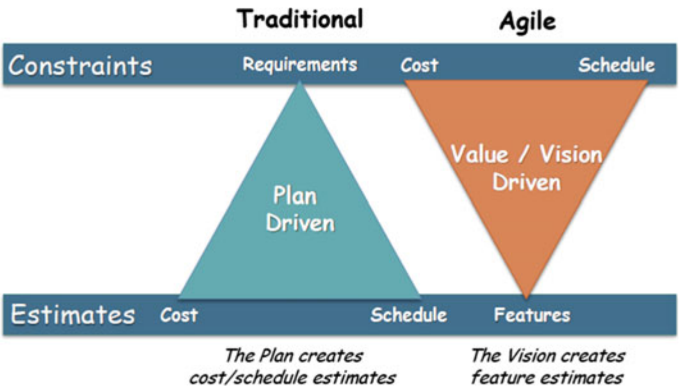


Fig. 10.11 Agile flips the triangle [473]

has to admit that it also met with criticism when it came to working on large complex projects in large organizations, and often faced with the problem that the maintenance of the system requires making changes to the software. If software is produced without sufficient documentation and the code generated is written by different teams in different standards, the cost of implementing significant changes in this situation is comparable or even exceeds the cost of writing the appropriate code fragments from scratch. Thus, a search for a suitable compromise between the formalization of the project, generally espoused by the traditional approach to project management, and its “lightness” in the approach to the compliance with design standards, especially for producing documentation solutions. Very often, when determining this trade-off, people responsible for selecting a proper project management methodology for a specific project, use the three-dimensional positioning method which include the following.

1. People: This dimension consists of components such as the knowledge and skills of people involved in a project, organizational culture, work habits and various measurable aspects such as the number of people in a team, the diversity of people in a team, etc.
2. Requirements: From near certain, with no risk of change, to completely uncertain, with vague, emergent descriptions and many foreseeable changes. In particular, we deal with a completely different situation in the case of innovative R&D projects and projects which have to produce a very simple and stable functionality.
3. Technology: It could be either well-known, already applied in many projects and understood by the project team, or completely new and uncertain.

These types of diagrams are called Stacey Graphs. An illustrative example of such a graph is shown in Fig. 10.12 (cf. Sect. 1, Chap. 1: The Crisis in Software: The Wrong Process Produces the Wrong Results in [479, 489]).

If a project is implemented under stable and well-known constraints, with definite requirements and technology, is based on a well-chosen competent team, and has

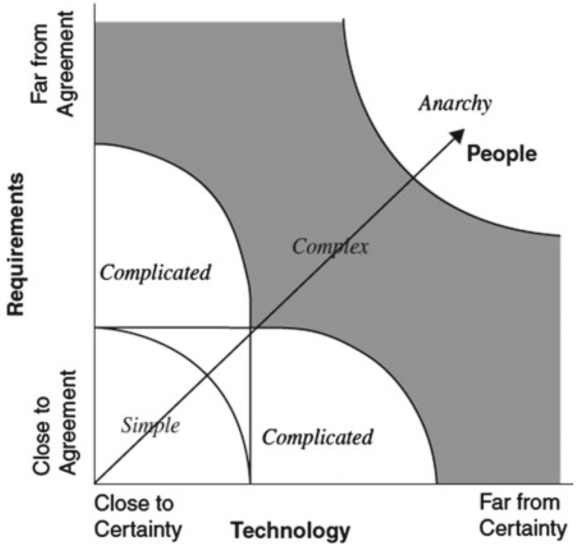


Fig. 10.12 Illustrative example of Stacey Graph [489]

appropriate resources, then essentially, the risk of failure is not high. However, for the design and construction of large complex systems, such a situation is very rare. Therefore, it is necessary to deal with problems arising from the uncertainty of the requirements, the lack of recognized technology, and other problems caused by the complexity of interactions that take place in the case of large project teams as well as problems arising from the need to replace a part of the team during the project's implementation.

Generally, in such cases, these types of challenges mean that we have a very big gap between the theoretical assumptions of the project and its practical implementation. This applies to all methods of project management and does not depend on whether we are in favor of the traditional approach to management, Agile or any other. Each project is unique, just as every human being in a team has her/his own individual properties.

A very big challenge is to apply universal methods and standards that fit all people in a team and match the requirements and technology used in a project. Luckily, over many decades of implementing complex projects, standards for accumulating CSE-related knowledge and experience in the form of valuable guidelines have been established for those responsible for the selection of appropriate methods and their adaptation to the specificity of a project.

Basically, according to these standards, it is not possible for all the participants to use the same language and be able to understand each other in practice. However, it is not just about this simple observation. It is not just the lack of communication or mutual understanding between the end users of a given product and those involved in the production of this product. This is a much more complex problem related to a shift in user's requirements and the way they are understood during the implementation of a project.

It turns out that in the case of complex projects, teams, or even functional groups, involved in various aspects of a project are not able to freely communicate with one another and often use the language and words that are sometimes misunderstood. Taking into account the observation by Brooks [68] about the square-increasing number of communication channels in relation to the number of team members, it is clear that if partners use different languages in these communication channels and assume that they understand each other, the chances for the success of such a project are very low. This observation regarding the problems with communication in the interaction between different groups of project's stakeholders is fundamental in the currently prevailing standards, aimed at adapting methods of the project's implementation to the project's specificity.

It is astonishing that throughout the whole history of managing the development of complex IT systems, the percentage of failed projects is almost constant, especially for large projects. Many reports published in recent years by various independent consultants confirm that the Agile Software Development is not a "silver bullet" for solving the problems of complex IT systems. For example, according to 2015 Chaos Report by Standish Group [257], Agile approach is quite successful for small size project (cf. Fig. 9.2). However this approach is not similarly successful for large size projects (cf. Fig. 9.2).

Of course, it has been proved that the small projects implemented by relatively small teams, under the stable constraints, stand a much better chance of success. However, large innovative complex projects are still exposed to a high risk of failure.

Summing up, let us recall that CSE encompasses virtually all forms of complex human activity. These activities are by no means limited only to software engineering, on which we focused our attention in this Chapter.

Each area of CSE application is characterized by specific knowledge, typical of a particular domain.

For example, the design, construction and development of macroeconomic financial systems have their own terminology and specificity, which is different from the terminology and specificity of the risk management of typical CSE projects. Nevertheless, there are numerous significant mechanisms that are common for both these domains.

Chapter 11
TPGP: The Concept of the Theory—Practice Gap Problem

The only constant in life is change.
Heraclitus of Ephesus (c. 535 BC–475 BC) (http://www.ancient.eu/Heraclitus_of_Ephesos/)

CSE projects are usually implemented in an unpredictable and continuously changing environment. Hence, very complex activities associated with the CSE projects (i.e., modeling, analysis, designing, building, testing, integration, acceptance, maintaining, development and project management) carry numerous risks. Often, they lead to the failure of a project (cf. Chaps. 1 and 9). Thus, for many CSE projects, we can find gaps between:

1. Practical achievements, and
2. Theoretical expectations stemming from engineering knowledge and constrains that have been already discovered.

This stimulates a very high demand for the development of effective techniques, aimed at minimizing the risk of failure for CSE projects. Intuitively, the essence of the problem can be expressed as follows:

Bridging Theory and Practice:

The *Theory—Practice Gap Problem (TPGP)*
Why we do not have satisfactory techniques for essential reducing the gap between theory and practice of CSE projects?
What should we do in addition to the current state of knowledge and technology to essentially increase the chances for a successful implementation of CSE projects?

By intuition, the essence of TPGP is searching for the most effective tools that can either minimize the risk of CSE project failure and/or, in case there is such a risk, minimize its negative consequences.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_11

The TPGP problem should be analyzed from a very broad perspective, which means taking into account any type of CSE project. It particularly applies to complex systems for enterprises, markets, financial systems, economic systems, political systems, ecosystems and management systems (e.g., risk, safety, quality, human resources).

It is worth noting that the TPGP problem is generally related to the process of producing new complex systems.

To a lesser extent, this concerns projects which make use the already verified schemes of activities in stable and well-recognized environments. In such environments, there is little chance for unexpected changes to occur and jeopardize the implementation of a project. The implementation of projects in such stable and well-recognized environments primarily consists in learning how to implement repetitive behavioral patterns in the most optimal way possible. Definitely, the learning and optimization processes can also be very complex.

Nevertheless, finding a general solution to the TPGP problem of building innovative new complex systems, especially in unstable and unpredictable environments, causes us the biggest problems. In a sense, dealing with the TPGP problem in such difficult environments, also from a theoretical point of view, can be seen as a process of discovering the changes in the environments and acquiring specific knowledge about these changes, that can be used for the adaptation of the already known solutions.

However, this can be very difficult and may require a great deal of creativity. It often happens that creativity can indeed be significantly constrained by our knowledge, experience and paradigms that are being used. In this process, it seems to be particularly difficult to find an appropriate terminology, language, and CSE principles (expressible in a language that would allow for a flexible adaptation of the already known solutions by the teams of people working on a given project).

One of the fundamental paradoxes related to the TPGP problem is that in spite of an extremely fast-growing industries of science and technology, as well as the rapid growth of the practical experience in the implementation of complex systems, a significant part of these projects end far beyond what has been expected.

For example, in the case of large complex IT projects—as reported by various sources—in the last decades, each year, an essential percentages of the projects ended contrary to the target goals and significantly beyond the pre-assumed use of resources.

While analyzing this surprising phenomenon, it is worth noting that in the successive years of the last decades, projects were getting bigger and more complex and thus, one can imagine that the nominal value of the negative consequences connected with the failures of CSE projects was steadily increasing. This brings us to formulate the following *CSE complexity paradox*:

> The failure costs of CSE projects are not decreasing despite the rapid growth of practical experience and theoretical knowledge on techniques for avoiding these failure costs.

It should be strongly emphasized that the complexity paradox is not limited to the field of IT. There are numerous examples of this phenomenon in many other areas (e.g., financial systems, economic systems, political systems, risk management, safety management).

A striking example illustrating the complexity paradox is the implementation of large-scale projects aimed at building financial risk management systems (cf. Chap. 9).

On one hand, at the end of the 20th century, the dominating belief was that practical experience and knowledge in this area are very large, valuable, and useful. Many complex mathematical theories for supporting the risk management of financial systems were developed. On the other hand, the systems failed to effectively predict and prevent many subsequent waves of financial crises, especially at the beginning of the 21st century.

Chapter 12
CSE Crisis: Some Examples of Causes

In theory, practice is simple.
Trygve M. H. Reenskaug

12.1 Causes of CSE Crisis: Communication and Vague Requirements

The problems with communication and understanding user requirements are illustrated by the wit, which has been known for many decades and which is shown in Fig. 12.1.

Based on a preliminary, superficial analysis, one of the most important causes of the CSE crisis may be the problem with communication between project participants. It stems from the fact that different teams taking part in a project use locally specific languages and conceptual frameworks, which are not properly understood by other participants. Moreover, these locally used concepts may change in meaning over time.

Hence, a metaphorical *CSE Babel Tower*, which changes with time, has been created for CSE projects. It is worth noting that as the tower develop, the role and responsibilities of CSE translators from one local language to another consistently expand.

In general, the role of *CSE translators* is usually underestimated or grossly overlooked in CSE projects. However, in order to ensure a sound cooperation, it is crucial to express the task completion plans in languages understood by the majority of local groups consisting of project participants. Also, it should be remembered that the specification of a task completion plan in a particular language may hinder the very completion of that task, as the method of task completion may depend directly on the concepts used in this language. The lack of languages specifically customized to problem-solving makes it very difficult or even impossible to provide appropriate

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_12

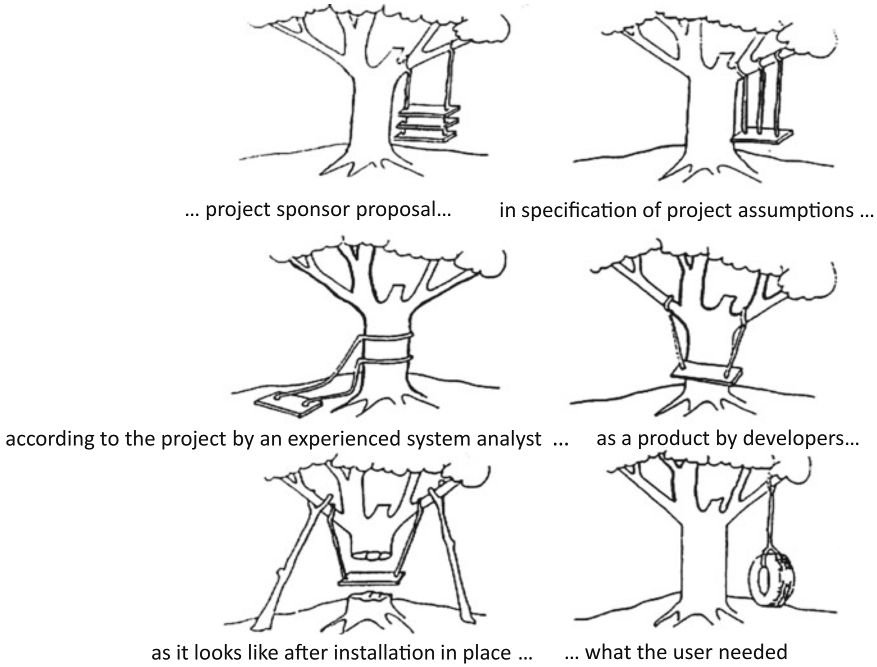


Fig. 12.1 Illustrative wit: consequences of the lack of communication and problems with mutual understanding among different project participants

solutions. Intuitively speaking, if we do not have the concepts of a "wheel" and a "cart" in a language that we use to define transportation tasks, we may end up having them defined as tasks that consist in moving heavy items rather than building a cart to carry them. Therefore, the absence of a common—that is defined by all project participants—appropriate conceptual apparatus and the language of communication during the implementation of a given project constitutes a significant problem in the design of complex systems. The language of communication should support the specification of both the objectives and constraints for the completion of a project (including architectural decisions), along with the specifications of task completion techniques and plans for the acceptance of complex projects. These specifications usually employ complex vague concepts which change their meaning during the life of a project.

Thus, it often happens that the concepts (**especially project requirements**) become understood by the project participants long after the project has been finished. Roughly speaking by *project requirements*, we mean specifications of requirements for the following components of a project:

1. Final products and services.
2. Tasks, deliverables and relationships among them.
3. Resources.

4. Assignment of resources to tasks.
5. Milestones and acceptance criteria.
6. Time schedule.
7. External project constrains.
8. Management.

In practice, failure to understand and communicate project requirements usually means the failure of a project.

Therefore, the phenomenon of in-project communication (especially project requirements) has led to the identification of a *candidate 1 for CSE crisis precause*, which can be formulated as follows:

UNSATISFACTORY SPECIFICATION AND UNSATISFACTORY UNDERSTANDING OF PROJECT REQUIREMENTS:
in almost every new CSE project it is impossible to develop (on-time), at a satisfactory level of details (for a particular non-trivial phase of the project), project requirements specification for a particular phase.

Somewhat simplistically speaking, and to paraphrase the Yourdon's book [545], if we do not know the destination of the march (that is, the final requirements of the project's products), we should not be surprised that we cannot find the right way to that destination (i.e., it is not possible to construct a model for a project implementation that would yield the final products as has been expected).

This surprisingly pessimistic—for CSE novices—conclusion arises from a number of reasons. First of all, let us remind that by a *CSE project*, we mean an innovative project, leading to the construction of a complex system. The CSE project is implemented in a dynamic and uncertain environment. Some justification for the selection of the candidate 1 for the CSE crisis precause can be as follows:

1. Before the beginning of any non-trivial phase of the CSE project, it is necessary to specify the requirements that should govern the implementation of a project during each such phase (to satisfactory level all of details). Usually, the project requirements specification is a process aiming at approximation of these specifications (to a satisfactory degree for the particular phase of a project). This is realized by iterative experiments and analysis in the PDCA cycle, implemented during the phase.
2. During the implementation of a typical CSE project, both the concepts defining the requirements as well as the requirements themselves are subject to change and development.
3. Any specification of the requirements for the final products and services in CSE contains important fragments expressed in a language of complex vague concepts or their temporary approximation in a 'precise' (formalized) language. As a consequence, during the implementation of a project, we often do not have the exact specifications of the requirements.

4. Usually, different contributors and end-users of the products understand and interpret requirement approximations quite differently. These discrepancies are revealed with a significant delay. The repair costs of these errors dramatically increase (often exponentially) when the implementation progresses.
5. Usually, during the implementation of a typical CSE project, new non-trivial attributes for improving the requirements approximations are revealed. In general, discovering any new attributes and approximations thereon requires costly changes to the project.
6. Usually, the very first specifications of the final product requirements and services prove unprofessional, contain many errors and misunderstandings. In consequence, with the project leaders clinging to these requirements, the project becomes more virtual and already at the beginning, its chances of success are gravely diminished.

The above points do not cover all arguments that could explain that in general, it is not possible to create requirement specifications in advance in such a way that they comply with the final products and services of a CSE project (on a satisfactory level of details, specific to a given phase of the project's completion).

In such a situation, the question that immediately comes to mind is how it is possible that so many CSE projects still end up a success. This is due to the fact that experienced project managers fully realize that the project requirements (in particular the specification requirements for the final products and services) are results of interactive processes, which often lasts much longer than a project itself (i.e., long time after the end of a project). Therefore, the products of a given project should be configurable within a predictable range of changes to the requirements. For example, any new version of the new software operating system (e.g., Windows) is subject to additional changes during its exploration by the end-users.

With respect to certain types of projects, many techniques for requirement management were developed. These techniques are generally based on the construction and evaluation of subsequent prototype versions, with a particular focus on the identification of potential areas of requirements, where changes may occur. These prototypes are built by combining many local implementation models, dedicated to a given project. This situation may be complicated by dynamic changes induced by mutual interactions between local implementation models and the environment.

It is difficult to come up with a non-trivial description of the aforesaid situation, not to mention the features of the interaction between the local implementation models of a given project. Properly implemented reasoning of this kind should allow for the identification of particularly important areas of uncertainty in terms of further changes to the requirements concerning the project's final products. Then, the reasoning concerning the interactions occurring between aggregations and local implementation models should help identify sets of parameters, which control the reconfiguration of the final products. In this way, we can brace ourselves for surprises, which may bring about the changes to end-user requirements at the end of a project, as well as many years after its completion.

For example, when designing—in the early 1990s—the complex data model for the POLTAX project (under the direction of the author of this book), main trends of changes in similar tax systems in the world were analyzed and on this basis, a very flexible data model was designed and implemented. This model has been providing the POLTAX system with a flexibility of operation till this day, which is at least a quarter of a century of unforeseen changes in the Polish tax system.

Other techniques for dealing with uncertainty in requirements are described by the author in Part IV of this book. Unfortunately, the currently available approaches to the problem of fully understanding the requirements do not provide consistent tools to deal with it. As long as we are dealing with "analogical" projects, it is easier (but this does not guarantee success) to construct "analogical" implementation models for complex projects, such as CAS, which are aggregations of many changing local models.

However, in the case of new projects (in particular concerning the development of an innovative technology), it is necessary to keep detecting (inducing, constructing, adapting and archiving) both local models and their aggregation, based on the dynamically changing and partially available large data sets, as well as diversified domain knowledge. This is a highly non-trivial task which often hinders the implementation of such projects.

The world of interaction between physical objects and humans is highly complex and unpredictable in the case of CSE projects. In practice, the implementation of each complex and unique project calls for changes on a daily basis. These changes—during the initial phases of a project—are usually unobtrusive and/or ignored by the project management. As a result, they lead to a systematic and cumulative erosion of a project and having reached a certain critical level, force project managers to choose one of the two possible directions:

1. To stop a project and restart it later,
2. To virtualize a project, for which a skillful Public Relations (PR) management with stake holders is needed to explain to all concerned that "nothing has happened" and "that's how it works."

The above considerations lead to the next, surprising for novice CSE project managers, conclusion that the global implementation models for CSE projects are so complex and ambiguous, with regard to the problems in understanding the requirements, that they are beyond the intellectual grasp of any individual. In other words,

one of the consequences of the problem of trying to understand the requirements is that the global implementation models for complex projects are beyond the reach of the one human mind.

It would be a sign of unrealistic optimism to count on a group of super-experts from a super-company to develop a sufficiently accurate global model for the implementation of a complex project, consistent with the analyses carried out using super-modeling and analysis techniques (for example, by using some equations and/or rules discovered by the company) and on this basis assign well-defined tasks that are to be completed. This is particularly improbable in the case of real—especially unique—CSE projects. If there was such a super-group of experts somewhere in the world, they would have to understand the basic concepts used in the construction of implementation models for complex projects, such as the project risk or design requirements, related with the project risk. According to the author, it is impossible to construct reasonable approximations of these concepts in such a way as to establish a global implementation model for a unique complex project, at the level of details sufficient to all uniquely defined tasks that fall within the scope of CSE projects.

This derives from the fact that the number of attributes needed for these approximations is generally unknown prior to the implementation of a project and besides, in practice—after an initial list of such attributes has been identified—it turns out that there are so many of them, that it is impossible to control them even by the well-cooperating groups of humans and computers.

Therefore, the only thing that can be done is the **interactive "granularization" of a given model into smaller local approximate models, which represent local and current aspects of the project's implementation at a given time**. Then, appropriate project teams are tasked with aggregating these local models into one large integrated model. Given the fact that this aggregation is often carried out by people who do not understand such local models, as well as the fact that people working on different local models speak different local dialects, it is evident that the integration of local models into a coherent whole may prove to be a very difficult challenge.

It is even more difficult in practice as we do not implement local models, but only their approximations, which must be adapted as quickly as possible to the changing circumstances of a project and to our instantaneous perception of the global approximation of this very project. In order to avoid the erosion of a project (already described above), we need to introduce all the necessary alteration as soon as possible. While making decisions to introduce such changes in the implementation of a project, we should maintain a balanced perception of the interaction between the local models of current global requirements for introducing changes (resulting from the approximation of a global model) and the local models of local requirements (resulting from the approximation of local models). In this way, we obtain the following conclusion:

an important consequence of the absence of the satisfactory requirements of projects, implemented in an unpredictable and dynamically changing environment, is the impossibility to construct and deliver "on time" to relevant project participants a global model for the implementation of a project, which will be precise to such an extent as to allow for the definition and assignment of well-defined tasks to particular project participants.

Therefore, also in this case, we have one argument which indicates that there is a need to build local models, tailored to the ability of project participants to understand and perform particular tasks and, at the same time, customized to locally understood current adaptation requirements of a project. Then, the local aggregations of models, remaining in very complex interactions with each other and with the environment, are required.

Both the number of the local implementation models for CSE projects and the complexity of the aggregation process—along with the frequently huge data and knowledge resources required for the implementation of such a project—mean that the creation of models for CAS (including the implementation of complex projects) poses a great challenge. It is a special case of an even broader challenge, which is to understand the nature of interactive computing, implemented by autonomous systems in an unpredictable environment. The degree of difficulty of these challenges is best illustrated by the examples of "adaptive complex systems," such as big economic organisms adapting to changes in their environment, cooperating communities of humans and artificial agents, or any living organism which can be treated as an example of a complex adaptive system performing interactive computing.

12.2 Causes of CSE Crisis: Partial Perception of Multidimensional Dynamic Complexity

Numerous CSE experts believe that the main challenges with the implementation of CSE projects stem from the complexity of these projects as well as the rare recurrence of their key components. The complexity of projects applies to all more significant dimensions of their implementation. At the same time, the complexity of CSE projects is one of the oldest perspectives through which to approach the CSE crisis. Much thought has been devoted to the subject in the book by Brooks [68]. In fact, when referring to software engineering projects, Brooks writes that "the complexity of software is an essential property." This property was defined by F. Brooks in the following fragment of his work [68] (crucial fragments were bolded by the author):

> Complexity. **Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level)**. If they are, we make the two similar parts into one, a subroutine, open or closed. In this respect software systems differ profoundly from computers, buildings, or automobiles, where repeated elements abound.
>
> Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more states than computers do.
>
> Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some nonlinear fashion, and the complexity of the whole increases much more than linearly.
>
> **The complexity of software is an essential property**, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.

The above conclusion by Brooks can be relatively easily generalized to encompass much more dimensions of CSE projects. For example, it is enough to substitute "software entities" for "models of innovative components of the CSE project" in the above fragment.

To put it simply, the complexity of CSE projects is so great and ubiquitous that it goes beyond the perceptive capabilities of small groups of people. What is more, various aspects of knowledge, that is key to a given project, are out of reach for the contemporary science. Therefore, to implement a project, it is necessary to discover new scientific knowledge.

For this, one needs to be able to skilfully plan the interactions and experiments that are to take place during the implementation of a project. If we want to develop a CSE project in a consistent manner, we have to accept the fact that at the same time, we will have to expand our domain knowledge, which is needed for a project to make any progress. To paraphrase Brooks, **when it comes to CSE, the times of an "easy" application of knowledge, that has been acquired over the course of the last three centuries "by constructing simplified models of complex phenomena, deriving, properties from the models, and verifying those properties experimentally," is now over** [68].

Regardless of the above paraphrase of Brooks that results from the "partial perception of multidimensional dynamic complexity," it should be emphasized that in the currently and commonly used mathematics, there are numerous traps awaiting the CSE novices, who are not experts in the fundamentals of contemporary mathematics. Some of these traps will be illustrated in the next Section.

12.3 Causes of CSE Crisis: Pitfalls of Thinking Using "Modern" Mathematics

CSE is heavily based on modern technology which utilizes the language of contemporary mathematics. This means that the performance and effectiveness of tools employed by Complex Systems Engineering depends directly on the performance and effectiveness of its own tools, derived from mathematical languages.

Naturally, the current state of mathematics is the effect of its development throughout history. To paraphrase Heraclitus, "there is nothing in science permanent except change." Consequently, we can be sure that the fundamentals as well as the whole apparatus of mathematics will be changed in the future. Obviously, we do not know yet in which directions these changes will go. As shown by the example of applying statistical techniques to risk modelling, as well as many other similar mathematical applications, the tools provided by modern mathematics do not guarantee an effective management of such important aspects of CSE as:

1. Dealing with complex and vague concepts (such as "risk", "user requirement", etc.), hence the need to employ vague concepts under uncertainty and to conduct reasoning about vague concepts which may change over time.

2. An open complex world in which complex systems are to be constructed and deployed and the need to take into account often unpredictable effects of environmental influences on these systems.

Modern mathematics is based primarily on the paradigm of the Cantorian set theory ontology [292]. The consequences are as follows:

1. The reliance of modern mathematics on “crisp” concepts and not on “vague” concepts.
2. Mathematical models are constructed using static, often non-constructive sets with highly counter-intuitive features, foreign to the practice of engineering. These sets exist in a “platonic” closed world, impermeable to external interactions.

Therefore, the tools of modern mathematics do not facilitate the management of the aforementioned important aspects of CSE. In this context, we need to think of prospective deeper problems and their possible solutions.

At the turn of the 20th century mathematics was looked at in terms of the ontology of set theory. The ontology is based on the concept of a set and the set membership relation.

A quite popular conviction is that these sets and entities constructed thereon (e.g., the empty set, a set containing only the empty set, the set of all subsets, …, a class of sets belonging to the von Neumann universe[1] exist beyond the reality of a physical space-time. They are static in nature (not subject to change over time) and moreover, they are not influenced by phenomena occurring in the physical world around us.

Other fundamental concepts include binary relations, understood as the static sets of sequences and functions representing a special type of relations. This metaphysical reality is ideal for the Turing machines[2] which, according to the Church-Turing thesis[3] model all functions algorithmically computable in an intuitive sense. Under this assumption, the Church-Turing thesis does influence our understanding of the modern fundamentals of computational models in IT.

The set theory ontology is essentially fundamental to the entire field of contemporary mathematics, in all forms of its application. Taking into consideration the fact that modern mathematics is regarded as a descriptive and modelling vehicle for the entire natural science and technology, one can realize how important it was to introduce set theory and thus, lay the foundation for the ontology of modern mathematics.

This decision was made as a result of the research, carried out by George Cantor and others, on the concept of infinity expressed by the concept of a set. This research, in a sense, referred to a number of previous attempts (known since the Antiquity) to encompass and describe the activities leading to infinity and the concept of infinity itself (e.g., Aristotle, Anaximander, Zeno of Elea). The key concept in the approach

[1] http://en.wikipedia.org/wiki/Von_Neumann_universe.

[2] http://en.wikipedia.org/wiki/Turing_machine.

[3] http://en.wikipedia.org/wiki/Church-Turing_thesis.

by Cantor is the concept of a set. The intuition of infinity is conveyed by Cantor as follows [83]:

> A set is a gathering together into a whole of definite, distinct objects of our perception (Anschauung) or of our thought which are called elements of the set.

It is worth emphasizing that this seemingly intuitive-looking definition and Cantor's conclusions drawn from set theory have led to some significant shifts in the way we understand the properties of infinity. Cantor's findings were completely different from the views of many prominent mathematicians before him. A prime example of this are the following words by Carl Friedrich Gauss, considered one of the most brilliant mathematicians, from 1831 (in letter to Schumacher, 12 July 1831 [156]):

> I protest against the use of infinite magnitude as something completed, which is never permissible in mathematics. Infinity is merely a way of speaking, the true meaning being a limit which certain ratios approach indefinitely close, while others are permitted to increase without restriction.

There was an interesting discussion among researchers who tried to discover the role of modern mathematics in the development of science in order to better understand the essentials of an advancement in mathematics, and how it influences science, within the context of scientific revolutions proposed by Thomas S. Kuhn in [291].

Within this context, it is important to understand and identify the characteristics of the turning points in the history of "The Queen of Science," which mathematics is often considered to be. In this research, the book [162] can serve as a notable reference. It analyses, among others, two case studies described in [99]. An early version of this paper was read at the New York Academy of Sciences on 27 September 1978, during which Dauben discussed two turning points in the history of mathematics:

1. The Pythagorean discovery of incommensurable numbers and
2. Georg Cantor's development of the transfinite set theory.

The analysis performed by Dauben indicated a possibility of a "bloodless" scientific revolution which, to a large extent, called for an extension of the ontology of the current theory by new entities and a consequent reinterpretation of the previous concepts in terms of this new ontology so that the useful and positive legacy of the previous theory is not wasted. However, with a new ontological basis, the way we view previous concepts, as well as the scope and soundness of the methods provided by the previous theory, will be altered. To better illustrate this important phenomenon, let us cite Dauben [99]:

> This transformation of the concept of number, however, entailed more than just extending the old concept of number by adding on the irrationals-the entire concept of number was inherently changed, transmuted as it were, from a world-view in which integers alone were numbers, to a view of number that was eventually related to the completeness of the entire system of real numbers. In much the same way, Georg Cantor's creation of transfinite numbers in the nineteenth century transformed mathematics by enlarging its domain from finite to infinite numbers. Above all, the conceptual step from transfinite sets to transfinite numbers

> represents a shift that was in many ways the same as the shift from irrational magnitudes to irrational numbers. From the concrete to the abstract, the transformation in both cases revolutionized mathematics.

The view on scientific revolutions as presented in [99], after about three decades, still appears intriguing and calls for a further in-depth analysis, particularly on how to explain the prospective changes in the ontological basis of modern mathematics. More recent works in the field seem to agree with the conclusions of the aforesaid article, published three decades ago. To better understand the mechanisms and premises of the Cantorian revolution in mathematics, the substantive arguments of both the proponents and opponents of founding modern mathematics on set theory are currently in focus. Quite an interesting view, which concisely summarized this exchange of arguments, was presented by Anne Newstead [94]. The author wrote that:

> Paradigm shifts need not be a matter of total replacement of one theory by another. Often what happens is a theory becomes subordinated to a more encompassing theory. Rather than throwing out a theory and its existing ontological commitments, the existing scientific theory is incorporated into a more encompassing theory. The result is an ontologically conservative, bloodless revolution. This model of change, well documented in theoretical physics, is applicable to mathematical knowledge as well. Dauben [1992] argues that Cantor's set theory is an example of a mathematical revolution. Cantorian set theory requires a commitment to the existence of actual infinities. There are (according to Cantorian set theory) actually infinite sets of various sizes, each of a certain transfinite cardinality. The infinite, once thought to be beyond the scope of mathematics, could be tamed.
>
> Cantorian set theory is revolutionary in many ways: philosophically and mathematically. It throws off the Aristotelian prohibition against actual infinities in mathematics.
>
> It provides a new way of conceiving of the structures of mathematics as sets. Set theory is powerful enough to embed hitherto existing mathematical theory within itself, thereby providing a foundation for mathematics. The Cartesian dream of a certain foundation for knowledge seems within reach, if only the axioms and premises of set theory could be known with certainty. Of course we know from subsequent history that the Cartesian dream was shattered. The assumptions of set theory particularly the assumption that the infinite sets are well-ordered have not met with universal agreement. Possibly, the best possible way of vindicating these axioms is post hoc: they produce powerful, fruitful mathematics and are consistent as far as we know, so they should be assumed to be true.
>
> Among some professional mathematicians, especially those of a constructivist bent, the view persists even today that Cantorian set theory is a kind of religion. The implication of this view of Cantorian mathematics is that its theorems must be taken on faith without argument. This tone for the reception of Cantor's work was set in the nineteenth century by the caustic remarks apocryphally attributed to Leopold Kronecker: I don't know what predominates in Cantor's theory—philosophy or theology, but I am sure that there is no mathematics there. The view was perpetuated by Poincaré's description of Cantorian set theory as a pathological disease. However, eventually Cantorian set theory went main-stream as far as mathematics is concerned, owing to the staunch advocacy of Hilbert and Russell as well as its axiomatization by Zermelo.

It is worth emphasizing that since the beginning of the 20th century, Cantor's ideas have been ultimately largely accepted. They were strongly supported by David Hilbert in particular. Hilbert, in [205], firmly states that:

> No one shall drive us out of the paradise which Cantor has created for us.

This statement was important to Hilbert as the Cantorian set theory was treated as an effective instrument during the implementation of the Hilbert's Program.[4] One of its central points was the formalization of the whole field of mathematics. It meant, according to Hilbert, that all mathematical statements should be written in a precise formal language, manipulated and verified according to the well-defined and widely accepted rules. One can confidently assess that the mathematics of the 20th century was dominated by the application of a formal deduction system in set theory (or the theory of higher order types based on logics, such as modal, temporal, fuzzy and intuitionistic logic) to nearly all problems in natural science.

What is more, there have been even attempts to employ the language and deductive apparatus of formalized logical calculus to resolve philosophical disputes. Probably, the most intriguing philosophical results of this kind were obtained by Kurt Gödel who, paradoxically, showed that **most of the goals of the Hilbert's program (understood in the most popular way) were impossible to achieve**.

This is a formal consequence of the well-known Godel's incompleteness theorems. On the other hand, K. Gödel surprised logicians with his ontological arguments on the existence of God [143]. In this book by Fitting, we can find a formal machinery of the higher-type modal logic—the intensional logic. The machinery has been applied to provide a semantics for natural languages, to model intensional notions and treat long-standing philosophical problems. What seems the most important is the fact that the book contains the full formalization of the Gödel's ontological argument.

It is easy to imagine that not everyone agreed with the statement that we all are in "the paradise which Cantor has created for us." For instance, Wittgenstein, in [540], replied to Hilbert:

> Imagine set theory's having been invented by a satirist as a kind of parody on mathematics. Later a reasonable meaning was seen in it and it was incorporated into mathematics. (For if one person can see it as a paradise of mathematicians, why should not another see it as a joke?).

A broader context of the debate on the Wittgenstein's view concerning the fundamentals of mathematics, in particular, can be found in Internet.[5]

Contemporary debates on the attempts to express the intuition of effective mathematical constructions, based on the ontology of set theory, are still valid and they touch upon numerous aspects. One of such aspects is the research on the role of the Axiom (AC) of Choice[6] and its modification. The research has become popular due to a theorem, proved by Stefan Banach and Alfred Tarski in 1924, known as the Banach-Tarski paradox[7] [528].

Intuitively, these results can be reduced to a paradoxically resounding statement: Given a solid ball in 3-dimensional space, there exists a decomposition of the ball into a finite number of non-overlapping pieces (i.e., disjoint subsets), which can

---

[4] http://en.wikipedia.org/wiki/Hilbert's_program.

[5] http://en.wikipedia.org/wiki/Remarks_on_the_Foundations_of_Mathematics.

[6] http://en.wikipedia.org/wiki/Axiom_of_choice.

[7] http://en.wikipedia.org/wiki/Banach-Tarski_paradox.

then be put back together in a different way to yield two copies of the original ball which have identical size as the original ball. There are some “constructive” versions of the Banach-Tarski paradox. For example,[8] one can find an animation of a “constructive” version, discovered by Jan Mycielski and Stan Wagon [528]. There are many paradoxes similar to the Banach-Tarski Paradox. For example, Tarski’s circle-squaring problem is the challenge, posed by Alfred Tarski in 1925, to take a disc in the plane, cut it into finitely many pieces and reassemble the pieces so as to get a square of equal area. This was proven to be possible by Mikóls Laczkovich in 1990; the decomposition makes a heavy use of the axiom AC and is therefore non-constructive. Laczkovich’s decomposition uses about 10^{50} different pieces.

These examples of paradoxes, analogical to the Banach-Tarski theorem, indicate that the existence of mathematical objects and proofs may not always have any meaningful use in practice. Moreover, an indiscriminate acceptance of theorems pertaining to the existence of certain mathematical constructions or objects by contemporary engineers may prove misleading in practice, as well as yield infeasible designs and cause an enormous waste of time.

This warning to engineers is equally valid when it comes to the designs that rely on a huge computational complexity. For example, the failure of the Fifth Generation Computer Systems project (FGCS)[9] is a direct consequence of the inability to comprehend the computational complexity of the Boolean logic as a knowledge representation and a problem-solving language for AI complex applications. These examples also show that when it comes to applied mathematics, models that are to be employed should be cautiously selected to fully reflect the domain knowledge. Of course, this ability is non-trivial, but nonetheless, it is a key to the successful implementation of any complex system project.

Moreover, once such a model is established, its validity ought to be constantly verified by means of tests and experiments through its entire lifespan. No model is prefect. To prove this stance, let us recall the famous quote by George Box [65] (p. 424):

> Essentially, all models are wrong, but some are useful!

Without any doubt, the contemporary engineers would be in a much better position if modern mathematics, during its development, had taken efforts to facilitate their work in constructing meaningful models for specific applications. Certainly, one possible course of action is the minimization of risk in the process of model construction, which is often counter-intuitive. Particularly, one must avoid building models that are based on highly non-constructive mathematical results, such as the Banach-Tarski paradox or the Tarski’s circle-squaring problem. What is the use of a mathematical proof which says that it is possible to split a gram of gold and then reassemble the bits to obtain a million tons of gold?

---

[8] http://demonstrations.wolfram.com/TheBanachTarskiParadox/.

[9] http://en.wikipedia.org/wiki/Fifth_generation_computer.

The need to avoid such dangerous mirages (which are logical consequences of contemporary mathematics) calls for a thorough explanation of the premises of such uncomfortable illusions to adept engineers. Hints on how to cope with similar temptations might be equally useful. Hence, the attempts to look for the ways to modify the permissible mathematical constructions are currently undertaken.

The main goal is to eliminate constructions which defy the intuition of engineers in an obvious manner. For instance, in the case of the Banach-Tarski paradox, sets unmeasurable in the Lebesgue sense seemed to be the culprit at first. However, it turned out that the main problem was with the AC[10] which allows for an unlimited leeway in constructing infinite sets out of single elements picked from a family of non-empty sets.

The AC was formulated in 1904 by Ernst Zermelo in order to formalize his proof of the well-ordering theorem. During the first hundred years since its invention, the axiom had had a tremendous influence on the development of mathematics all around the world. Schools of mathematics emerged as a result of the analysis of its various aspects, with prime example being the Polish Mathematical School of the interbellum period, the establishment of which was described in [352]. Another book, [150], referred to the problem of AC in each chapter, while presenting the main achievements of set theory in the 20th century. Methods investigating the independence of AC from other axioms of Zermelo–Fraenkel set theory (ZF), have, on one hand, handsomely influenced the research on the Boolean-valued models for formalized theories [38] and on the other hand, helped understand the extent to which mathematics would change if AC was to be removed [215]. It should be noted that there was quite a number of attempts to modify mathematics by the elimination AC.

Among the attempts based on a classical logic, a proposal by Jan Mycielski and Hugo Steinhaus appeared particularly intriguing. It concerned a construction of objects, using strategies originated from two-player games, and allowed for the construction of mathematical entities using infinite choices. For instance, we can develop mathematics where all subsets of real numbers are determined (Axiom of Real Determinacy). This axiom is a version of the Axiom of Determinancy (AD) and, as it turned out, it is inconsistent with AC. In mathematics containing AD, the Banach-Tarski paradox would be invalid.

The idea of constructing mathematical entities using two-player game strategies, where infinite numbers of moves are possible, are not new. The intuition of such constructions was investigated about a hundred years ago by Stanisław Mazur, Stafan Banach and Wacław Sierpiński[11] [501]. It should be noted that AD has many interesting mathematical properties. It particularly implies that all subsets of real numbers are Lebesgue-measurable, have the property of Baire and a perfect set property. The latter implies a weak form of the continuum hypothesis (namely, that every uncountable set of real numbers has the same cardinality as the full set of real numbers). AD also implies that there is no uncountable well-ordered sequence of real numbers.

[10] http://en.wikipedia.org/wiki/Axiom_of_choice.

[11] http://www.telgarsky.com/1987-RMJMTelgarsky-Topological-Games.pdf.

AD was not the only attempt to negate AC when it comes to the creation of "non-constructive" sets. However, the majority of other attempts, aimed at refuting AC, had been largely rejected by most mathematicians. It was, among other things, due to the fact that AC was deemed consistent with or equivalent to many important theorems that has been already widely used in mathematics. For instance, it is equivalent to the statement that every linear space has a basis [53]. It is hard to imagine a contemporary mathematician who would work on linear algebra where some linear space may not have its basis. The rejection of AC would immediately require answers to such additional questions as "which linear spaces may have their bases and how to establish them." In other words, the rejection of AC would call for a gigantic effort in reconstructing the fundamentals of mathematics, in particular the analysis with its results intuitively deemed "constructive." An interesting result in this direction was obtained by Banach and Mazur in 1936–1937 [331]. These proposals, on the other hand, were not the only ones in the field of redefining the notion of "constructiveness" on the basis of classical logic. Many mathematicians maintain that a good basis for the investigation on the constructiveness may be the mathematics expressed by the theory of types, which can be investigated by both classical logic and intuitionistic logic. A highly concise abstract, summarizing the formal and logical aspects of this 100-year-long debate on the role of AC in the fundamentals of mathematics, can be found in [330].

12.4 Causes of CSE Crisis: Some Other Common Causes

Complexity paradox (cf. Chap. 11) and the CSE crisis pose a great challenge for today's managers, engineers and theoreticians involved in modeling, analyzing, designing, building, testing, maintaining, developing and managing large complex systems. Understanding the mechanisms that effectively reduce the risk of a project failure is not an easy task. In order to avoid the limitations of the "symptomatic treatment" of these problems, the critical issue is to realize what causes them. Certainly, the discovery of all the mechanisms that lead to the complexity paradox and the CSE crisis is a highly non-trivial task. Most probably, we are still nowhere near a full comprehension of these causes. Nevertheless, there are some groups of causes that are obvious for any experienced practitioner. For illustrative purposes, let us mention the following ones:

1. **Risk management of large complex projects is itself a very complex project**—apparently, it may seem that the level of our knowledge on risk management systems is so high that in order to maximize the chances of success for a given project, it is sufficient to ensure the wise implementation of risk management system, adapted to the specificity of a given project.

This is certainly true from a theoretical point of view. However, in this way, even from a theoretical point of view, we simply transfer one of the challenging problems with the implementation of a project into another complex project, which also requires risk management at a meta-level. In this way, by induction, we obtain an extremely complex hierarchy, integrating many levels of complex projects. Therefore, this course of action does not look encouraging. On the other hand, this controversial approach is based on the assumption that the level of our risk management knowledge is high enough for its adaptation to the specificity of our basic project. Certainly, this could be true for many types of projects, but definitely not for all of them.

What is worse, increasing the size and complexity of systems that are being constructed also increases the need for more sophisticated risk management methods, which are necessarily based on sufficiently proven solutions. Unfortunately, this increases the risk of a project failure.

There is one aspect of risk management, consisting in the ongoing and on-time risk assessment, which is particularly difficult. Treating the problem metaphorically, it comes down to a proper understanding and application of the mechanisms by which we perceive changes in environments in which we are acting. This is where we enter a very dynamic field of perception theory. Each specialist in this area is aware of the need to find a proper approach to perception modeling.

According to the author of this study, a particularly valuable approach is presented in the book [366]. It is based on the idea that perceiving is a way of acting. In other words, when we take into consideration a society of agents acting under some risk, then the perception of such a society of agents is tantamount to the way of acting of such a society. In other words, this is an appropriate model of interactive computations, based on active risk management agents.

It is worth noting that on the basis of the enormous experience gained in risk management, the proposals of standards and recommended procedures were created. Certainly, a special attention should be paid to the risk management ISO standard 31000:2009 and a family of related standards, in particular the risk assessment techniques standard ISO/IEC 31010:2009 (dealing with risk assessment concepts, risk assessment process and the selection of risk assessment techniques) as well as the risk management vocabulary standard ISO Guide 73:2009. The creation of the ISO 31000 family of standards is a very important step towards minimizing the risk of failure of large projects. However, this is only an important step in the right direction, which does not solve all the problems on the way to its goal. On the other hand, as indicated by practice, unfortunately, it often happens that large complex projects are not implemented according to the basic recommendations of the ISO 31000 family standards.

2. **Fast and slow thinking for planning fallacy**—Daniel Kahneman in his book [263] summarizes the research carried out by him and his collaborators (especially Amos Tversky) related to such domains as: cognitive bias, prospect theory, judgment and decision-making processes, behavioral economics, hedonic psychology and others. He noticed that the traditional approach to risk management,

using pure logical analysis and calculation of the "utility function," is based on the false assumption that people are generally rational. He stresses out that "the ideal of logical consistency [...] is not achievable by our limited mind." [263], (p. 335). In prospect theory, we can find a clear explanation related to the experimental errors of the Daniel Bernoulli's traditional utility theory. What seems the most important is that the Daniel Bernoulli's traditional utility (and its developments, such as the von Neumann-Morgenstern's approach to utility theory[12]) make logical assumptions that do not reflect human actual choices as they do not take into account behavioral biases. Kahneman [263] describes two very important and different ways the brain forms thoughts (cf. also Greenspan considerations about financial crisis in Sect. 9.3):

- System 1: Fast, automatic, frequent, emotional, stereotypic, subconscious.
- System 2: Slow, effortful, analytic, infrequent, logical, calculating, conscious.

According to Kahnemen [263], the interactive realtionships between System 1, System 2, and the environment, forms our thoughts and decisions.

Typically, we do not have any chances to apply System 2 or we do not even want to use it. Moreover, we do have a tendency to overestimate benefits and underestimate costs, impelling people to take on risky projects, which is one of the mechanisms of the "planning fallacy." Such an attitude leads to the so-called "pervasive optimistic bias" which "may well be the most significant of the cognitive biases." This bias generates the "illusion of control," which means that we think we have a substantial control of our lives. Optimists are more psychologically resilient, have stronger immune systems, and, on average, live longer than their more reality-based opposites. Optimism protects from the so-called "loss aversion," which is our tendency to fear losses more than we value gains. Unfortunately, when we see that something is wrong, we start to search for arguments that could confirm our wishes ("illusion of control"). "When people believe a conclusion is true, they are also very likely to believe its allegedly supporting arguments, even when these arguments are unsound. The conclusion comes first and the arguments follow" ([263], p. 45).

The overconfidence tendency is explained by Kahneman using the notion of "What You See Is All There Is (WYSIATI)." The WYSIATI determines our decisions. Primarily, it deals with the so-called *Known Knowns* (i.e., the phenomena that have been already observed). It rarely considers the so-called *Known Unknowns*, the phenomena that it knows to be relevant, but it has no information about them. In this context, we do not typically allow the possibility of the so-called *Unknown Unknowns*, the unknown phenomena of unknown relevance.

One of the many important applications of the approach to creative thinking and decision-making mechanisms is the appropriate construction of teams for problem-solving and project management. On one hand, it is important that the participants of each team are independent in thought-forming and do not display similar paradigm biases. On the other hand, it is very important that every team

[12] https://en.wikipedia.org/wiki/Von_Neumann-Morgenstern_utility_theorem.

(and team hierarchies) has relevant procedures for conflict resolution as well as decision-making mechanisms.

3. **There is no global model of complex system that could be understandable by one man and usually, the aggregation of local models is a very complex and costly procedure.** In the case of very complex systems, they are generally interdisciplinary in nature and one should not expect that a single person will be competent enough in all do-mains pertaining to a given project.

   For example, one can see this especially in IT, where the identification of user requirements and their consequent analysis are very difficult processes. There is a world of difference between the perception of users and that of the engineers. Each group builds its own model of perception, perceiving the system from a very local perspective. Even worse, in practice, it often happens that these local models are contradictory and fast-changing. With the increasing complexity of the design, the integration of these local models and the common perception of a target system become very challenging and requiring more and more advanced conflict resolution strategies.

   Of course, with the increasing complexity of projects by a rapidly growing number of local specialties, more and more complex interrelationships are being produced. In a dynamic environment, very often, there is a need to make changes to the local models and to the interrelationships among them. The decision which changes to make and how to achieve them despite the increasing complexity of a project is in itself a complex project, that has very strong constraints in terms of available resources, time and objectives.

   Only those who have already managed large projects and surmounted the consequences of introducing changes to them, fully understand how dramatic this experience is. Of course, there is a rapidly growing theoretical knowledge on this issue, covering a number of disciplines (e.g., configuration management).[13]

   Analogously to the experience in risk management, when it comes to the mechanisms for integrating "local models" into one integrated system architecture based on experience (acquired especially over the last decades), many standards and practices were established. In this regard, the ISO/IEC/IEEE 42010:2011—Systems and software engineering—Architecture description is particularly noteworthy. The standard specifies some of the best practices for describing the architectures that maximize utility throughout the life cycle. The key point of this standard is the specification of a typical content of the architecture framework. Of course, we have to select an appropriate architecture framework that matches the specificity of a project. In Internet, one can find the Survey of Architecture Frameworks, created in the spirit of the ISO/IEC/IEEE 42010:2011 standard.[14]

[13]http://en.wikipedia.org/wiki/Configuration_management.

[14]http://www.iso-architecture.org/42010/afs/frameworks-table.html.

4. **Too much knowledge can keep us away from true wisdom.** It should be noted that knowledge describing a given system is rapidly growing, with the increasing complexity and size of the systems that are being built. This expansion of knowledge can be observed both in a situation where we are fortunate enough to introduce our system in a single, integrated model, understandable to all stakeholders, as well as in a situation when a system that has been decomposed into a number of local subsystems can be reintegrated as a whole system.
   One can observe here some analogy to the minimum description length principle [414, 415].
   Each practitioner is well aware of the fact that in general, it is easier to build a system whose model has a small number of attributes and relationships, than a system with hundreds of thousands of attributes and relationships that must be analyzed on-line as in this case, the data streams come from multiple sources. Having to process so much data, it is easy to get lost in what is and what is not important at a given moment of time.
   There is, therefore, a fundamental difficulty, commonly known as the IT scalability problem. The problem lies in the fact that we are able to build programs (i.e., data structures and algorithms) which behave fairly well with small input datasets, but which, unfortunately, are not quite suitable for very large data collections. In such situations, it is necessary to use other programs. Unfortunately, for a number of important problems, there is no efficient program that could solve the problems with large input datasets. In this case, it is necessary to modify certain requirements, e.g., by accepting approximate solutions. This could be illustrated by the feature selection problem in data mining.
5. **Language and culture of project communication**—Fred Brooks in his book [68] strongly emphasized the problem of a rapidly increasing complexity and difficulty of problems, aiming at ensuring a proper communication between different subgroups of a given project.
   From the experience of the authors of this book, it can be inferred that the problem of a rapidly increasing complexity of communication networks in large projects is a "trivial task" in comparison with the problems arising from the use of "private languages" by almost every participant in a given project and by each, important for this project, group. However, even this problem is "trifle" when comapres to the problem of the "cultural personal level of communication with the environment." The larger the group is, the greater the likelihood of major problems occurring in this area is.
   There are many motivational techniques, which may influence certain habits of personal culture, but people are not machines and it is not easy, in a short period of time, to change their cultural habits. Therefore, it is often easier to replace a significant part of project participants with new people, than try to overcome this deceivingly innocent problem in a different way. However, in general, such an exchange of project participants is not possible and the person in charge of a given project must deal with a significantly increased risk of project failure that

may follow from this. The issues of language and culture and their influence on in-project communication discussed here, which lead to the occurrence of the complexity paradox and the CSE crisis, are unfortunately much more complex.
In addition to the issues that have been already raised, it is worth mentioning the problem associated with the discovery of important concepts, that are relevant for CSE, namely a dynamic and gradual shift in the meaning of these concepts. Intuitively speaking, while defining the objectives of any project and related quality requirements, we have to use some concepts that may change their meaning over time (e.g., concept drift in data mining [193]).
The experience acquired during the implementation of many projects shows that frequently, at the beginning of a project, complex concepts used to define the objectives and requirements are known and understood by everyone. However, these concepts change their meaning as a result of changes observed in environments and a better perception of them. Moreover, such concepts are often vague,[15] which makes the situation even worse. There are several approaches to this problem, for example, by means of rough sets paradigms and fuzzy logic. Certainly, an important aspect of a dynamic semantic issue is the understanding of perception. Of particular importance here is—previously mentioned—the approach to perception as a way of acting, outlined in the book [366].

6. **Sin of Pride**. The more practical experience and theoretical knowledge on the CSE crisis and complexity paradox we have, the braver we become. Consequently, we have to face more and more complex issues. As a result—especially in unstable and unpredictable environments—we are increasingly exposed to the risk of applying our knowledge and experience to situations to which this knowledge and experience are no longer adequate. In this way, step by step, there appear differences between the theoretical expectations that we have of a project and the reality. At the beginning, these gaps appear to be very small and they are often invisible and/or neglected by project participants. However, unfortunately, the previously microscopic gaps appearing in many places are often becoming larger and larger. Once the “critical mass” (which depends on the nature of a project and techniques of its implementation) is reached, they often trigger a “chain reaction,” leading to the failure of a project.
7. **Others**. In addition to the aforesaid examples of mechanisms that may cause the CSE crisis and the complexity paradox, there are many other very important mechanisms. Of particular interest seem to be the mechanisms involving laws concerning the complex phenomena, which are analogous to the well-known laws of physics and information theory, e.g., the third thermodynamics law about the entropy increase in closed systems, the Heisenberg uncertainty principle which says that the product of the position and the momentum of a project is higher than some constant, Newton’s principles, relativistic time principles, etc.

[15] http://en.wikipedia.org/wiki/Vagueness.

The above examples of mechanisms that contribute to the CSE crisis and the complexity paradox show how complex these problems are. They suggest that we should not expect simple solutions. Moreover, there is no certainty that we fully understand all important aspects of the issues that has been discussed here. One of the tasks which we have set for ourselves in this book is to propose a formal approach to a better understanding of the CSE crisis and the complexity paradox.

Chapter 13
Fundamental Precause of CSE Crisis (FP3C)

In theory, theory and practice are the same. In practice, they are not.

Albert Einstein (http://www.azquotes.com/quote/361676)

13.1 FP3C: Description

It should be very strongly emphasized that the example of the problem in understanding the requirements (cf. Sect. 12.1) is only one of many reasons that contribute to the CSE Crisis. More causes of CSE Crisis are discussed in Chap. 12 and in the Part IV of the book. For virtually all these causes, justifies FP3C (the general description of FP3C one may find in Sect. 1.2.2).

These systems, local models, and their aggregations should allow for the dynamics of change, determined by both their mutual interactions and interactions with their dynamically changing environment. Both local models, as well as their methods of aggregation, must be discovered on an ongoing basis (induced, constructed, adapted and appropriately archived), using dynamically changing, partially available, large data sets and diversified, specialized domain knowledge. Once constructed, they should be systematically analyzed, evaluated and adjusted to the changing needs of a project. To this end, we should also possess specific adaptive methods for reasoning about behavior of local models and their aggregations, which should allow, among others, for the adaptation of models (simple or aggregated) so as to achieve the intended objectives.

The lack of such methods is one of the major precause of the CSE Crisis. In this book, we offer an outline of such methods, based on the so-called adaptive judgment, applied to complex GrC.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_13

To paraphrase the title of Yourdon's book, "Death March" [545], one could say that for a project team, it is not enough to know the destination (a metaphor for the final product requirements and the results of a project). The keys to success are also:

1. Acknowledging the fact that before a project team marches in an unknown environment, we cannot foresee a global model, illustrating the actual path. Intuitively, when we choose to travel by car, it is not possible (and given the unpredictability on the road—it is not needed) to presume too much about the exact route of the car. We can only:
   a. imagine a very preliminary rough approximations of travel routes,
   b. on the basis of these approximations, initiate reconnaissance actions to precisiate and coordinate our perceptions,
   c. deploy adaptive organizational structures and related behaviors in such a way that the route of the march is identified on the basis of a cost-benefit analysis, including a risk analysis, instead of the random and often emotional behaviors of sub-teams.
2. Adapting the decomposition of currently important local tasks among a design team working on adaptive components and integrating their results. This decomposition should also include mechanisms for adaptive aggregation of components in the network/hierarchical structures.
3. Adapting the quality of communication, involving a systematic development and acquisition of required skills and knowledge, along with the assignment of tasks, adequate skill resources and the scope of autonomy and mechanisms to account for and introduce changes.
4. Adaptive implementation of effective techniques (if possible, independent from the team) for quality control, rapid identification of local errors and failures across all teams, as well as a systematic improvement of the learning process during a march, especially in terms of such aspects as:
   a. sufficiently accurate precisiation of meaning of specifications to which we are heading, tailored to the specificity of a march,
   b. important features of the environment in which we locally move and the environment in which we are planning to move,
   c. survival techniques and "travelling" in local environments to determine the current perception of the final destination,
   d. techniques for reasoning about next segments of a march, based on the collected data and generated knowledge,
5. Analyzing and discovering necessary changes, especially those regarding the reorganization of the local teams and their aggregation in networks/hierarchical structures.
6. Improving march adaptation techniques as well as techniques for identifying and evaluating important features of the environment associated with the march.

13.2 FP3C: Complexity of the Problem

The issue of complexity has been present in both theoretical and practical considerations on CSE since the emergence of the field. In the Sect. 12.2, we have used a quotation by Brooks [68] to emphasize and illustrate the problem. We have modified it to encompass any CSE project—instead of **"software entities,"** we have used **"aggregations of changing local models, interacting in an open environment."** In this way, we have arrived at the following paraphrase of the original statement by Brooks:

> Complexity. **"Aggregations of changing local models interacting in an open environment" are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the "local model level"). If they are, we make the two similar parts into one "local model level." In this respect "complex adaptive interactive systems" differ profoundly from computers, buildings, or automobiles, where repeated elements abound.**

The paraphrase presented above is relatively general. To specify it, let us recall that the FP3C is expressed by means of highly complex and vague concepts. For the practitioners seasoned in the implementation of complex projects, these concepts are related with a significant body of relevant experience and therefore, they understand those concepts better than theorists.

Every experienced practitioner has worked out his own techniques to minimize the risks arising from the FP3C. Junior adepts at CSE may be interested in understanding general mechanisms that cause difficulties in dealing with the precause of the CSE Crisis. Understanding these mechanisms should serve as a guideline for the development of effective tools for dealing with the CSE Crisis. Therefore, it is very important to identify these mechanisms.

Naturally, such an identification is subjective. When it comes to the conclusions presented in this book, they were drawn from the analysis of the commonly known literature and its confrontation with the author's personal and practical experience. On this basis, the author suggests paying attention to the difficulty of understanding and modeling the following mechanisms, which constitute the essential components of the FP3C:

1. Granulation of complex problem specifications, including the techniques for complex problem interpretation, specifications of plans for investigating potential solutions to complex problems, execution of these plans, verification of solutions, and representation of knowledge, both from external and internal sources.
2. Distribution of processes for data, information, knowledge, and other materials, related to the implementation of a project.
3. Interactions and adaptation strategies for problem solving through a network of cooperating agents of hierarchical structures.
4. Adaptive reasoning about IGrC, the induction of local models, aggregation of these models and aggregation of aggregations.

Herein, we shall briefly explain the key aspects of these difficulties.

13.2.1 FP3C: Problem of Granulation

Intuitively, the term **"granulation" refers to exploring, constructing, processing and communicating selected concepts (granules)** so as to effectively to deal with the problem-solving. It refers to the old principle of "divide and conquer." According to Zadeh [549] information granulation plays a key role in implementation of the strategy of divide-and-conquer in human problem-solving. This technique has long been known under different names. It effectively supports solving many problems in all areas of human activity, particularly CSE. The main difficulty associated with granulation lies in the ability to understand the mechanisms used for the discovery of new, useful concepts (granules) and their effective processing (e.g., aggregation, decomposition, scaling as required by application). These concepts should exhibit optimal efficiency in solving important problems and facilitating the communication between agents. In order to better illustrate the extremely large degree of difficulty, let us recall two examples from the history of granulation, aimed at supporting problem-solving in the following areas:

i. Storage, processing and communication of information. Major milestones include the discovery of such concepts as: language, speech, writing, paper, printing, computers and computer networks.
ii. The possibility to use various forms of energy. In this regard, the milestones are designated by discoveries of such concepts as: fire, fireplace, the relationship between mass and energy and technologies for energy storage and delivery.

These two examples show the importance of the techniques for granulation and granulation refinement. They also indicate that granulation refinement is, in general, a never-ending process.

While using concepts that are familiar to them, people think about the world in a language that comprises of these concepts. Using the very same language, they plan and implement their activities. Hence, they are restricted by a network of concepts that they have at their disposal. This observation also applies to CSE projects which could not have been implemented if it had not been for the human ability to create concept granulations with regard to such activities as specification of complex problems, including techniques for their interpretation, specification of task plans, verification of solutions and representation of knowledge from external sources.

In order to better explain the difficulty in dealing with granulation, let us once again take an example from the field of software engineering. In software engineering, one of the most important techniques used to minimize the risk of the CSE Crisis involves the adaptation of software development paradigms to the specificity of a project.

Over the past decades, there emerged a number of programming paradigms, such as: procedural, functional, structural, modular, object-oriented, agent-based, etc. Any amendments to these paradigms usually involve changes in the scope and form of representing granulation techniques for analysis, design, testing, deployment, and software development. Of course, each practitioner has his own preferences for paradigms (including project-specific granulation techniques). The author of this book is especially fond of the technique for "managing by small patterns the pieces of software," referred to in the book [152], which is quite an effective tool to reuse "granules" produced in other projects. This technique can be generalized to any complex project (including CSE) as a technique of "managing by small patterns the pieces of design solutions" or, even more generally, a technique of "managing by small patterns the granules/design thoughts"

The concept of granulation and its relationship with language is one of the central issues of this book. The key results regarding the concept of granulation can be found in Chaps. 31 and 32. It is also important that in the book, "agents' thoughts" are modeled in the form of the so-called c-granules (as described in Chap. 29), that are processed by the so-called inter-active granular computing (as described in Chap. 31). Consequently, general c-granules can represent "small patterns of software pieces," "local models for project completion" or collections of concepts used in a given language.

13.2.2 FP3C: Problem of Distributed Computing

Since the beginning of a century, the search for the better models of distributed computing has been closely associated with the application of computer science. The classic von Neumann architecture was introduced in the mid-forties of the last century, along with the Ulam-Neumann cellular automata computing model. In the late 70 s and 80 s of the last century, the idea to implement AI on millions (and later billions) microprocessors was formulated. It was inspired by the observation of biological phenomena. Namely, if the cerebral cortex of animals, such as a mouse, contains approximately 4 million neurons, computers with a giant number of microprocessors should be able to simulate the behavior of a mouse. It was to be a first step towards simulating a network of about 19–23 billion neurons (in the case of the human cerebral cortex).

Out of these considerations, computer projects with a very large number of processors were conducted. A good example is the Connection Machines project [206, 494]. This was a series of supercomputers manufactured by Thinking Machines Corporation in the 80s and 90s of the 20th century. Their architecture—as opposed to the classical von Neumann architecture—assumed massive data processing across multiple processors. The creation of this series of supercomputers gave hopes for the development of AI technologies. A special version of the LISP programming language (dominant at that time in the United States in the field of AI) was adapted to contemporary models for parallel processing implemented on Connection Machines. The computers achieved only limited marketing success. For example, in 1993, a version of the CM-5 FROSTBURG became the world's fastest supercomputer, reaching a computing power of 65 GFLOPS. However, the project as a whole did not meet the expectations concerning the progress in the development of AI technology and the company, Thinking Machines Corporation, went bankrupt in 1994. The Connection

Machines project was a very costly investment, aiming at unleashing a giant potential in distributed processing. The project has shown how difficult this can be.

Modern programming languages perform fairly well on a single-core processor (or a small number of CPUs). Computer programs can also be reproduced on many millions of CPUs. The problem, however, is that you are still only duplicating a small number of computing models. They do not really address the problem with the organization of concurrent and autonomous information processing by a great number of adaptive agents, engaged in the process of information exchange.

Therefore, for the time being, we do not have effective tools to model and simulate computing, carried out by the autonomous societies of agents. In recent decades, we have even regressed, namely, we have discovered how complex and "intelligent" the societies of various organisms, which were never suspected of possessing any intelligence, are. The bacteria that are able to communicate and cooperate with each other, in order to optimize their survival in the environment [41], is one such example. It is believed that a better understanding of the mechanisms of distributed computing and a better communication within a colony of cells can help find effective techniques for the treatment of diseases, such as cancer, or infections by bacteria resistant to antibiotics. The observation of these mechanisms also contributes to broadening our perspective on CSE project engineering.

When implementing CSE projects, we are primarily dealing with multiple human teams that carry out different types of computing, based on the habits and skills of their members. Often, in the case of large CSE projects, there are thousands of participants operating in different cultural environments. From the point of view of the management of such a project, it can very conveniently be regarded as a complex model of distributed computing to reason about and to steer.

The IGrC models, proposed in this book (especially in Chap. 32), are geared towards a distributed computing model, based on the interactions between autonomous agents and the environment.

13.2.3 FP3C: Problem of Interactions and Adaptive Strategies

Let us imagine that we have a societies of agents, whose job is to find a path to the metaphorical "food." Each agent may perceive the path by the metaphorical "intensity of smell," gave off either by the food or by other agents. Agents may, but do not have to, communicate with each other about their actual positions. In general, we can illustrate this type of a problem in such a way that the metaphorical "intensity of smell," gave off by the food, is represented in the form of a map, resembling a geographical map with mountains, valleys and submerged areas. Thus, we will use the commonly known graphic conventions as a language for the interpretation of our problem. For example, higher geographical spots may denote less intense smell of

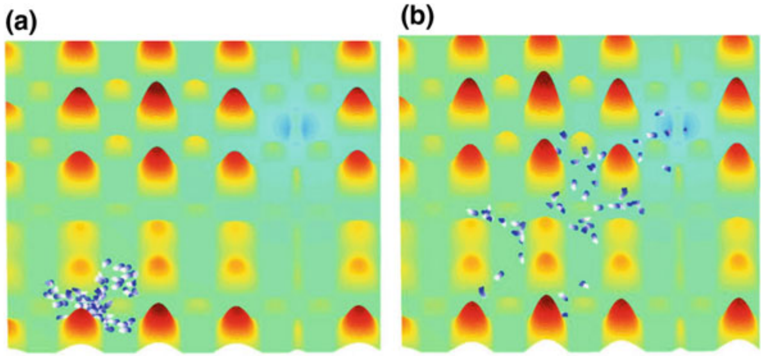


Fig. 13.1 Modeling the behavior of agents looking for metaphorical food [434]

the food. Most food is found where it is the deepest. For example, in Fig. 13.1, we have two frames, (a) and (b), depicting the search for food by the societies of agents.

In the frame (b), more agents are near the food, which is marked blue. In this typical problem, which illustrates the potential cooperation mechanisms between the agents, we can differentiate at least three types of agents on the basis of their behavior:

A. **Individualists**—individual agents that try to reach the food using their own ability to solve problems, maintaining a minimum level of cooperation with other agents.
B. **Agents engaged in interactive cooperation**—cooperating in accordance with the principles established for problem-solving. These rules may be refined by the conclusions drawn from solving the subsequent versions of the same problem. We may even assume that these principles were optimized in terms of exchanges performed by agents in interaction with each other and with the environment. This requires the selection of different parameters, such as a minimum/maximum distance between the agents, the scope and form of communication, etc. Intuitively, we can presume that each agent has established her/his own behavioral control vector for the entire problem-solving process.
C. **Agents engaged in adaptive interactive cooperation**—this is a variant of the interactive cooperation, in which principles of problem-solving can be adapted to the current constraints of problem-solving and circumstances present in the environment. Intuitively, we can presume that each agent has her/his own behavior control vector which may be changed during the process of problem-solving by adjusting its value to the circumstances of the current state of cooperation and the constraints of the environment.

The difference between interactive cooperation and adaptive interactive cooperation among the agents can be quite significant. For example, you can even assume that during the process of adaptation, the control parameters and/or functional structure of agents may change. Intuitively, the best type of cooperation should include adaptive

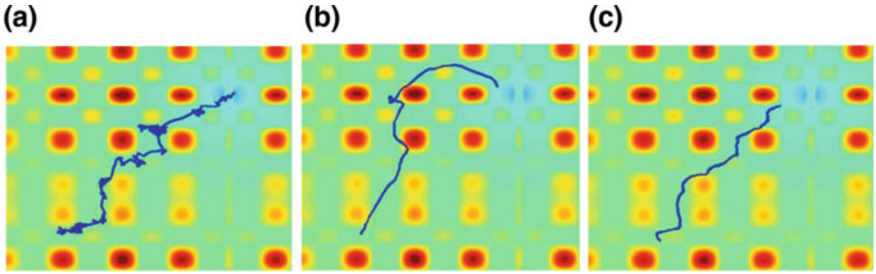
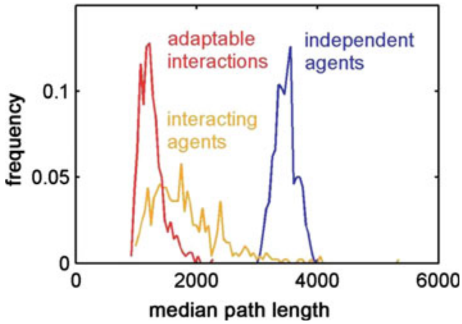


Fig. 13.2 Sample paths under three types of cooperation between agents [434]

Fig. 13.3 Comparison of the effects of three types of cooperation [434]



interaction. This is confirmed by the experience of the author who witnessed both the cooperation between people during the development of complex projects (POLTAX, Merix, Excavio, AlgoTradix) as well as the simulated behavior of agents, cooperating in solving combinatorial problems (AlgoTradix). It seems that these phenomena are well illustrated in [434], a work which focused on simulating a simplified behavioral model of bacteria societies. In Fig. 13.2, we can see the simulation of sample paths that various bacteria travel. The bacteria cooperate with other bacteria under one of the three types of cooperation: (a), (b) and (c).

In this figure, it is clear that the path traveled by the bacteria cooperating under the C mode has the smallest margin of error. A better comparison of the effects of these three types of cooperation is presented in Fig. 13.3.

This figure illustrates the effects of a series of experiments, carried out for each of the three types of cooperation between agents. The final result of each of these experiments can be represented by a random variable, denoting the median length of a path travelled to the food by the agents participating in the experiment. Figure 13.3 represents the probability density function resulting from the three random variables for the three types of cooperation among the agents: (a), (b) and (c). Therefore, on the *X* axis, we have the median length of a path for a given society of agents, while on the *Y* axis, we have the frequencies of median values.

It is worth emphasizing that the above experiment is also a highly simplified example of how to model calculations for adaptive problem-solving by a network of cooperating agents that constitute a hierarchical structure on their own. These structures may also be the objects of adaptation. For example, such a functionality can be added to the model presented above by taking into account genetic algorithms, which can support the evolutionary development of a computing model for a colony of bacteria.

The conclusions drawn from the experiments described in the cited paper demonstrate the superiority of adaptive interactive cooperation between agents. This simple experiment on distributed computing models, carried out using the societies of bacteria, conveys a profound message about the hidden links between adaptivity, hierarchical structures and interactions that support troubleshooting and problem-solving. Such relationships are also taken into account in the computing models proposed in this book. These models are based on the metaphorical equation, named the *Wisdom Equation* (cf. Eq. 8.1) To elaborate on the approach proposed in Chap. 29 of this book, we will explain how we understand the concept of "wisdom" in the context of the Wisdom Equation. At the core of our explanation is the presentation of a model for an interactive exploration, processing and representing the building blocks of "wisdom." We call these processes *Interactive Granulation*. In Chap. 29, considerations on Interactive Granulation and Wisdom Equation are presented in the following order:

1. A conceptual framework for Wisdom Equation and Interactive Granulation is described in Chap. 29,
2. A formalization of conventions for the application of Wisdom Equation to Interactive Granulation is described in Chap. 29,
3. A computational model framework for Wisdom Equation and Interactive Granulation is described in Chap. 29.

The metaphorical Wisdom Equation refers both to a single agent and/or the societies of agents attempting to act wisely in any possible field of application, that is X. For example, X can be a field of activity related to the implementation of a complex project. Taking X into account, the equation we be symbolically presented as follows:

$$\begin{aligned} &WISDOM\,[X] = \\ &\qquad INTERACTIONS[X] + \\ &\qquad ADAPTIVE\ JUDGMENT[X] + \\ &\qquad KNOWLEDGE[X]. \end{aligned} \tag{13.1}$$

Consequently, according to the equation presented above, the essence of an agent's wisdom (or the wisdom of a society of agents) in a given field of application marked as X is the ability of this agent (or the society of agents) to adaptively refine her/his processes of judgment (including analysis and evaluation) on the phenomena perceived by the agent in the surrounding world of physical interactions and in her/his memorized world of "impressions and mental perceptions." Therefore, if we treat the agent's world of mental impressions on par with her/his world of knowledge, which is a metaphorical simplification, then it can be said that the phenomena observed by the agent occur simultaneously in two different worlds:

- the world of physical interactions by interactive processes in which the foremost participant is the physical environment of an agent (or a society of agents),
- the world of the agent's mental perceptions of a given field of activity (X), contained in the mental world of an agent (or a society of agents), and of different media for the storage of knowledge.

13.2.4 FP3C: Problem of Adaptive Reasoning About Interactive Granular Computations, Induction of Local Models, Their Aggregations and Aggregation of Aggregations

The effectiveness and efficiency of IGrC depends significantly on the ability to predict the expected results. This statement is particularly relevant given the scale of negative consequences of the CSE Crisis. In this context, another highly complex issue is associated with both the understanding and the effective selection of problem-solving tools applicable in this field.

Regardless of whether we are talking about IGrC as carried out by the "societies of bacteria" that are looking for food or the "societies of engineers" responsible for planning and implementing CSE projects, when it comes to the success of such projects, the reasoning ability of those who actively participate in these interactions is crucial. They should also be able to adapt their reasoning to the past, current and possible future circumstances of this type of computing.

This type of reasoning can be implemented by sophisticated control systems, embedded at different levels in the organizational control hierarchy. Generally, these control systems, located both at the same and different levels of the control hierarchy, should maintain mutual interactions. For example, this could be the reasoning of a single participant, cooperating team, or a group of teams and, of course, it can also be the reasoning at the level of the entire society. What is more, each of these organizational levels of reasoning may include concepts of different degrees of complexity and abstraction. For example, the reasoning can refer to simple attributes, as well as to the relationships between these attributes, relationships between relationships, etc.

This type of ability to reason about the features of computing may allow one to construct and modify *basic components of interaction plan*, such as:

1. *Action rules* comprising of:

   i. *Concepts inherent to action rules*, in particular, the constraints on the process of interaction control.
   ii. *Pointers* to physical objects in the environment, the "action rules" of which trigger interactions.

2. *Contexts for using action rules and their hierarchy*. Contexts for using action rules are assigned to various *types of situations*. Each recognizable situation is assigned a different set of actions rules, along with their priorities. These different types of *situations* are characterized by the *dynamics of change in action scenes templates*. We assume that an agent which is able to perceive the environment of her/his activity can attribute to this environment a recognizable *action scene template*. In case such a template is not available, the agent can generate a new one. The dynamic paths of changes to this template, discovered, implied and projected, reflect the type of a situation in which the agent is involved. Generally—in order to predict the development of changes to the templates— an agent is capable of performing a reconfiguration and/or modification of templates in her/his memory, in accordance with the rules that it has previously learnt. It is worth emphasizing that the action scene templates may be defined using complex vague concepts. For example, this may include concepts related to the anticipated needs of an agent. What is more, often, these concepts are relative (in relation to other relative concepts). For example, when we have an interaction plan, the related action rules can be used in at least two contexts: external and internal.

   i. <u>Rules external</u> to a given interaction plan (external rules) rely on supporting the construction, selection, deployment, and supervision of its proper implementation, along with its modification, improvement, and completion.
   ii. <u>Rules internal</u> to a given action plan (internal rules) concern the implementation of a route according to the pre-specified plan.

3. *Control support concepts*. These are all concepts used for the sake of control support, including concepts employed in action rules and hierarchies of contexts. These concepts, among others, facilitate the construction of interaction plans and support an agent in controlling the failure and fulfilment of these plans. For a given interaction plan, concepts that drive *control over interaction* plan constitute an important group. These concepts feature such properties as:

   i. Conditions expected to be met by the agent's action scene upon a normal maintenance of the interaction plan control.
   ii. Conditions determining the failure of the interaction plan control.
   iii. Conditions to be preserved during the interaction plan control.
   iv. Conditions defining circumstances in which changes in the interaction plan control are needed.
   v. Conditions defining the necessity of introducing (adapting) recommended changes to the interaction plan control for particular circumstances.
   vi. Conditions indicating other recommended interaction plans in case of a failure in the control of an earlier interaction plan.
   vii. Conditions for stopping and/or interrupting an interaction plan.

The key components of the interaction plan control that have been described above occur at all local and organizational levels of the societies, executing interactive computing. Mutually dependent components of these plans can be limited to those that have been applied on one of the local layers of organizational control. A system of components specified in such a way is called a *local model of GrC control* or, in brief, a *local control model*.

Local control models are generally constructed on the basis of the interaction with other local control models and on the basis of the parallel, ongoing interaction with the environment. In a sense, using these interactions, these local control models can learn how to locally achieve their objectives and discover more and more subtle concepts that will contribute to an increasingly subtle expression and "understanding" of objectives, constraints and needs (to be met at a given organizational level of control).

Due to the interactions between different levels of control, we can expect that the entire society of local control models also learns. More vividly, we can say that for given conditions in the environment, a kind of the "crystallization" process occurs within local control models. As a result, a model of behavior for the entire society, adapted to the current conditions in the environment, emerges. When the conditions change, there may be a new suitable way of adapting the behavior of the entire society of local control models to the new situation.

At a rather high level of generality, a society of local control models can be regarded as an agent (for example, an agent understood as a design team or a colony of bacteria). In such a situation, we can select a "local control model" for the entire society of cooperating "local control models." Such a model is called a *global control model*.

It is also worth noting that a project-specific implementation plan, devised by a design team, can be considered a special case of a global control model. This is because such a plan is usually represented by a hierarchy of appropriate labelled, directed acyclic graphs. A label on each edge of the graph can be treated as an appropriate action rule in an interaction plan that controls the implementation of a project.

The reasoning leading to the construction, modification and development of local and global control models can be carried out in many ways. From a theoretical point of view, these can be divided into three groups: *bottom-up*, *top down* and *mixed*. In order to avoid misunderstandings, note that—in the case of non-trivial applications—in practice, there are only mixed ways of reasoning. Therefore, practitioners speaking of bottom-up (top-down) may actually mean the so-called mixed reasoning in which the bottom-up (top-down) reasoning merely dominates. To illustrate these concepts, consider the following examples:

1. Bottom-up In general, a colony of bacteria first generates local control models at the lowest organizational levels, along with the local sets of action rules. Then—on this basis—the local models are aggregated at the higher organizational levels (usually without the participation of the global control!) until a model, which we interpret as a global control model of the behavior displayed by the societies of bacteria, emerges. It is worth emphasizing that this is a special instance of an important class of natural computing models [420], which comprises of self-organizing computing models. Computing models leading to self-organization

[59, 110, 130, 136, 161, 184, 287, 311, 343, 365, 405, 420] are closely connected with the history of AI. In the dawn of the AI, such methods as cellular automata, neural networks and genetic algorithms were particularly popular. Currently, a significant portion of this type of computing models falls within the scope of the Natural Computing models.

2. Top-down With an overall vision of the aims and principles governing the completion of a CSE project—based on the experience and knowledge—we can decompose the project into smaller local tasks. Then, on the basis of the conclusions drawn from inductive reasoning conducted on similar local tasks, we construct a local computing model to support the process of solving local problems.
3. Mixed In practice, reasoning is carried out both bottom up and top down and it is aimed at constructing the sets of rules. Thus, we have interactions in both directions (on a back and forth basis). Therefore, in this case, we not only aggregate local models, but we also construct aggregations of these models for higher layers. In this way, we obtain the aggregation of aggregations, which often undergo decomposition and/or re-aggregation, etc. Depending on the nature of a CSE project, one particular direction can dominate.

The aforesaid mechanisms for the synchronization and aggregation of project-specific action rule sets may be illustrated using the adaptive resource management. In the case of a complex CSE project, we always have limited resources for its implementation. The members of a project team possess a limited level of competence and experience. Often, these resources are used in complex and previously unpredictable situations. Therefore, you need to take some precautions while making the decisions on assigning available resources, as well as related decisions on the decomposition and/or aggregation of tasks according to the, frequently obvious, substantive criteria that are a direct consequence of the do-main knowledge, necessary for a given project. These decisions should be further adapted to the constantly changing conditions during the implementation of a project.

In this book, we propose some techniques by which to control the process of reasoning, based on the concept of Adaptive Judgment. In this context, we propose that the adaptive reasoning about IGrC be grounded on basic techniques of the theory of management based on the SWOT analysis (S—Strengths, W—Weaknesses, O—Opportunities, T—Threats), cost benefit analysis (CBA) and Risk Assessment. A proposal of such an approach is presented in Chap. 33 of this book.

It is worth noting that the above-mentioned concepts on adaptive reasoning about IGrC, induction of local models, their aggregation and aggregation of aggregations, which involve such terms as action rules, interaction plans, hierarchy of contexts, control support concepts, etc., constitute specific examples of the basic concept of this book, which is the concept of a complex granule (c-granule). These relations with c-granules are more widely discussed in Chap. 33.

Chapter 14
TPGP: An Approach to Large Scale Context by Adaptive Selection of CSE Principles

14.1 TPGP: Y2K and Ultra-Large-Scale Systems (ULS-S) as Examples of Large Scale Contexts

At the end of the 20th century, the governments of many countries realized that there was a great need to be prepared for potential risks, resulting from the fact that a large part of the existing infrastructure used software which was unable to smoothly incorporate the change of dates between 1999 and 2000. This issue is known today as the *Year 2000 problem* (Y2K).[1]

For example, the author of this book was commissioned by the Deputy Prime Minister (Deputy PM) Leszek Balcerowicz to plan and coordinate the preparations to the Y2M issue in Poland, within the scope of the ministries that acted under the jurisdiction of the Deputy PM, including the Ministry of Finance. Luckily—in that case—a large part of the IT systems were prepared for the Y2K problem. At this point, we refer particularly to the POLTAX system, which was used to collect and transfer taxes into the national budget. Thus, the costs of preparations undertaken in Poland within this scope were insignificant.

However, when it comes to systems that were designed in the 1960s and 1970s, no one could expect at that time that they would still be operable at the turn of 1999 and 2000. Thus, in the second half of the 20th century, the most developed countries had to face significant costs related to counteracting the Y2K problem.

The accurate data concerning the resources devoted to Y2K remediation are unknown. There exist only some reports that provide certain estimates. For example, [347] presents the following information:

> In November of 1999, the U.S. Department of Commerce put the total cost of Y2K remediation at $100 billion.
>
> By 2006, the number had climbed. IDC published a report that year calculating that the preparation and New Year's Eve costs in the U.S. totaled $134 billion, with an additional

[1] http://en.wikipedia.org/wiki/Year_2000_problem.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_14

> $13 billion spent fixing minor problems in 2000 and 2001, according to analyst John Gantz. Worldwide, organizations were estimated to have spent $308 billion before the millennium on remediation efforts.

Considering the impact of inflation, the global costs related to Y2K were enormous.

A the same time, it should also be emphasized that Y2K had numerous global consequences that were proved positive. For example, one of those consequences was seen as a strong impulse for the IT sector. However, perhaps the most important positive effect of the Y2K crisis was that **the elites realized that the safety and functioning of the modern society depend to a significant degree on the ability to predict the *negative consequences of software operations***. These consequences are understood as the results of both software errors and unsatisfactory delineation of the requirements for software suits.

This is particularly related to a minimized risk of negative social consequences, resulting from software operations, which may lead to such significant problems as:

1. Breach of safety, (e.g., to a degree characterized by the consequences of the military network breach in the White House).[2]
2. Interruptions in the proper economic growth (e.g., to a degree characterized by the consequences of the erroneous computer assistance within the scope of mortgage risk assessment).[3]
3. Obstruction in the proper functioning of the financial markets, e.g., within the scope characterized by the results of the so called "Flash Crash".[4]

The three examples mentioned above show the distance we need to overcome in order to minimize the risk of negative consequences of complex software operations.

The Y2K problem resulted in the intensification of various initiatives, the aim of which was to *minimize the risk of negative consequences, related to the erroneous operation of software*. For example, as a result of this type of activities, "The Software 2015 Report" was published in the US back in 2005.[5]

This report strongly emphasizes the fact that software constitutes "critical infrastructure" within the "critical infrastructure." At that time, the 2015 Software Vision was stated as:

> Achieving the ability to routinely develop trustworthy software products and systems, while ensuring the continued competitiveness of the U.S. Software industry.

Within the context of the above-mentioned activities, the US Department of Defense ordered a project that was conducted by the Software Engineering Institute (SEI) at the Carnegie Mellon University (CMU)[6]: *Ultra-Large-Scale Systems*

---

[2] http://www.speartip.com/white-house-hacked-chinese-hackers-break-into-military-office-network-in-charge-of-the-presidents-nuclear-football/.

[3] http://en.wikipedia.org/wiki/Subprime_mortgage_crisis.

[4] http://en.wikipedia.org/wiki/2010_Flash_Crash.

[5] http://all-experts.com/assets/roadmaps/457_NSS2FinalReport04-29-05PDF.pdf.

[6] http://www.sei.cmu.edu/.

The Software Challenge of the Future [369]. According to the report which summed up the results of the aforesaid project [369], the motives that forced the researchers to launch the initiative were as follows:

> The U. S. Department of Defense (DoD) has a goal of information dominance-to achieve and exploit superior collection, fusion, analysis, and use of information to meet mission objectives. This goal depends on increasingly complex systems characterized by thousands of platforms, sensors, decision nodes, weapons, and warfighters connected through heterogeneous wired and wireless networks. These systems will push far beyond the size of today's systems and systems of systems by every measure: number of lines of code; number of people employing the system for different purposes; amount of data stored, accessed, manipulated, and refined; number of connections and interdependencies among software components; and number of hardware elements. *They will be ultra-large scale (ULS-S) systems.*

This report [369] includes a statement according to which, one of the biggest challenges formed by the ordering party (DoD) and faced by the SEI CMU team, was as follows:

> Given the issues with today's software engineering, how can we build the systems of the future that are likely to have billions of lines of code?

Seven years later, in 2013, Linda Northrop (Chief Scientist Software Solutions Division, SEI CMU), the leader of the ULS-S project, presented a paper, entitled "Does Scale Really Matter?: Ultra-Large-Scale Systems Seven Years after the Study".[7] One of the most important conclusions from her presentation is included in the first motto of this book.

As a result, we are not surprised that "A Five-Year Technical Strategic Plan for the SEI",[8] published in w 2015 by Kevin Fall (Chief Technology Officer SEI) is quite modest when confronted with the challenges referred to in the above-mentioned reports, which were created during the first decade of the 21st century ("The Software 2015 Report" and "Ultra-Large-Scale Systems The Software Challenge of the Future"). The following fragment illustrates the strategic plan of SEI for 2015 quite well:

> Software behaves differently than "physics-based" systems, such as engines, airframes, and ship hulls. Understanding its complexity and risks is hard, especially for large-scale systems-of-systems composed of many components of differing origins and pedigree. Our work in this area therefore focuses on enabling the government to obtain software-based "capabilities with confidence." Confidence is multi-faceted, encompassing cost and schedule, functionality, security, monitorability and other desirable properties including the -ilities (i.e., non-functional architectural features such as extensibility, flexibility, availability, and efficiency). Confidence also encompasses the level of assurance that individuals with conventional levels of education and training are able to effectively and safely operate software-reliant systems.

The above-listed phenomena, **related to the difficulties in understanding and handling the ULS-S complexity, particularly within the scope of understanding the reasons and causes of the "quickly growing gap between our research and**

[7] http://2013.icse-conferences.org/documents/publicity/ICSE_Keynote-Northrop.pdf.

[8] http://blog.sei.cmu.edu/post.cfm/sei-five-year-technical-plan-047.

the reality," constitute very important components of the motivation behind the research discussed in the present work. The main aim of the research was to determine the causes of TPGP and develop interactive computational models to ensure a better understanding of the TPGP nature (especially in the context of ULS-S. Some of the aspects related to the ULS-S problem will be covered in Sect. 6.5 and Part VIII.

14.2 TPGP: Links to Risk Management, Software Engineering, and Artificial Intelligence

The TPGP problem affects virtually all contemporary CSE projects that are carried out in dynamically changing environments, with incomplete information available. For example, in the field of software engineering, the TPGP problem appears under the name of a *software crisis* [68, 545]. The implementation of such complex projects often has serious repercussions. They lead to a significant limitation of the project's aims, extend the time allocated for the project's implementation, and increase its costs. Unfortunately, increased cost may sometimes lead to an early termination of a project and a significant waste of resources. Similar problems occur in all other fields of CSE. They are especially visible in the case of complex R&D projects and the creation of new technologies. On a global scale, TPGP causes difficulty to quantify, yet regularly occurring giant losses. The TPGP issue has been discussed in numerous publications[9] [69, 368, 545], based on the observation of problems that occur in numerous areas of CSE application. Still, there are no satisfactory answers to many basic questions.

The extent to which it is difficult to make a significant breakthrough in understanding and dealing with TPGP (in particular, with regard to complex software development projects) is fairly evident if one looks at a relatively slow progress in this field. This progress can be measured by the ratio of projects ending either with a disaster or costly problems to all projects completed. Perhaps, this stems from the fact that many contemporary diagnoses of the reasons behind TPGP as well as proposals of practical techniques for dealing with this problem are, to a large extent, based on problems and techniques of operation, which were developed several decades ago. Of course, it is evident that there has been a progress in this area over the last decades. There appeared many monographs, reports by consulting firms, and scientific papers which attempt to diagnose the reasons behind TPGP and propose solutions to the problem.

For example, among these activities there have been a number of new programming paradigms in software engineering. As a consequence, a progress was achieved in relation to numerous such activities. This progress was regarded by practitioners merely as the extensions and updates of the turn of the 70–80s. Therefore, these practitioners raised concern as to whether such a progress constituted a breakthrough or

[9] https://en.wikipedia.org/wiki/Software_crisis.

not. For example, about two decades ago, there appeared a useful book for practitioners of software engineering which summarized techniques applied at that time [152] to handle small pieces of software in such a way as to optimally manage their re-use in processes devoted to the construction of complex software systems.

One of the first deep analyses that investigated the causes of the software crisis can be found in the book from 1975 [68], written by Frederic Brooks who received the Turing Award in 1999 for breakthroughs in computer architecture, operating systems, and software engineering. It is worth recalling that similar aspects were highlighted by Brooks [68] and by other eminent scientists, the Turing Award winners, who were involved in the field of software engineering in the 1970s and 1980s, e.g., Edsger Dijkstra (Turing Award in 1972) and Niklaus Wirth (Turing Award in 1984). Twenty years after the publication of [152], the Python language started to gain popularity. Automatically, similar books, discussing these techniques as used in the Python environment [486], started to appear. These "new books," however, were not considered by practitioners as a breakthrough when it comes to understanding the causes of TPGP and/or developing techniques for dealing with this problem.

Returning to the somewhat pessimistic predictions expressed by Brooks in his work from 1986 [69], namely that there are limited possibilities to alleviate the problems related to the software crisis, it is worth emphasizing that eleven years later, i.e., in 1997, the first edition of—still popular today—Edward Yourdon's guide was published, which bore a telling title: "Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects" [545]. In this book, one may find the following criteria project failures:

> A death march project is one for which an unbiased, objective risk assessment (which includes an assessment of technical risks, personal risks, legal risks, political risks, etc.) determines that the likelihood of failure is > 50 *percent*.

Books by Brooks [68] and Yourdon [545] are still very valuable sources of knowledge for the beginners in managing complex projects. Those more experienced ones, however, often quote the following statement, attributed to Yogi Berra (as well as to Jan L. A. van de Snepscheut and Chuck Reid)[10]:

> In theory there is no difference between theory and practice. In practice there is.

As the complexity of projects grows, so does their cost of failure. Therefore, there is a growing interest in finding any possibly effective actions, that could help to reduce the gap between the practice and theory of CSE. An increasing number of works that are being published and more and more complex theories are being developed, which makes it difficult to form an opinion on the practical value of many new approaches. Certainly, a primary criterion is practical experience. It indicates that today, we deal with risk management in a wrong way. Consequently, in practice, this means that it is still impossible to have an "unbiased, objective risk assessment" of complex projects, as postulated by Yourdon [545]. For example, in 2008, Alan

[10] www.barrypopik.com/index.php/new_york_city/entry/in_theory_there_is_no_difference_between_theory_and_practice_but_in_practic.

Greenspan, who served as the head of the Fed Reserve for 20 years, stated before the US Senate Committee that[11]:

> modern risk-management paradigm held sway for decades, [...] The whole intellectual edifice, HOWEVER, collapsed in the summer of last year.

In other words, with the experience of many global financial crises, we have realized that the concept of risk is too complex to be measured by sets of attributes, known since the beginning of the 21st century.

To better understand the unsatisfactory performance of modern tools for supporting risk management, let us imagine that we are to evaluate the risk of a project devoted to a legal, organizational, and technical modernization of the Ukrainian tax system in 2015, from the point of view of its integration with the EU standards. Someone may argue that this is a rather unique situation. In practice, however, under a seemingly calm surface, strong forces and tensions are often hidden, which can unexpectedly and rapidly bring about enormous changes. For example, this happened in Poland during the transformation of the country's tax system at the turn of 1989/1990. Luckily, there was no civil war. However, although the situation seemed stable, there were numerous unexpected scenarios, also pessimistic ones, waiting to unfold. Despite this large uncertainty margin, at that time, the planners of the project for a legal, technical, and organizational modernization of the Polish tax system had to manage the risk of its possible failure as well. Today, only few people understand the scale and complexity of the POLTAX system development, the core of which continues to support the operation of the contemporary Polish tax system, including its risk management, planning, design, construction, and implementation (especially at the beginning of the 1990s). These examples show that under certain circumstances, it is impossible to perform an "unbiased, objective" risk assessment, which is recommended by Yourdon [545] before starting any project. On the other hand, sometimes, the circumstances in the market or a given political situation require the acceptance of the pre-identified risk when starting a certain project and an effective management of this risk only after the project has been launched.

The above example shows how important it is for the participants of a CSE project to rapidly learn complex and changing concepts, which constitute the basis for information processing and decision-making processes related to, e.g., risk or user's requirements. These decisions form the basis for a wise interaction among human beings and societies of artificial agents in a partially known environment.

Let us illustrate the degree of difficulty of MACAM and managing of PC-TPGP. To do it, let us imagine an idealized CSE project where we achieved sufficient level of understanding of information processing mechanisms so that it helps systems achieve goals (specified by agent's needs) through learning the optimal principles of wise cooperation among human beings and the societies of artificial agents. According to this assumption, we have tools which enable us to identify and learn, in an adaptive and quick manner, the appropriate approximations of important, complex,

[11] http://economix.blogs.nytimes.com/2008/10/23/greenspans-lament/?_php=true\&_type=blogs\&_r=0.

and evolving concepts, (e.g., concept of risk), providing, at the same time, an effective support for the completion of a project and ensuring a wise cooperation among agents (human beings and artificial agents) in a changing environment.

Therefore, for such an idealized case, project's participants are able to decide whether a given project meets the Yourdon's criteria [545], i.e., whether it is eligible for a "death march project." To this end, the participants should first, quickly learn the concepts expressed in the criterion: "for which an unbiased, objective risk assessment (which includes an assessment of technical, personal, legal, and political risks, etc.) determines that the likelihood of failure is more than 50%" and, on this basis, determine whether a given project is feasible or not. If they decide that it is not, then they should provide reasons. Next, the participants can assess the opportunities and conditions for the success of a project. If it is necessary to launch a project under high risk conditions, then—in this ideal situation—the participants have a great chance to identify those concepts, which constitute the mechanisms for maximizing the chance of success, as well as concepts that determine optimal moments to undertake appropriate actions aimed at minimizing the risk of a failure.

Note that in this context, the term "concept" is used to represent a phenomenon, a property, a relation, or a choice of an optimal—at least satisfactorily optimal—scenario for the next action. On the other hand, "to learn a concept" means that we are able to learn effective and sufficiently fast techniques, which can be used to decide whether a given concept is true (to a satisfactory degree) or not. Sometimes, instead of saying that "a concept is true" (to a satisfactory degree), we can also use another term, rooted in logic, that is that "a concept is satisfied" (to a satisfactory degree).

In practice, such an idealized complex project, in which participants would have perfectly mastered the ability to identify and adaptively learn appropriate approximations of important vague complex concepts, does not exist. Moreover, it is often impossible to learn any satisfactory approximation of even such concepts as the "requirements of the future users." This is because these requirements often change unpredictably over time and their description requires the use of vague complex concepts [74, 118, 268, 269, 387, 417, 443, 471, 542], which cannot be precisely defined, using a set of available attributes. Fortunately, in many situations, it is possible to learn the approximations of concepts and trends in their potential changes to satisfactorily meet the current and future needs of agents. In other words, in practice, it is generally necessary to operate on appropriate approximations of important concepts in a wise manner and learn the optimal approximation of trends in the changes of concepts that are important to agents.

14.3 TPGP: The Problem of Selection and Adaptation of CSE Principles

The principles discussed further in the book constitute a summary of the author's extensive experience in implementing and/or monitoring about 200 complex projects, which were carried out in different work cultures (including work culture characteristic of East-Central Europe, the USA, Japan, France, or Germany). **They should not be treated as the best and, let alone, the only principles under any circumstances**.

The analysis of CSE principles presented below (Sect. 14.3) is a sample representation of a fragment of knowledge, on the basis of which it is possible to select and/or discover adequate principles for implementing complex projects. Therefore, it does not present the full knowledge related to this subject in a comprehensive way.

On the other hand, the author would like to encourage everyone who is interested in a successful implementation of complex projects (especially projects implemented by the public administration and/or state-owned companies) to consider the possibility of increasing the financial efficiency of projects, both planned and implemented, by obeying a relevant family of principles and using adequate strategies to adjust those principles to the constantly changing project constraints.

It is very difficult to understand why **in numerous countries (including Poland), gigantic public funds are still being wasted on the implementation of complex projects** which are doomed to failure due to simple mistakes. It is even more surprising given the fact that every experienced CSE project manager knows exactly that **failing to obey appropriate principles, adjusted to the specificity and context of a given project, leads to a project failure**. In spite of this, it is still easy to identify numerous examples of situations in which massive public funds were wasted due to financing complex projects without obeying an appropriate family of CSE principles (adjusted to the specificity and context of a given project). At the same time, families of such principles can be relatively easily constructed by adjusting the simplest principle, which has been verified in practice (such as the ABC Book for the Implementation of Complex Projects—cf. Sect. 14.3.2.6), to a given project. It should be emphasized that the principle of the ABC Book refers to the well-known ancient principles of implementing complex construction projects.

The author suggests that in the future, legal regulations concerning the processes of planning, launching, implementing, and developing complex projects should be introduced (especially in countries such as Poland). The principles gathered in this book (e.g., BPDC principles and their specifications in the form of POLTAX implementation principles) may be treated as a material for a further discussion on this topic.

As a result of this process, legal regulations (like the metaphorically understood construction law) could be developed to force both the decision makers (especially in the public administration and/or state-owned companies), who launch complex projects, and the participants of these projects to obey an appropriate family of principles. The regulations should particularly enforce the compliance with the principle of Independent Verification and Validation (IV&V) (cf. Sect. 14.3.2.5), the results of which should be, among others, the preparation of a documentation for top managers of a given organization with information about the risks and problems related to the implementation of complex projects.

Taking the aforesaid motivations into consideration, the primary aim of this chapter is to sum up, in a concise manner, practical experiences of the author. They are presented in a form of universal and exemplary principles for the implementation of complex projects.

Moreover, CSE principles were presented here to **indicate the ontological complexity of TPGP problems (which also includes the complexity related to the dynamics of ontological changes)** and **illustrate the basic relationships between the implementation of complex projects and computations in IGrC**. These relationships constitute the motivation and basis for important research trends presented in this book. For example, in Sects. 1.4 and 14.4, we present selected R&D problems related to the selection and adaptation of CSE principles. Moreover, these relationships are elaborated on further in the book, particularly in Chaps. 21 and 31.

The principles mentioned in this chapter are mainly universal in nature. However, in the third part of the book, exemplary specifications of these principles, in the form of specific principles for particular domains and contexts of their applications, are presented in following sections: POLTAX—Sect. 16.3, AlgoTradix—Sect. 17.2, Merix—Sect. 18.2, Excavio—Sect. 19.2, and UAV—Sect. 20.2.

14.3.1 Approach to TPGP Based on a Skillful Selection, Obedience, and Adaptation of an Appropriate Family of Principles

The facts mentioned in Sects. 1.3 and 14.1 prove that the problem of TPGP (signalized in the Preface in point A and discussed in Part III i.e., Chaps. 9–15) has been known and intensively investigated for numerous decades.

The majority of practitioners still believe that **the progresses in solving the problem, which have been done so far, are insufficient. What is even more important, these progresses are disproportionately small in comparison to huge expenditures incurred as a result of initiatives that, as intended by their designers, should lead to a noticeable decrease in losses related to the negative consequences of TPGP**. Unfortunately, these losses, despite all initiatives that have been undertaken, still turn out to be extremely big.

It is therefore not surprising that project sponsors, managers, and/or engineers **draw, above all, on past experiences** (both theirs and their authorities') to increase the likehood of success. These experiences are often expressed in the form of *principles of engineering*, which are referred to, in short, as *principles* (cf. [100, 107, 164, 279, 530, 531, 545]). They particularly include the principles for the implementation of CSE projects, especially in terms of planning, designing, implementing, developing, and migrating CAS systems. To emphasize that a given principle is related to a CSE project, it is sometimes referred to as a *CSE principle*.

In the context of the book, **the intuitions concerning the principles described above indicate that they refer to the properties of a project and/or good practices in terms of planning, commissioning, stabilizing, implementing, adapting, developing, and completing a project**. These principles usually fall under one of the following groups:

1. *Specification of desired project's properties* (e.g., compliance with a family of standards, good practices, and/or required patterns of behaviours and/or patterns of solutions that can be adapted and/or re-used in a project).
2. *Rules of behavior*, including rules for the development of structures related to management, organization, competences, and knowledge, which consist of:

   a. *Specification of conditions for launching an action* (usually expressed by means of vague concepts) along with postulated actions (or plans) that are to be undertaken when the satisfiability degrees of these conditions meet adequate bounds (vague conditions).
   b. *Specifications of conditions for prohibiting an action* (generally expressed by means of complex vague concepts) along with prohibitions to perform certain actions (or plans) when the satisfiability degrees of these conditions meet adequate bounds (vague conditions).

3. *Recommended techniques of adaptive judgment* on significant aspects related to a project and its state of implementation. For example, by analyzing the streams of project's data, these techniques can generate arguments "for" and "against" the occurrence of some sort of a risk (or co-risk cf. Sect. 32.4.4) and/or the fulfillment of important properties. Next, the techniques that are being used may indicate other techniques for the aggregation of arguments "for" and "against" in order to determine the final judgment concerning the level of risk (or co-risk) that is being investigated and/or the satisfiability degree of a given property. **The techniques of adaptive judgment should also recommend "best possible" proposals of reasoning schemes and domains of knowledge that are useful in solving current problems related to the implementation of priority needs**.

Experienced managers and engineers, who deal with principles of software engineering [100, 545] and, more generally, with CSE principles [107, 530], engage in discussions about the criteria for evaluating the value of implementing and obeying these principles, taking into consideration the current properties of a project and its constraints.

Intuitively speaking, judgment and deciding what the consequences of obeying a given principle will be (positive, negative, or neutral) or whether it is actually possible to evaluate this, depends on perception of the current specificity and context of the project's implementation.

Taking into account the observations made in the previous paragraphs, it might be possible to classify the usefulness of obeying the aforesaid principles, depending on a given specificity and context of the project's implementation. In this way, it is possible to define various types of principles and conditions (determined by the project's specificity and context) for obeying them. From a practical point of view, the following two types of principles and conditions are particularly important:

PoS. *Principle of project survival*, referred to, in short, as the *survival principle*. These principles support a successful and efficient implementation of a project in its current context and specificity. It means that a skillful selection (i.e., one that takes into account the context and specificity of a project) of and compliance with survival principles when implementing a complex project gives good chances for both **reducing the risk of a project's failure and increasing the efficiency of its satisfactory implementation**.

PoF. *Principle of project failure*, referred to, in short, as the *failure principle*. Obeying these principles (in a given context and with a given specificity of a project) usually leads to a project's failure. This failure means fulfilling (at least) one of the following, potentially measurable conditions:

(i) a significant increase (e.g., by at least 50% in comparison to the initial assumptions) in costs and/or time needed for the implementation of a project,
(ii) a significant reduction (e.g., by at least 50% in comparison to the initial assumptions) in potentially achievable final results and/or their required quality,
(iii) a significant increase (e.g., by at least 50% in comparison to the initial assumptions) in the risk of having to terminate a given project during its implementation (e.g., due to the lack of necessary resources and/or motivation for its continuation).

The properties, conditions, and judgment techniques which are part of the survival principles constitute bounds that determine the area into which the states of currently implemented projects (or, more generally, currently implemented and/or forecasted computations) should fall.

Analogously, the failure principles specify the "area" where relevant action should be taken, to remove project from this area, as soon as possible.

Practitioners [545] very often use the phrase "survival principle" (or, respectively, "failure principle") to refer to a principle and circumstances under which a given principle has a nature of a survival principle and/or failure principle. Of course, they are aware of the fact that under certain circumstances a given principle will no longer be a survival principle (or, respectively, a failure principle). Therefore, it should be emphasized that the difficulty with the selection of principles for a specific project significantly depends on the quality of perception with regard to the dynamically changing specificity of a project and its context of implementation. Every experienced manager and/or engineer knows exactly how difficult a challenge it is to appropriately select principles for planning and implementing CSE projects and adjust them to the constantly changing project constraints (including the context and specificity of a project, which reflects the evolution of a project in time).

Let us notice that we have discussed *survival principles*, not *success principles*. It is so because numerous experienced managers, engineers, and scientists still recognize the validity of the previously cited thesis by Frederic Brooks [68, 69], despite the fact that it was formulated three decades ago. It says that in "the nearest decade" (which is the very next decade), no universal CSE principles, that will guarantee the success of complex projects, will be created.

Practically every project sponsor, manager, and/or engineer engaged in the implementation of a complex project claims that her/his aim is to implement

a given project with success. They are convinced that they skillfully select and obey (during the project's implementation) survival principles (PoS) and at the same time, avoid any actions which are considered improper (**e.g.**,**support failure principles (PoF)). Of course, if everyone truly obeyed appropriately selected family of survival principles, failures of complex projects would not be so frequent. Moreover, the negative consequences of TPGP would not be so great.**

It means that in practice, the aforesaid optimistic declarations made by sponsors, managers, and/or engineers are only a wishful thinking and do not reflect the actual state of affairs. This results from the fact that "survival principles" are wrongly selected and do not match the current specificity and context of a project or that appropriately selected families of principles are not obeyed.

Often, it leads to actions which should be otherwise avoided. For example, this may be caused by obeying a survival principle which was significant in a previous context, but not in the current context of a project. In this new context, obeying this principle may in fact mean supporting or following a failure principle.

Managers and/or engineers (even those experienced ones) **often cannot properly identify and discern between survival and failure principles**. The properties which distinguish survival principles from failure principles may significantly change over time. Once again, let us remind that whether a given principle is considered a survival principle and/or a failure principle significantly depends on the current specificity and context of a project.

Properties which define the criteria used to classify the aforesaid principles are often expressed by means of very complex vague concepts (e.g., concerning the risk cf. Sect. 9.3), which change their meanings over time. At the same time, approximations of concepts used by agents are usually continuously improved. These approximations may concern concepts which changed their meanings (without the knowledge of an agent). Consequently, quality and the usefulness of such approximations decreases. It should be also emphasized that often, managers do not know all conditions that determine the current classification criteria (with regard to survival and/or failure principles). Thus, these conditions are often unattainable for the agent's perception (e.g., because a given agent does not use an adequate ontology to express these classification criteria). **As a result, for both experienced and inexperienced project sponsors, managers, and/or engineers, it is extremely important to use mechanisms that help discover the best possible approximations of criteria for identifying, analyzing, classifing (** **i.e.**, **deciding whether a given principle is currently a PoS or PoF), evaluating, and adapting a family of actually obeyed (and not allegedly obeyed) principles**.

In his practice, the author noticed that the more experienced a sponsor, manager, and/or engineer is, the less sure she/he is whether the principles that are currently used are the principles of survival.

It may be observed that managers and/or engineers with little experience in CSE projects (with less than 10 years of practice) are often too self-confident when it comes to selecting survival principles that are to be used during the implementation of a project. Especially, they do not take into account subtle, but visible changes in the context of a project and its constraints.

However, managers and/or engineers who are experienced in CSE projects (with more than 20 years of practice) are very sensitive to any changes in the context and specificity of the project's implementation. Therefore, they consistently verify, validate, and evaluate which principles are obeyed and whether they are properly selected and adapted to the actual and current specificity and context of a project.

Summing up, we obtain the following conclusion:

With a limited access to information about the context and specificity of a given project, it is often impossible to objectively discern between survival and failure principles. Moreover, when identifying the type of a given principle, it may turn out that meanwhile, the context and specificity of a project changed and a principle which was a PoS became a PoF.

14.3.2 Examples of Universal CSE Principles

It should be emphasized that principles presented in this book may be used in other (than mentioned) work environments and cultures. They are great for cultures characteristic of the USA, UE as well as multi-cultural projects.

Every experienced manager—of CSE projects—has her/his own family of basic CSE principles. When formulating and improving these principles, she/he tries to make them as much universal as possible for the potential conditions of use. Certainly, an experienced manager does this with a great deal of caution as she/he **is aware of the fact that there are no two identical CSE projects**.

14.3.2.1 Principle of Anchoring a Project in Top Management

Based on the author's experiences, the *Principle of Anchoring a Project in Top Management* is, to a large extent, one of the most important critical susses factors for any strategic CSE project for any organization. Unfortunately, in general it is not obvious for public sector (especially in the countries of East and Central Europe). Roughly speaking the principle is **related to ensuring, by top management, a proper implementation of tasks** listed in Appendix A.

Small and easy to implement projects, which have no strategic significance for a given organization, do not require anchoring in top management. Therefore, the principle in question does not need to be obeyed in this case. However, the lack of anchoring (in a similar sense to the one presented above) in the case of complex CSE projects can be treated as a classical example of the most important failure principle (PoF, as defined in Sect. 14.3.1). Consequently, in theory (quality management standards, sets of recommended practices for the implementation of CSE projects), it is strongly recommended to anchor complex CSE projects in top management. Unfortunately, these theoretical recommendations are not always properly reflected in practice. The lack of understanding when it comes to the role of top managers is often linked with their relatively little responsibility (especially in the public sector) for losses resulting from project failures.

At the same time, it is worth emphasizing that anchoring a project in top management should not mean that from now on, top managers of a given organization will directly manage projects or that they will directly perform the aforesaid tasks related to "project anchoring." Their role is to ensure a proper implementation of tasks a1–a10 in Appendix A by establishing appropriate structures and launching appropriate processes in an organization.

From about four decades of the author's experiences and observations, it seems that the most important failure principle described above is extremely reliable. If a project (a CSE project) is not anchored in top management, it always ends in failure, as the principles says (by a failure, we mean a "project failure" as determined in PoF in Sect. 14.3.1).

An experienced manager can relatively quickly and easily recognize whether the top management of a given organization complies with the most important CSE projects' failure principles (with regard to a particular project managed by them). Undoubtedly, if there is a great risk that the principle will not be obeyed, a good manager should think whether she/he may continue her/his mission in such a way as to minimize the negative consequences of this risk. If not, the manager should perform a broadly understood cost/benefit analysis to check whether it is reasonable to continue the mission. The author experienced situations in which he asked for the discontinuation of the mission (e.g., [280]). However, there are also some circumstances, in which it is necessary to continue such a "mission impossible."

Based on the author's experiences, if a manager is faced with such a situation (that is, she/he decides to continue such a "mission impossible"), she/he should not directly inform top management about obeying the most important failure principle until she/he clarifies this for the following issues:

1. Manager may be mistaken and jump to conclusions. Moreover, she/he may not understand well enough the actual specificity and context of a project. In such a case, informing top management about using the most important PoS is unprofessional. First, all doubts and ambiguities should be resolved.
2. In general, top management needs a lot of time to understand that the most important failure principle is being implemented. Stating the fact by a project manager, who is not part of the top management of an organization responsible for a project, may be treated not as a suggestion to change the decision made by top managers, but as a request to dismiss the project manager.
3. Top management may understand and use the principle on purpose. For example, they may be motivated by a willingness to "show" that the previous top management acted to the detriment of an organization. Of course, in such a situation, a project manager has low chances of influencing the decision made by the top management.

In all case scenarios mentioned above, a project manager is in a very difficult situation. If still she/he assumes the responsibility for the success of a project, the author recommends to focus on discussing and analyzing, together with top management, issues related to the appropriate selection of survival principles. If talks with top management are especially difficult, the author recommends to start the conversation from explaining the aims and analyzing the scope of a family of principles presented further in the book, especially the *ABC Book Principle* (cf. Sect. 14.3.2.6) and/or its an extension *BPCD family of principles* (cf. Sect. 14.3.2.9).

The most difficult situation occurs when top management is not interested in the quality of projects implemented by an organization and top managers, empowered to make specific decisions, do not find time to deal with uneconomical waste (often considerable) of resources on project failures.

14.3.2.2 Principle of Continuous Identification, Judgment, and Adaptation of a Family of Currently Applied Principles

Earlier, we emphasized that each survival principle in specific contexts of the project's implementation and under specific circumstances may become a failure principle and vice versa—each failure principle may become a survival principle. Therefore, all principles must be continuously identified and interactively evaluated to check which type they represent with regard to the changing specificity and context of a project. If justified, in lieu of a family of previously obeyed principles, a new family will be adopted. At the same time, it is recommended to optimize a family of principles, paying particular attention to those criteria, which help achieve the best possible objectives of a project, while fulfilling its assumptions. Intuitively the principle can be treated as some kind of enforcement of "adaptive interactive cooperation" in the sense of bacterial cooperation as it is described in Sect. 13.2.3, especially see Fig. 13.2. An example of a family of universal criteria for optimizing a family of principles applied in a given project can be found in Appendix G.

14.3.2.3 Principle of Providing an Appropriate Role of the Human Factor in Management

The author noticed that in a not yet well developed work culture characteristic of numerous countries (especially those which undergo systemic transformations), project sponsors, managers, and/or engineers with little experience have a tendency to **underestimate the role of the human factor in management of CSE projects**. It can be very costly and/or risky for the success of the project.

Contrary to this, experienced project sponsors, managers, and/or engineers recognize the importance of this factor in a much better way. They skillfully select and consistently use principles related to the human factor in CSE projects. However, the recruitment of an experienced staff consumes a lot of time, effort, and money.

On the other hand, there are countries such as Japan or the USA, where the value of principles related to the human factor in CSE projects is recognized, even when it comes to those with little experience. Such an attitude helps in the implementation of CSE projects and increases the chances of success. The author noticed that human factor principles in the USA are different from those adopted in Japan. The issue has been signalized in the book in Sect. 16.3.6, entitled "POLTAX CSE Principles and Japanese Approach to Complex IT Projects for the Ministry of Finance in Japan."

Further in this section, we will mention principles which are mainly targeted at the work culture in EU countries, especially in the public sector and state-owned companies of East-Central Europe. In such a work culture, **human factor principles, adjusted to specific project constraints, are especially needed, but difficult to implement** in the realization of CSE projects.

The Principle for Providing an Appropriate Role of the Human Factor in Management of CSE Projects can be, to a large extent, reduced to a continuous implementation of such human factor principles as principles hf01–hf11, listed in Appendix B.

14.3.2.4 Principle of Providing an Appropriate Level of Communication

Numerous experienced managers of complex projects are aware of how difficult it is to provide an appropriate level of communication between various participants of a project. This problem has been commonly known for several decades. Some of its aspects are illustrated in Fig. 14.1. The figure is the adaptation of the presentation by Sydney Yoshida, entitled "Quality Improvement and TQC Management at Colsonic on Japan and Overseas," which was presented during the Second International Quality Symposium in Mexico (1989).[12]

In each CSE project, we can identify a range of different project roles assumed by stakeholders and project participants. The most common project roles are listed in points r01–r12 in Appendix C. However, it should be emphasized that the list is not complete. Its aim is to indicate that the same roles may appear in numerous different projects. At the same time, depending on the specificity of a project, there are numerous significant project roles that were not included in the list.

The quality of communication in a CSE project (especially between the representatives of project roles listed in points r01–r12 in Appendix C) is one of the basic conditions for the success of a given project. Nevertheless, inexperienced project participants do not give due weight to basic issues, which decide about the quality of communication in a project. Often, in such projects, managers, sponsors, and directors of a given organization neglect basic issues which affect the quality of communication. These basic issues are related to the answers to questions listed in points c01–c13 in Appendix C.

From the author's observations, it seems that when it comes to CSE projects, the quality of communication, which supports an appropriate aggregation of the current

[12]http://thethrivingsmallbusiness.com/small-business-problems/.

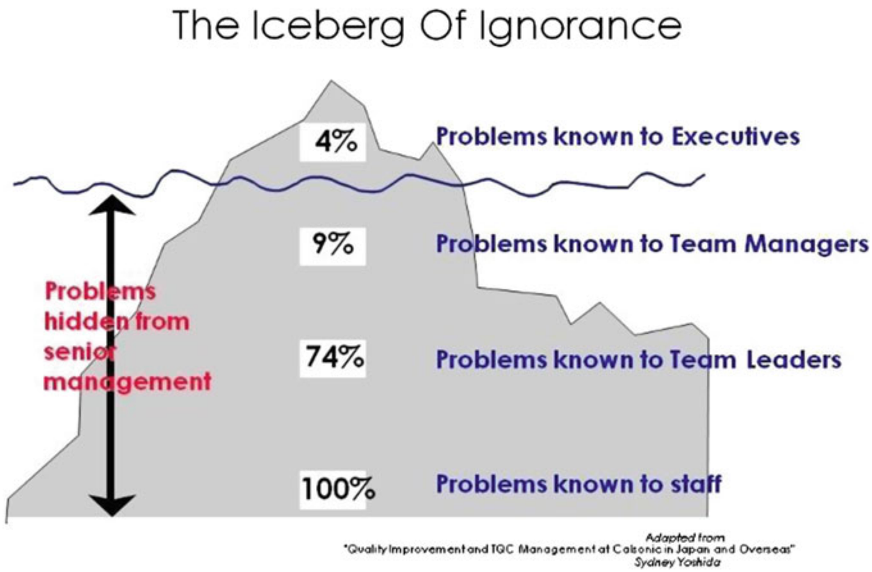


Fig. 14.1 Iceberg of ignorance of higher level management on the basis of "Quality Improvement and TQC Management at Colsonic on Japan and Overseas" presented by Sydney Yoshida at the Second International Quality Symposium, Mexico (1989)

knowledge (within the scope of points K1–K10 listed in Appendix C), is very often underestimated.

The scope and specificity of knowledge possessed by a given project participant should be adjusted to her/his project role. As a result, the scopes of knowledge that encompass the above-mentioned points (K1–K10) from the Appendix C are scattered among various project roles.

By the *Principle of Providing an Appropriate Level of Communication*, we mean arriving at and implementing the best possible (as far as the resources allow) mechanisms of communication that fulfill the requirements (cl-1)–(cl-4) listed in Appendix C.

The Principle of Providing an Appropriate Level of Communication—as other principles discussed in the book—was formulated at a relatively high level of generality. Therefore, it (like other principles) can be implemented in numerous ways.

Theoretically, it seems that the choice of such an implementation technique, taking into account the specificity and context of a project, is relatively easy. However, according to the author, there is more to this than meets the eye. Contrary to appearances, **even a very simplified implementation of the Principle of Providing an Appropriate Level of Communication in the case of a CSE project is an extremely difficult process and, moreover, it is often impossible to fulfill this principle to a satisfactory degree given the project constraints**.

Therefore, the principle can be treated as a sort of a strategic objective, which is difficult to attain. However, conditions listed in Appendix C ((cl-1)–(cl-4)) can be specified in the form of measurable indicators that determine the level of the quality of communication in a project. When determining this indicator, one may reassign the weights which characterize the fulfillment of criteria listed in points r01–r12 and K1–K10 in Appendix C. These weights should depend on the specificity and context of a project in its current state of implementation.

Luckily, in practice, the quality with which the aforesaid principle is implemented can be significantly improved by introducing numerous other principles that are more detailed and more properly selected. For example, by implementing the principle IV&V, described further in the book (cf. Sect. 14.3.2.5), one may significantly improve the quality of communication with regard to the most critical communication channels in a project. Moreover, in the case of CSE projects, it is virtually impossible to achieve a good-quality communication between a project manager and key project stakeholders without the implementation of the IV&V principle. At the same time, this is one of the many examples of principles, which hinder the improvement of communication in a CSE project.

Summing up, a satisfactory implementation of the Principle of Providing an Appropriate Level of Communication in a project is practically impossible without the support of other, specific principles (supporting, among others, the quality of communication). These supporting principles must be properly selected and adjusted to the current specificity and context of the project's implementation.

14.3.2.5 Principle of Independent Verification and Validation (IV&V)

The Principle of Providing an Appropriate Level of Communication, discussed in a previous section, is very difficult to implement. At the same time, for the success of each project, it is extremely important to ensure a good communication between a project manager, top management, sponsors, and end users of the end products.

Abiding by the old rule *nemo iudex in causa sua* (lat. no-one should be a judge in his own cause), we should avoid situations in which a project manager and her/his subordinates make and communicate judgments related to the current knowledge about a project (mainly within the scope of points K1-K10 from Appendix C) to sponsors and top management.

To objectify and increase the quality of communication between sponsors, top management, and a project manager, it is very important to implement the IV&V principle (cf. [90, 108, 285, 305, 321, 355, 500, 536]) independently from the project manager.

Basically, the IV&V principle supports broadly understood activities which are undertaken to minimize the risk of a project failure and ensure the required quality of the project's implementation (Quality Assurance, QA) [101, 129, 160, 261], along with allocating dedicated (independent of a project manager) resources (including human resources). Human resources realizing the IV&V principle are called the *IV&V team*.

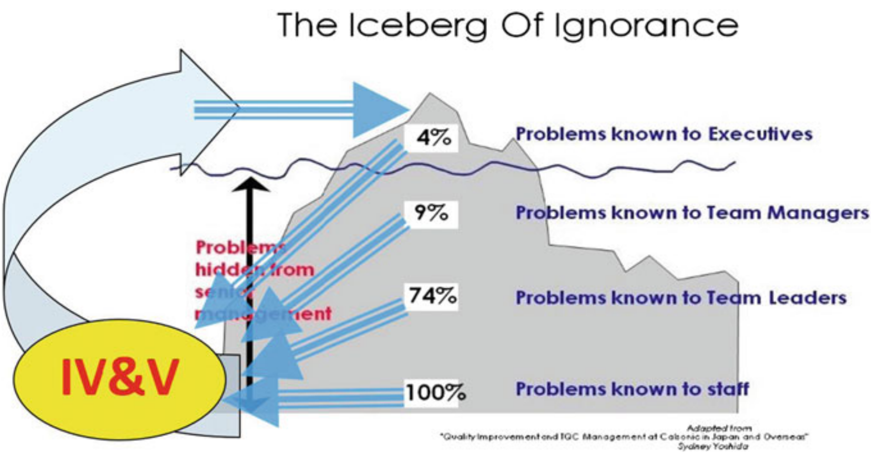


Fig. 14.2 Increasing of communication quality with *top* management in the project through IV&V team

During the project's implementation. **The IV&V team should ensure a continuous fulfillment**—at the highest possible level of quality—of conditions determined in Appendix D.

To simplify, one of the basic aims of implementing the IV&V principle is to **"unclog" the communication channels used for the communication between the top management of a given organization, project sponsors and managers, as well as the end users of end products and project results**. At this point, it should be added that if the implementation of the Principle of Providing an Appropriate Level of Communication in CSE projects caused no problems, then—from this point of view—the development and implementation of the IV&V principle would be pointless.

To illustrate the mechanisms which support the processes aimed at improving the quality of communication in a project by means of the IV&V principle, let us return to the Fig. 14.2, which shows the "Iceberg of Ignorance."

Unfortunately, as it has been previously emphasized, providing a satisfactory communication in CSE projects constitutes one of the biggest challenges that decide about the success of such projects. Without a doubt, an IV&V team helps improve the flow of information. However, this does not entirely solve the problem of communication in a project. As a result, very often, sponsors and top management of a given organization are not aware of the fact that **the critical, current knowledge, which falls within the scope of points K1–K10 (Appendix C), is not conveyed to them**. On this basis, stakeholders usually have a wrong perception of a project and its current state of implementation.

It should be emphasized that **the lack of satisfactory communication between the manager and stakeholders of a project does not usually stem from one's ill will or the actions of sabotaging project participants**. It rather results from the lack of adequate mechanisms for identifying problems during the implementation of a project and, if problems are identified, from a wrong prioritization of these problems (by a project manager). Consequently, a manager is not focused on issues that are critical for the success of a project. This requires “an urgent and ongoing” involvement of a project manager in solving current problems (that are, in fact, of a low priority), which only seem important.

Since sponsors and top management have no access to the properly aggregated current knowledge, which falls within the scope of points K1-K10 (Appendix C), **they make wrong decisions (based on false assumptions)**. As observed by the author of this book, this phenomenon often **leads to losses which exceed, a hundred times over, the costs of launching and implementing adequate IV&V principles**. It particularly concerns organizations that spend public funds, as well as state-owned companies.

In light of the aforesaid remarks, it seems justified to consider the introduction of a legal principle (e.g., in a form of a parliamentary act) that would enforce (especially on organizations that spend public funds and state owned companies) the allocation of, for example, 5–10% of the contract’s value for the purpose of financing the mechanisms for a continuous implementation of the IV&V principle in (publicly funded and state owned companies) CSE projects. In the best-case scenario, **the IV&V principle should be implemented by an entity that will assume some of the responsibility for a possible failure of a project within the scope of an approved catalogue of easily identifiable mechanisms and risks, which an IV&V team will fail to indicate**. At this point, a “project failure” is understood as determined in Sect. 14.3.1 in the PoF definition (points i.–iii.). From the author’s experience, it seems that in practice, costs related to the activities of an IV&V team bring savings in a sense that additional costs, stemming from a possible failure of a project, are avoided.

In some countries, recommendations of good practices concerning the IV&V principle were implemented in the public sector. For example, they are very common in the US administration (cf. [90, 108, 285, 321, 355, 500, 536]). In this context, the IV&V principle is understood mainly in the following way [108]:

> IV&V stands for:
>
> - Independent - assessments are performed by an independent third party
> - Verification - verifies that the product is well engineered
> - Validation - validates that the product conforms with client requirements.
>
> Maintaining independence of the verification and validation process is an essential element of the IV&V process. The Institute of Electrical and Electronics Engineers standard for Software Verification and Validation (IEEE Std 1012 - 2004) defines independence in IV&V using three main parameters:
>
> - Technical independence is achieved by IV&V practitioners who use their expertise to assess development processes and products independent of those performing the development.

- Managerial independence requires responsibility for the IV&V effort to be vested in an organization separate from that responsible for building and/or implementing the project's product. The IV&V effort independently selects the segments of the product to analyze and test, chooses the appropriate technique(s), defines the schedule of activities, and selects any specific issues to act upon
- Financial independence requires that the IV&V effort be funded from a general & administrative expense account in order to insulate the IV&V team from any potential financial pressures.

The aforesaid IV&V principle and an example of how it can be understood, provided above, are expressed at a relatively high level of generality. Therefore, there are numerous techniques to specify the specialization during implementation of this principle. **It is extremely important to include the context of the project's implementation, along with project constraints, in these techniques**. Contrary to appearances, it is a great challenge also for experienced managers as it is easy to make typical mistakes, which lead to:

1. the loss of the independence from a project manager or, quite the contrary, the loss of the independence and autonomy of a project manager herself/himself,
2. an unnecessary slowdown in the pace of the project's implementation due to excessive activities stemming from the overuse of the IV&V principle,
3. the imposition (by an IV&V team) of a wrong implementation direction on the top management of a given organization and project sponsors,
4. the use of the IV&V principle, by top management and/or a dedicated IV&V team, to jeopardize a project in a way that has no substantive justification.

In order to reduce the risk of errors while specifying and implementing the IV&V principle, one may use recommended good practices (which especially concern the public sector) from the USA (e.g., cf. [90, 108, 285, 321, 355, 500, 536]). However, too great a formalization of the techniques for specifying the IV&V principle in countries with no adequate traditions may significantly hinder the optimal adjustment of this principle to the specificity and context of a given project.

On the other hand, if the implementation of the IV&V principle in its specific form is performed by a sufficiently experienced team, the aforesaid risks are very low. Instead, there appear numerous potential benefits resulting from the implementation of the IV&V principle. A skillful compliance with the principle significantly increases the quality and value of the end products and results of a project.

Examples of mechanisms that can have a very positive influence on the implementation of a project are discussed further in the book and include, e.g., the mechanisms of enhancing the ontology and/or knowledge, and the influence of an IV&V principle on the paradigm shift [291] when solving project issues (by a team of engineers and/or experts hired in a project).

In Part IV of the book, we illustrate how to implement the IV&V principle with an example of projects that were conducted under the direction of the author (POLTAX, AlgoTradix, Merix, Excavio). These recommendations are relatively well adjusted to the specificity of work culture in the public administration and state-owned companies in East-Central Europe. The author, as a member of the Information and

Communication Council by the Polish Prime Minister, had a chance to directly experience **numerous benefits resulting from the use of the IV&V principle. At the same time, the majority of projects monitored by the author and implemented without the IV&V principle ended in failure (as understood by the PoF definition)**.

When discussing the IV&V principle, it is worth recalling that the idea of this principle did not evolve in the 20th century. The principle, under quite different forms, **has been rooted and used in numerous domains of engineering for a long time**.

For example, such principles have been known in construction engineering since antiquity. Over thousands of years, various traditions and good practices were developed to help reduce the risk of construction catastrophes, e.g., by a good implementation of mechanisms (the so-called **independent government's construction supervision**) and **legal requirements concerning construction engineering**.

In a sense, systems and agencies providing military intelligence services can also be treated as a specific form of the IV&V principle in military projects.

The examples signalized above show a long history of gaining experience in implementing various forms of the IV&V principle in civil and military engineering. On the other hand, the **significance of the IV&V principle in CSE projects is often underestimated**. Consequently, some sponsors and top management representatives still wonder why complex CSE projects (including IT projects), which are designed and constructed maintaining an "independent" quality supervision (which serves a metaphorical role of the "independent government's construction supervision"), almost always end in a "project's failure" (as discussed in Sect. 14.3.1 in the PoF definition).

During the last quarter of a century, the author had many chances to talk to numerous experts and influential politicians from parties that governed in Poland over the last decades (starting from 1990) about the need to implement the IV&V principle in the Polish public sector in the 21st century. Except for the Deputy Prime Minister Leszek Balcerowicz and Prime Minister Jan Olszewski, who definitely agreed to use the principle in the modernization of the Polish administration (thanks to this, e.g., POLTAX was developed and implemented), other interlocutors were basically "for and even against it" (therefore, for the last quarter of a century, we could not create a system that would supersede POLTAX). Only some agreed to make direct decisions to prepare for the implementation of the principle (however, if that happened, usually just before the implementation of the IV&V principle, there was a change of the governing party). Consequently, so far, the principle which was supposed to support the implementation of complex projects, especially those implemented by means of public funds and state owned companies, has not been implemented.

It is worth emphasizing that contrary to appearances, the principles related to IV&V are not as trivial as many budding managers and engineers think.

The essence of a modernly understood metaphorical "independent government's construction supervision" is not only limited to the independent verification, validation, and quality assurance of "physical" project deliverables. Moreover, it is not only about counteracting purposefully bad and/or dishonest practices of those who perform project tasks.

It relies on a subtle interactive "cooperation" of IV&V teams with teams of contractors, aimed at discovering the risks and co-risks of a given project (cf. Chap. 32). Next, the cooperation should lead to joint efforts in preventing the negative consequences of project risks and enhancing the positive results of co-risks.

The situation is even more complex as "supervisors" from an IV&V team, "contractors" that report to a project manager, and future end users of products usually use different ontologies. Despite this, it is necessary to maintain communication that will lead to some positive results. It requires the ability (of all entities engaged in communication) to use appropriate techniques for "approximate" reasoning about concepts and the ability to implement reasonings, based on the satisfactory approximations of concepts, as well as reasoning schemes, which are used by different entities.

In other words, **contractors are limited by concepts and/or paradigms [56] that they use, which are expressed in a language of these, often vague, concepts. This requires the use of advanced techniques for the approximations of concepts, as well as approximate reasoning techniques** (cf. [27, 33, 34, 218, 360, 457]).

As a result, a modern implementation of the IV&V principle should allow contractors to identify a place where it is possible to discover new concepts or practices and to support all activities related to this. To simplify, it means that an IV&V team should support the process of discovering, by contractors, the concept of a metaphorical "wheel," which is necessary to solve current problems in a more efficient way (e.g., facilitating the transport of big objects). Thus, an "independent" quality control should contribute to the best optimal (in a current context of the project's implementation) paradigm shifts [291], used by a project team. It particularly refers to the paradigms for constructing models of project requirements and project solutions.

In other words, an **IV&V team should not only help find basic errors (e.g., in a software code) or deviations from expected project requirements, but also some more sophisticated errors (** e.g., in ontologies, approximations of reasonings, or paradigm shifts). An important aspect in the modern approach to the IV&V principle is including it within the scope of the analysis aimed at identifying the motivation behind launching and implementing a given project, especially when it comes to the motivation of key stakeholders (including the end users of a given product). These motivations should be confronted with the current state of project requirements by an IV&V team.

In the approach proposed in this book, these important processes that were discussed above are treated as a particular case of IGrC comupting models. At the same time, the principles of quality control implementation (including the IV&V principle) constitute an important, but also a very specific example of problems related to constraints in IGrC computations.

14.3.2.6 The ABC Book Principle

Having in mind that "Rome was not built in a day," it is good idea to gradually identify, analyse, implement, verify, and adapt project implementation principles. Thus, it seems sensible to develop a principle, which could be treated as **a guide for the selection, adjustment, and implementation of families principles, in order to support CSE projects**. This guide is mainly addressed to less experienced managers, sponsors, and engineers. Nevertheless, in order to minimize project risks, more CSE experienced managers may also take it into consideration.

In the following parts of the book, the guide is referred to as the principle of the *ABC Book for CSE Projects* or the *ABC Book principle*, in short.

The essence of complying with the *ABC Book principle* lies in the **interactive and adaptive development, implementation, and verification of an optimal ontology of concepts**, necessary to implement a given project and to express principles (expressed in a language of these concepts), which improve the efficiency of the project's implementation.

According to the ABC Book principle, a family of principles implemented in a project **should by developed in an interactive (according to the PDCA cycles) and evolutionary (not revolutionary) manner**. This means that one should not implement (especially in new and not yet recognized CSE projects) any complex family of principles (expressed in a language of concepts which is not adjusted to the specificity of a project) in a rapid manner.

The evolutionary nature of implementing the ABC Book principle means that along with selecting, discovering, developing, and improving the way principles are obeyed in a given project, we try to use possibly the most optimal family of concepts, characteristic of a given project domain. This optimization of concepts takes into account multi-criteria optimization. The basic criteria recommended for the selection of concepts include: the usability of concepts for expressing project objectives in an understandable and condensed manner, the usability of concepts for expressing the techniques used to implement a project and solve problems, the understandability of concepts by adequate groups of project participants, the satisfiability of expected project requirements (especially in terms of following terminological standards and language recommendations).

In reference to a new project, it is recommended to start obeying the ABC Book principle from selecting appropriate concepts (comprehensible to key domain experts and project stakeholders) and using them to describe a project.

Next, it is recommended to gradually add details in order to interactively adjust (to the project's specificity) more complex principles. Of course, this adjustment may mean that only some recommendations from these principles will be taken into consideration.

For a given a CSE project, the *ABC Book principle* contains adaptive principles for monitoring and/or controlling processes which cover the following minimal scope:

a^* **Objectives and project requirements**. It represents the adaptive principles for monitoring and/or controlling processes for continuous learning, verification, validation, and/or updating of important knowledge about: stakeholders, their motivations, project objectives, project requirements, project constraints, Key Performance Indicators (KPI),[13] and ontologies supporting project communication.

b^* **Plans, principles (including principles of change control), and Critical Success Factors (CSF)**[14] (adjusted to current knowledge in the scope of a^*). It represents adaptive principles for monitoring and/or controlling the processes for continuous learning, verification, validation, and/or updating of important knowledge about:

i. *project implementation plans* with highlighted *milestones and their acceptance criteria*,
ii. a family of principles to obey as part of the project's implementation (including standards, practices, and techniques), we assume that:
    1. these principles should support effective and efficient implementation of the project leading to the successful achievement of the project objectives,
    2. these principles may use in their description some specific methods, tools, and more generally techniques, which are defined in other places,
    3. for any CSE project we should have *configuration management principles*,[15,16]
iii. *Project Critical Success Factors* (CSF).

c^* **Nomination of the project manager, implementation of organizational and communication systems, and appropriate resource allocation** (adjusted to current knowledge in the scope of a^* and b^*). It represents adaptive principles for monitoring and/or controlling the processes assuring proper and documented resource allocation and management (according to a^*–b^*). In particular, it includes appointment of a project manager, along with organizational and communication systems, and documented allocation of resources for the project's implementation (according to a^*, principles, and plans specified in b^*), along with adequate roles and responsibilities. We assume that *project manager is responsible for effective and efficient implementation of the project* leading to the successful achievement of the project objectives.

d^* **IV&V Principle implementation** (adjusted to current knowledge in the scope of a^*, b^*, and c^*). It represents adaptive principles for monitoring and/or controlling the processes assuring proper implementation of the IV&V principle (adjusted to current knowledge in the scope a^*, b^*, and c^*) and techniques for assessing the effectiveness and efficiency of the IV&V implementation.

[13] https://en.wikipedia.org/wiki/Performance_indicator.

[14] https://en.wikipedia.org/wiki/Critical_success_factor.

[15] https://en.wikipedia.org/wiki/Configuration_management.

[16] http://www.sei.cmu.edu/productlines/frame_report/config.man.htm.

The correct implementation of the *ABC Book principle* for CSE projects assumes that the top management enforces optimal and documented decisions in the scope of a^*–d^*.

Budding managers have problems with understanding and recording the current state of knowledge about a given CSE project, defined and/or implemented, with regard to points a^*–d^*. At the same time, they are often not aware of the fact that **this significantly reduces the quality of communication** in a CSE project. As a result, project participants have limited possibilities to cooperate and this decreases the chances for making sound decisions and/or actions concerning a given project.

At this point, it is worth reminding that a **decision about launching a CSE project without performing a valid feasibility study is a common failure principle**. Decision-makers who plan a successful implementation of a project should be obliged to make decisions about launching a given project on the basis of conclusions drawn from a valid feasibility study. Having in mind this, let us also notice that it is impossible to perform a valid project's feasibility study without an appropriate understanding and/or recording of the currently "expected project requirements."

Moreover, if current project requirements are not documented, than during the project's implementation, it is very difficult to fulfill them.

Apart from the interactive specification and development of a project itself, the specifications of all important aspects of a project should be consistently expanded, developed, and improved. For CSE projects the specification of points a^*–d^* can be complex task. Thus it is good idea to start the specification using at the beginning specification of some kind of simplification of the points a^*–d^*. For example, the simplification can be presented in the form of the following points a^-–d^-:

- a^- **Objectives and requirements**. It is documented current knowledge in the scope of project objectives, requirements, and glossaries.
- b^- **Plans, principles (including principles of change control), and CSF**. It is documented current knowledge in the scope of project action plans,[17] project principles, milestones, CSF, and project change control procedures.[18]
- c^- **Nomination of the project manager, implementation of organizational and communication systems, and appropriate resource allocation**. It is documented current knowledge in the scope of appointment of a project manager, specification of organizational structures, and allocation of resources for project implementation.
- d^- **IV&V Principle implementation**. It is specification of IV&V Principle Implementation and current reports of IV&V team.

For the CSE projects, the points a^-–d^- should be treated as the first step in order to specify, implement, and develop the a^*–d^* points.

The specifications and modifications of the project documentation presented above should encompass both current and expected key aspects related to the specificity and context of the project's implementation. It is very important to follow ade-

[17] https://en.wikipedia.org/wiki/Action_plan.

[18] https://en.wikipedia.org/wiki/Change_control.

quate procedures of introducing changes when making these modifications (along with analyzing the consequences resulting from introducing changes and informing about these changes).

Generally, for experienced project managers and stakeholders, a valid documentation and implementation of aspects described in points a^*–d^* are obvious. Moreover, for experienced CSE project managers it is easy for implementation. However, in practice, these aspects are overlooked and/or neglected in numerous countries (especially in the public sector and state-owned companies).

In other words, in in numerous countries, it happens that decision-makers launch projects which fulfill the *ABC Book failure principle*, meaning that at least one of the following conditions is met:

- $\neg a^*$ We don't have important updated and documented knowledge in the scope a^*.
- $\neg b^*$ We don't have important updated and documented knowledge in the scope b^*.
- $\neg c^*$ We don't have proper implementation of resource allocation and management structures (as defined in c^*).
- $\neg d^*$ We don't have proper implementation of IV&V principle (as defined in d^*).

Summing up, not keeping records of the current state of knowledge within the scope of points a^*–d^* affects the implementation of a CSE project, leading to its failure.

Based on the analyses of the author (especially with regard to the course of projects that were monitored by him and implemented under his direction), following the ABC Book failure principle significantly increases the risk of failure in the case of CSE projects. At the same time, this leads to an unjustified waste of significant resources (of course, this usually involves public funds that are spent "according to the binding procedures," e.g., in line with the Public Procurement Act).

On numerous occasions, the author used the ABC Book principle discussed above, constructing and developing complex families of self-complementing principles for the implementation of CSE projects around the aspects mentioned in points a^*–d^*. This particularly concerns projects described in Chaps. 16–20.

Simplified versions of the ABC Book principle and the ABC Book failure principle with regard to the POLTAX project were discussed by the author of this book in an interview published in [280].

14.3.2.7 Principle of an Appropriate Document Management

The implementation of each complex CSE project relies on *basic documentation*, which includes documents that have been formally approved and updated by competent decision-makers in accordance with the binding principles (including principles of introducing changes). As a project progresses, the documents should be developed and managed in an evolutionary manner.

The scope of the documentation should depend on the complexity, specificity, and constraints of a project, its phase of implementation, as well as the level of work culture in a given organization. In general, at the beginning of a project's implementation, the documentation is quite modest. Next, it should be consistently expanded and developed along with the adaptive needs of a project and its evolution.

In the case of typical CSE projects, from the very beginning of conceptual works on a project, the basic documentation should define, in a transparent way, the current state of knowledge about a planned and/or implemented project. In general, this state of knowledge encompasses aspects identified in points a-d in Appendix E (cf. also Section on ABC Book Principle (cf. Sect. 14.3.2.6).

In the case of complex projects whose total cost (the cost of using internal and external resources) exceeds an approved threshold (e.g., several millions euros), it is recommended to introduce, possibly from the very earliest phases of project works, a greater specificity when it comes to the minimal content of the basic documentation. For example, the documentation may be prepared according to the scheme presented in points A, B, C, and D in Appendix E.

The *Principle of an Appropriate Document Management* means that along with the development of a project, a current version of basic documentation is appropriately collected, validated, versioning, distributed, and archive. The current version of documentation should be as compliant with the current state of the project's development as possible. The documentation should be formally approved and updated by competent decision-makers in accordance with the binding principles. Its scope and form should be adjusted to the specificity and context of the project's implementation.

Generally, with relatively simple CSE projects, it is recommended that the basic documentation encompass at least points a.-d. mentioned above.

In the case of very big projects, it is recommended to keep records which encompass aspects from points A, B, C, and D, presented in Appendix E.

Both in the case of simple and more complex CSE projects, one ought to remember that the *basic project documentation* should additionally include *technical project documentation*. This type of documentation depends on the technique and standards used for the implementation of a project. As a result, technical documentation should be specified as part of the *partial project deliverables*. In general, project documentation, which is part of technical documentation, encompasses points td1–td9 in Appendix E.

It should be underlined that not all experienced managers and/or engineers are enthusiastic about following the Principle of an Appropriate Document Management (especially in small projects). Among them, there are also ardent supporters of conducting project in an *agile style* [93, 204, 473, 489]. Some of them believe that the Principle of an Appropriate Document Management with regard to non-complex projects constitutes a failure principle.

The author, based on his experience, noticed that the compliance with the Principle of an Appropriate Document Management to a large extent depends on the organizational level of an institution which conducts a given project, the degree of work culture and professionalism of project teams, and specificity and context of the project's implementation. For example, if the Principle of an Appropriate Document Management is not properly obeyed when implementing CSE projects in a work culture characteristic of the public administration in Central and East Europe, it leads to billion-euro losses within one decade. On the other hand, the author thinks that the Principle of an Appropriate Document Management should be adjusted and used when implementing any CSE project. Still, when specifying the techniques for implementing this principle, one must remember about adjusting the implementation to the current project constraints. The principle can be specified in numerous ways. In the case of very big projects, it is recommended to implement an electronic system for managing project documentation (including basic documentation), developed around an easily accessible, properly designed Project Wikipedia.

It should be emphasized that an appropriate compliance with the Principle of an Appropriate Document Management is difficult, as in particular, it requires providing optimal answers to the following questions:

1. What should be the minimal scope of documentation, from which to start recordkeeping in the case of CSE projects, and which principles to follow when developing and improving project documentation?
2. What is the optimal scope of documentation at a given moment of the project's implementation?
3. In what form should records be kept and how to manage changes in documentation (including how to negotiate and communicate changes)?
4. In what language (including language for modeling requirements and iterative prototyping of the process of arriving at solutions to the main project problems) should records be kept in a project, in which experts from numerous unrelated areas of specialization participate?
5. How to minimize the cost of "translating" documents expressed in hermetic languages, that are incomprehensible to important sub-groups of a project team?
6. How to evaluate the usability of project documentation and how to establish priorities when it comes to recordkeeping?

In the case of CSE projects, all questions mentioned above are very important and difficult to answer. It is even more difficult to answer them, if a project is unique and innovative (at the moment of its implementation). Less experienced project managers neglect to think of how to find optimal answers to these questions. They use "the best"—according to them—technique and believe that it will provide good enough solutions to the issues mentioned above.

The author had a chance to monitor projects in which the lack of proper answers to Question no. 1 led to a failure. Having in mind the aforesaid difficulties, the author developed the Principle of the The ABC Book Priniciple (cf. Sect. 14.3.2.6). It is discussed in the next section and its aim is to indicate a way in which it is possible to some approximations of the answer Question no. 1.

14.3.2.8 Fundamental Survival Principle

When implementing large strategic CSE projects in technically mature organizations, it is thought of as necessary to follow and mutually synchronize the following principles:

- Principle of Anchoring a Project in Top Management.
- Principle of Continuous Identification, Judgment, and Adaptation of a Family of Currently Used Principles.
- Principle of Providing an Appropriate Role of the Human Factor.
- IV&V Principle.
- Principle of Providing an Appropriate Level of Communication.
- ABC Book Principle.
- Principle of an Appropriate Document Management.

There are numerous techniques for using and following these principles and the lack of synchronization in their implementation may become a dangerous failure principle. On the other hand, if skillfully synchronized, the implementation of these principles results in a powerful synergy, which may be a deciding factor when it comes to the success of a difficult and complex project. Unfortunately, these seemingly simple principles are rarely comprehensible to the actual decision-makers. Even less people are aware of a need to synchronize their implementation.

In order to emphasize the importance of this, we will introduce a principle that will draw our attention to the necessity of following the process of adaptive integration and improving the synergy of the seven principles listed above. The principle will be referred to as the *Fundamental Survival Principle*. Depending on the specificity of a project and its context, the principle (like its constituents) may be specified, implemented, and improved in numerous ways.

14.3.2.9 Basic Principles of CSE Project Development (BPCD)

In the Part IV of the book, Chaps. 16–19, we present some proposals for directions of a further specification of the ABC Book principle, adjusted to the specificity of particular projects. These principles constitute a unique form of knowledge representation under changing conditions of the project's implementation (its specificity and context).

Apart from very specific principles that constitute the representation of specialist knowledge about project constraints and related recommended actions, there are also numerous universal principles which may be used when constructing and developing principles according to the ABC Book principle. These universal principles can be adjusted to numerous types of projects. In order to illustrate them, we will present a family of principles for the development of CSE projects, which the author recommends using along with the ABC Book principle. This recommended family of principles is referred to as the *Basic Principles of CSE Project Development* or, in short, the *BPCD principles* (cf. bp01–bp25 in Appendix F). These principles are evolutionarily introduced and consistently adapted to the current specificity and context of a project. A recommended family of BPCD principles may, during the initial phases of a project, encompass principles proposed in Appendix F.

A family of BPCD principles constitutes a simplified overview of the author's selected conclusions, drawn from his several decades of practical experience in managing and running projects. The principles were expressed in a language comprehensible to experienced managers and/or engineers who work in CSE projects. Still, the principles are often not obvious for inexperienced CSE project leaders. Moreover, in numerous cases the aforesaid principles are not well understood by project sponsors. Unfortunately, in such situations, the total cost of a project very often exceeds the estimates, which leads to a project failure (as understood by a PoF definition in Sect. 14.3.1, points (i)–(iii)).

The BPCD is a very condensed and simplified overview of the author's most important conclusions from his practical experiences. It is expressed in a language for experienced CSE engineers. Hence, in general the BPCD should be easily understood by experienced CSE project leaders. Moreover, the main recommendations of the BPCD have been widely known in IT for several decades. However, typically they are not as clear for non-experienced CSE project leaders and, in many cases, they are not accepted by many sponsors. Hence, the total cost of a project is usually much higher than it was initially planned. According to some sources (e.g., [279]), **one needs approximately 20 years to become an experienced CSE leader** (cf. Fig. 8.5). Usually, the leaders of CSE projects do not have 20 years of experience in CSE. Having this in mind, author prepared some kind of a "simplification" of the BPCD. It is especially addressed to the beginners in CSE (especially to project leaders and project sponsors). It is called the *ABC book for CSE projects* (cf. [280] and Sects. 14.3.2.6 and 16.3.5). In some sense, the BPCD can be treated as an extension (a more advanced version, which assumes basic understanding of the ABC book) to the ABC book for CSE for experienced CSE project leaders. Thus the BPCD is addressed to experienced leaders and/or sponsors of CSE projects.

14.3.3 Examples of Dedicated CSE Principles Concerning Projects Described in Case Studies

Some of the principles described in Chaps. 16–19 are characteristic of a given application domain and specific conditions of the project's implementation. These principles are sometimes referred to as *dedicated principles*. These are the principles which mainly concern the works of project teams on developing the architecture of a given technical solution and creating the end products of a project. They are constructed using a specific domain knowledge related to the implementation of a given project.

Principles presented in Chaps. 16–19, which concern particular projects, **were created by using, detailing, and specifying appropriate versions of BPCD principles based on the ABC Book principle**. Next, the principles were developed, specified, and adapted to the changing specificities and contexts of projects. In cases justified by project constraints, the author developed new principles in order to satisfy the current project requirements determined by the changing specificity and context of the project's implementation.

The most significant CSE principles are presented in this book in chapters devoted to particular projects (case studies) in Part IV (i.e., POLTAX—Sect. 16.3, AlgoTradix—Sect. 17.2, Merix—Sect. 18.2, Excavio—Sect. 19.2, and UAV—Sect. 20.2).

14.4 Research Motivations Related to the Selection of CSE Principles

For experienced managers, the following observation is obvious:

The better the selection and adaptation to the current constraints of a project, including its specificity and context) of a family of CSE principles to obey when implementing a given project, the smaller the risk of failure.

The family of principles for a new CSE project should be selected by means of an adequate strategy and should be continuously adapted to the changing specificity and context of the project's implementation.

All principles are expressed in an appropriate fragment of a natural language. This *language of principles* uses complex vague concepts and their relations between them. **The subtler the language of principles is, the easier it is to express valuable principles with it**.

Choosing an optimal language of concepts and relations between them constitutes a central problem of scientific research in numerous domains. From the point of view of the machine learning of new concepts, the problem commonly occurs in such domains as extraction and knowledge discovery,[19] ontology engineering,[20] learning of ontology approximation [27, 31, 33, 363], feature selection [27, 31, 33, 363], inducing (extracting) new features (e.g., discretization of continuous features,[21]) construction of classifies (e.g., [27, 29, 30, 119, 419, 425, 470, 541]).

Experienced managers are well aware of the fact that it is difficult to choose an appropriate language of principles and express in it an optimal family of principles for a CSE project. They know that **using a metaphorical "wheel" requires one to understand and discover techniques of using this concept in practice** (in particular in the context of other language expressions). Therefore, they treat this as a big challenge.

Very often managers construct and adapt a family of principles for a new project, using their experiences. As the complexity of CSE projects grows, so does a need for a better systematization of knowledge about the principles that could support the selection of CSE project implementation principles.

The approach proposed in the book consists in an interactive specification and adjustment of principles to the specificity of a given project in line with the ABC Book principle and, in a broader perspective, with BPCD principles. **At the same**

[19] https://en.wikipedia.org/wiki/Knowledge_extraction.

[20] https://en.wikipedia.org/wiki/Ontology_engineering.

[21] https://en.wikipedia.org/wiki/Discretization_of_continuous_features.

time, in our approach, the implementation of a complex project is treated as a unique IGrC computational model. In such a case, principles may be treated as interaction rules, which are the constituents of interaction plans. In this way, we try to identify and use mutual, potential synergies from progresses in the research on CSE project implementation and models for controlling computations in IGrC (cf. Chaps. 21 and 31).

For example, from the point of view of controlling computations in IGrC, the IV&V principle may be understood as, e.g., verification and validation of the project's status and forecasting its future on the basis of the verification and validation by a society of agents, which have a specific ontology thanks to their experiences in the implementation of numerous projects. Thanks to this observation, the experiences in using the IV&V mechanisms, in particular, mechanism off "construction supervision" may be transferred onto the mechanisms of controlling computations in IGrC.

Computations in IGrC should satisfy less or more complex constraints. The degrees to which these constraints are satisfied may be verified and validated by dedicated agents. If there is too great a risk of deviations from the desired satisfiability degree of certain conditions (concepts), agents may launch actions to minimize the negative consequences of these risks and return to the acceptable level of risk. In this context, it is important to understand in what way such a cooperation of agents may help improve the efficiency of controlling computations to obtain the desired results. This cooperation should be based on non-trivial dialogues among agents (including human experts). These dialogues are often held between agents who use different ontologies. Therefore, there is a need for further and more detailed research on negotiations, dialogue, etc. (cf. [26, 105, 216, 283, 388, 435, 529, 532]).

The Excavio project, presented in Chap. 19, refers to the research on supporting the dialogue between human beings and a network of computers. One of the aims of the project was to make an attempt to implement the mechanisms of an "intelligent" dialogue among agents (including human experts) and computer network system. Such dialogue systems based on IGrC computations have numerous potential applications. What is particularly important, they may increase the efficiency of the project's implementation and open the possibilities of constructing new generations of systems for implementing the IV&V principle in the future.

As the complexity of CSE projects grows, so does the risk of failure [54, 368]. Therefore, effective techniques of implementing the Principle of Providing an Appropriate Level of Communication and the IV&V principle will become increasingly important. In the future, computational IGrC models may be used in systems for supporting communication within project teams and among teams that use various dialects. For example, project participants will be able to communicate with a computer system by means of a dialogue.

The aim of this dialogue will be to generate the "approximation" of concepts and knowledge (expressed in a language of these concepts). Next, these concepts, by means of relevant approximations, could be communicated to other project participants and/or other computer systems. The development of mechanisms for conveying the reasoning of one of the parties involved in a dialogue to the other party is another challenge.

The essence of this challenge is reflected in the motto of the book concerning the need for constructing a language that will be "friendly" to a language used in research. Such a language should make it possible to express knowledge about cause and effect relationships and be comprehensible to various types of agents (including human beings and artificial agents). Moreover, it would be possible to add to this the knowledge acquired by, e.g., sensors that measure progresses in the implementation of a project and knowledge resulting from simple reasoning of a system and/or aggregations of numerous fragments of knowledge.

Unfortunately, for budding project managers, these simple rules do not carry a sufficiently significant semantic burden. Often, in such situations, numerous problems arise, both technical and organizational, which in turn leads to the so-called "death march" [545].

In other words, everyone with no adequate technical knowledge or skills will face difficulties when implementing technical solutions under project constraints caused by the so-called "Bermuda Triangle," which encompasses:

- technical objectives motivated by business objectives,
- technical possibilities and available resources,
- continuous adaptation of a family of currently used CSE principles to the perceived specificity of a project.

The situation becomes even more complicated when we take into consideration the fact that the vertices of this "Bermuda Triangle" must include project constraints which change in real time. What is worse, even the meanings of concepts in a language, in which the conditions of the—vertices are expressed, undergo constant changes. Therefore, even at this stage we must face typical problems of IGrC computations related, e.g., to *concept drift in data mining*.[22]

The most common mistake made during the initial phases of the project's implementation is determining a technically incorrect implementation of CSE principles and/or failing to take any steps to create any principles, which will be in force during the implementation process. With complex projects, such a behaviour hinders the success of a project.

It is also very important for such principles to include changing domain knowledge and significant aspects of the context (related to the specificity of implementing particular complex projects) in which a project is implemented. The lack of adaptation mechanisms, which take into account the development of a language in which the domain knowledge is expressed and changes to the knowledge itself, causes that following such principles may become dangerous.

To illustrate the above remark, it is worth recalling a large project conducted in the 70 s at the Stanford University—MYCIN.[23] It was then when one of the first big expert domain systems, in which knowledge is represented by means of rules [73], was created. The project resulted in a failure, mainly because of the **lack of effective mechanisms for the adaptation of knowledge, represented by rules implemented**

[22]https://en.wikipedia.org/wiki/Concept_drift.

[23]https://en.wikipedia.org/wiki/Mycin.

in the system. After a relatively short period of time, knowledge represented in this system became outdated and worthless, or even misleading. Still, the time and money allocated for the "manual" adaptation did not provide the economic and objective justification for the continuation of the project, using this very form of knowledge representation.

When implementing the current versions of the MYCIN system, it is worth remembering about the "openness" of the computational world and the impossibility to predict all results of interactions. One should also remember about various shortcomings of the "current" state of domain knowledge, which will never be perfect. At the same time, adaptation principles for the CSE principles (e.g., in a form of mechanisms for improving the principles by means of schemes compliant with the Deming cycle[24]—Plan-Do-Check-Act or Plan-Do-Check-Adjust) (PDCA) becomes very important.

A deeper understanding of the causes that lead to a crisis in the implementation of complex projects requires a deeper scientific understanding of the computational basics of complex systems, especially in the case of undertakings in which complex projects or systems are implemented. One should take into consideration the fact that these computations are performed based on the perception of situations, conditioned by interactions between project participants and other project constituents and by interactions with the dynamically changing surroundings.

From the point of view of some further reflections presented in the book, it is worth noticing an **analogy between learning survival principles in CSE projects and learning the rules of conduct in interactions with the surroundings**. Such analogies refer to the research by Valiant in [521], where he claims that:

> Learning becomes more powerful, and indeed indispensable,
>
> (i) whenever we cannot specify explicitly the outcome we want,
> (ii) whenever we cannot specify exactly what the system that is to execute it knows already, or
> (iii) whenever we cannot get direct programming access to the system.
>
> When the system in question is a human, all three of these conditions hold, and we have no alternative to learning.

The aforesaid analogy **constitutes one of the most important motivations of the scientific research presented in this book. This direction is related to the potential synergies of the research on techniques used for the implementation of CSE projects and computational models in IGrC**.

The motivation are discussed in more details in Chap. 21 entitled "Case Studies Conclusions: Toward Science-Friendly Languages for Interaction Rules and CSE Principles." Scientific studies motivated by this analogy will be described in the following chapters. A concise summary of the most important scientific results is presented in Chap. 36 and a family of recommended postulates concerning computational models in IGrC is provided in Chap. 32.

[24] https://en.wikipedia.org/wiki/PDCA.

Learning the survival principles and/or failure principles in a CSE project meets the conditions i.-iii. in the citation presented above. It shows how valuable and significant it is for a currently implemented project to use empirical principles, which have already been proved in numerous projects of a similar specificity and context.

The above remarks emphasize a particular role of the mechanisms for learning principles of interactions between agents (cooperating and/or competing). These principles are often discovered as a result of data interaction, supported by domain knowledge. Taking this into consideration and the incompleteness of information about the physical world where interactions occur, which is available to an agent, these principles are uncertain. Consequently, this means that they have to be constantly adapted to take into account the results of gaining new knowledge or new experiences. A deeper discussion about the study based on relations between CSE principles and scenarios of controlling computations in IGrC can be found in Chaps. 21, 28, 31–36.

Chapter 15
TPGP: WisTech as a "Silver Bullet" for Interactive Approximations

True wisdom comes to each of us when we realize how little we understand about life, ourselves, and the world around us.

Socrates (469 BC–399 BC) (www.brainyquote.com/quotes/quotes/s/socrates391046.html).

As shown by numerous reports on the implementation of complex projects and its effects, there is still no "Silver Bullet" for CSE which could effectively help achieve success. At the same time, increasingly advanced methods of implementation as well as more complex recommendations and standards are being created. Each practitioner has its own view on the subject. However, it is difficult to single out one common view to help increase the chances for the success of a complex project. While agreeing with Frederick P. Brooks who claims that there is "No Silver Bullet" [68], it is worth noting that seasoned professionals are far more successful than those without experience. This means that there is a significant area of knowledge and wisdom, possessed by the people with experience. At the same time, we have acute problems with the awareness and comprehensible description of knowledge and wisdom in a language understandable for the new adepts of complex engineering system. The approach proposed in this book boils down to explaining and imparting wisdom on complex engineering system, based on the paradigms of WisTech, with a particular emphasis put on its implementation paradigms, using rough sets [244, 245, 247, 253, 448]. The starting point of this approach is the Wisdom Equation (cf. Eq. 8.1). In essence, this equation refers to the concept of wisdom as it was understood in ancient Greece. As we remember, at that time, it was stressed out that a human mind has a limited ability to acquire absolute wisdom. Pythagoras (6th century BC), when asked whether he was wise [sophos], answered: "No, I'm just a lover of wisdom and [philo-sophos]." Thus, he emphasized that true wisdom is generally not available to

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_15

people. People only have access to some kind of an approximation of true wisdom. Consequently, engineers (as engineering-wisdom lovers) only have access to some kind of an approximation of true wisdom, expressed by the mythical "Silver Bullet" for CSE. Of course, this approximation accessible to people is becoming more complete and more valuable when wiser people begin to share their wisdom during discussions that lead to common conclusions. The previously cited metaphor related to an elephant (cf. Fig. 10.3) means that individual perceptions (approximations) of reality may be contradictory, but they should be aggregated by a dialogue (discussion, exchanging arguments) that leads to an agreement, resulting in a common and logically consistent perspective. Consequently, this means that the wisdom of the ancient Greeks was often understood as the reflection of the collective wisdom, corresponding to the current state of discussion and exchange of arguments between participants in the society, based on a rational argumentation.

However, such an argumentation is not always possible. Then again, a reference is made to the mechanism of the decision-making process, based on a team of agents rather than on a single agent. The judgment of the team may be based on the internal political system or the rules of a metaphorical court.

Thus, the wisdom is based on the dynamic adaptive judgment, rules for conducting a discussion and common knowledge available at a given time. The driving force that triggered the expansion of wisdom in ancient Greece was a great skepticism in disputes, along with a need to use increasingly sophisticated techniques of discussion and substantive arguments. In the works by Plato, we find the representation of wisdom in the form of dialogues between conflicting views. Aristotle in particular developed and systematized the techniques for conducting effective dialogues that lead to the development of common wisdom. Another very important idea taken from ancient Greece and introduces by Heraclitus was that everything around us, including us, keeps changing ("The only constant in life is change"). Surprising as it may appear, in the 21st century, the thoughts of the ancient Greeks are validated anew in numerous ways.

The quality of cooperation between the society of agents is influenced by many factors. However, communication language seems to be especially crucial. It should not be too universal, but at the same time, it should be flexible and capacious enough for the content that is being transmitted. For example, communication between the users of particular functionalities of a given system and the analysts identifying their requirements should be understandable for both groups involved in a project and, at the same time, as compact as possible. If the analysts perfectly understand the user requirements, but describe them in a hermetic specification language of requirements on several thousand pages, it is likely that the documentation will be rejected by the users. Often, the situation is even worse in practice, with analysts generating large studies that describe the ill-identified user requirements which in turn, are unfortunately accepted by the users. This may be due to the fact that the specification requirements were misunderstood by the users or that they were put under the pressure of time. In such a case, the failure of a project is almost certain. This means that the language of communication is extremely critical for the effective cooperation between the analysts of requirements and prospective end-users. This example can be extended to the collaboration between other groups participating in complex projects, as well as to the collaboration between human beings and computing devices transforming project-related knowledge. This shows the significance of understanding the mechanisms that lead both to the construction of an appropriate communication language between different members of project teams and to the human knowledge processing in computers.

Referring to the motto of this book, expressed by Lotfi Zadeh, a special role in increasing the chances for a positive implementation of a project is played by a suitably chosen environment for CWW, which today is often called GrC when granules are labeled by words from natural language. In other words, the success of a project depends largely on the choice of both granules representing concepts, related to the implementation of a complex system, and the methods for the processing of these granules (enabling effective risk assessment as well as appropriate domain knowledge granulation and distribution).

Of course, GrC on its own is not sufficient for a successful implementation of complex projects because the success of such a project depends on a number of human factors. Particularly important are the following human factors:

a. Motivations of the individuals involved in a project (especially the stakeholders),
b. The ability to understand a necessary domain knowledge as well as its technicalities (especially, the ability of professionals to understand the domain knowledge of a system that is being developed and the ability of domain experts to understand the "everyday" language used by those professionals).

These two human factors are often overlooked in a variety of methods, standards and recommendations developed for the implementation of complex projects. What is even more striking is that they are overlooked despite the fact that all project managers are aware of them. Unfortunately, they often fail to take very simple remedial actions.

Apart from the sphere of human factors, much more should be done for the development of advanced interactive methods of GrC in order to create efficient tools, supporting the effective implementation of complex projects. Certainly, understanding the mechanisms that allow us to access the concepts important for the field of CSE, along with the mechanisms that optimize the processing of these concepts, have a significant impact on the rapid adjustment of the overall experience and knowledge to the specificity of particular projects that are being implemented.

In the remaining part of the book, IGrC models, based on rough sets and WisTech, will be presented (especially Chaps. 33–35). This proposal is not a "Silver Bullet" for solving the problems related with the implementation of complex projects. Rather, it is a proposal towards a **discovery of fundamental laws that govern the implementation of complex projects using interactive approximations of important (for the success of the project) complex vague concepts**. In other words, it is an approximation of a potentially ideal and universal method, by means of interactive models based on GrC and WisTech.

Part IV
CSE: Case Studies

Chapter 16
POLTAX

> *Interaction is a fundamental dimension for modeling and engineering complex computational systems. More generally, interaction is a critical issue in the understanding of complex systems of any sort: as such, it has emerged in several well-established scientific areas other than computer science, like biology, physics, social and organizational sciences.*
>
> Omicini A, Ricci A, Viroli M [372]

16.1 Introduction to POLTAX

We use the word POLTAX in the following two meanings:

1. **POLTAX as an IT system** is an integrated family of cooperating and evolving subsystems built and deployed according to the system mission, objectives, and obeying *fundamental POLTAX principles* (cf. Sects. 14.3.1, 14.3.2 and 16.3) of implementation, exploitation, and development.
   The mission of POLTAX system is an effective, efficient, and secure support of modernization, operation, and evolution of the tax system in Poland, following an in-depth transformation in the country, which occurred at the beginning of the 1990s. During this period, the country went from a centrally-planned socialist economy system to a market economy conformed to the EU standards.
2. **POLTAX as an IT project** is a family of long term and short term interaction plans which support implementation of the system POLTAX mission, objectives, and support for obeying *fundamental POLTAX principles* (cf. Sects. 14.3.1, 14.3.2 and 16.3).
   The interaction plans of POLTAX were subject of continuous adjustment to the current context and/or specificity of the state of project. The POLTAX

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_16

principles are especially prepared for monitoring and controlling of such processes like: Human Resources (HR) development, analysis, design, constructions, operations, exploitation, development, security management, and quality assurance of POLTAX.

It should be emphasized that POLTAX is one of the largest IT systems and/or project implemented and deployed in Central and Eastern Europe in the twentieth century. The system was built from scratch. In particular, in 1989, Poland did not have a tax system that could be compared with Western European standards as it did not fulfill the basic criteria for the integration with Western Europe. Moreover, Poland did not have any advanced educational programs for teaching practical knowledge and/or skills necessary for CSE.

Before 1989, depending on the specific needs of a propaganda, it was claimed that in a socialist system, citizens who made a honest living, earning decent money, were not subject to taxation or that special taxes, were being collected from the bourgeoisie that exploited the proletariat, as well as from speculators who preyed on hard-working wage-earners.

As a consequence, the majority of Poles were not aware of the fact that they did pay income taxes and, what is more, they did not know how big the tax burden was (e.g., which tax scale was being used).

Tax offices often functioned like small principalities with a number of its own organizational procedures for manual data processing and data archiving.

Poland lacked an adequate know-how, experience, and awareness (especially in the Ministry of Finance) when it came to the design, construction, and implementation of complex IT systems. There were rarely any computers in tax offices.

The economy was in shambles. New acts, connected with the so-called Balcerowicz Plan,[1] were prepared and passed by a newly-established Polish Parliament in record time (in the second half of 1989). In fact, these acts have made profound changes in the economic system of Poland. They prepared the legal and economic system for the development and implementation of a new tax system in the beginning of the 1990s. At the same time, in 1990, the Polish Parliament was working on the legislation on the basis of which, works concerning the functional and organizational requirements for the newly-introduced system could be initiated. In particular, a new accounting and reporting system for tax offices was developed.

All the circumstances that led to the establishment of POLTAX—were presented above in a nutshell. The description provides the examples of only some (and not the greatest ones) *problems and threats related to the development of an IT system that would for supporting budget replenishment in a reforming country, aspiring to integrate with Western Europe*. At the turn of the decade, i.e., 1989–1990, there appeared many more problems, threats and question marks. Nevertheless, the majority of them are beyond the scope of this book.

It was estimated that approximately 20 millions of taxpayers would file separate tax returns with dedicated tax offices. Therefore, everyone recognized the importance

[1] http://en.wikipedia.org/wiki/Balcerowicz_Plan.

of developing and implementing such an automatic data processing system that could efficiently support the implementation of the new tax system at its further stages in the nearest future (namely, at the beginning of the 1990s).

As a consequence, in 1990, the main objective of the POLTAX project was

> to guarantee an efficient tax and non-tax revenue collection, distribution, planning and monitoring in the reforming tax system of Poland.

The project was meant to be developed in consecutive versions, named POLTAX0.x, POLTAX1.x, POLTAX2.x, etc. These versions were highly flexible in order to respond to hardly foreseeable changes in the Polish tax system that was being developed at the same time. The implementations of the first versions of the POLTAX system at the beginning of the 1990s were closely synchronized with the plans for the implementation of a completely new Polish tax system, that was being developed in parallel.

After the first versions of appropriate POLTAX modules were implemented in Poland, it was possible to implement and introduce PIT and CIT taxes, starting from 1992. Next, after a correct personal and corporation tax assessment for 1992, the value added tax (VAT) was introduced in mid-1993.

The implementation of a tax system based on PIT/CIT/VAT radically improved the State budget (cf. Figs. 16.3 and 16.2) after a huge economic crisis of the 80s (cf. Fig. 16.1), the scope of which is difficult to imagine for the contemporary generation of young Poles. It also paved the way for the Contract of Association between Poland and the European Union related to the eventual access of Poland to the EU and contributed to a relatively systematic economic growth of the country over the next decades (Figs. 16.2 and 16.3).

The functional core of POLTAX consists of the following functional modules which constitute the co-operating subsystems for registration (in particular, Central National Taxpayer Register), assessment, declarations, fines, accounting, monitoring, debt collection, automatic recommendations for tax audit, dedicated data warehouses, reporting, and ontological foundations. By assumption, the system was meant to function on the basis of a separated and secure extensive network of integrated data

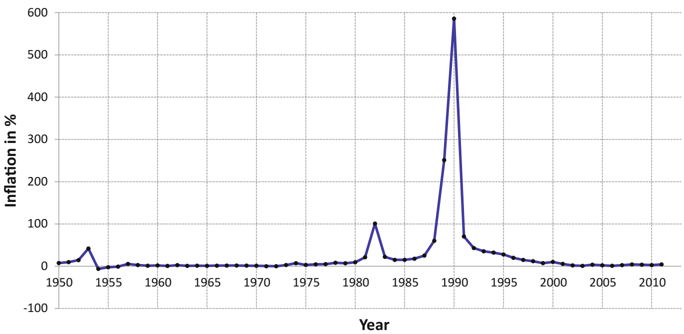


Fig. 16.1 Inflation in Poland between 1950 and 2007 (data from the Central Statistical Office in Poland)

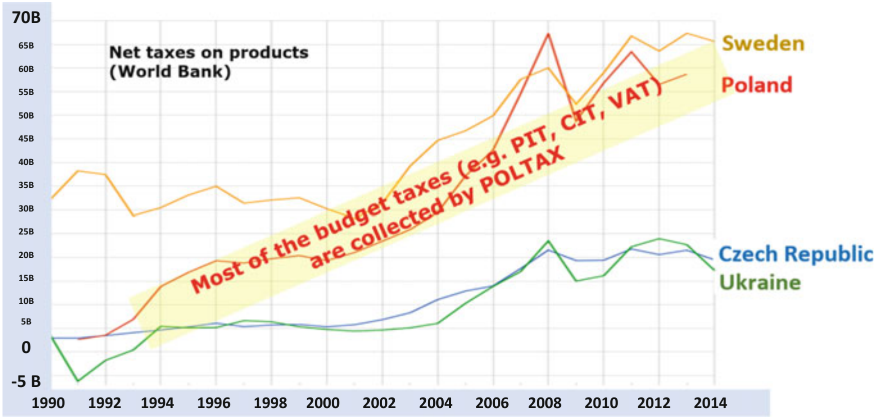


Fig. 16.2 The growth rate of collection of the net taxes on products in Poland (cf. http://data.worldbank.org/indicator/NY.TAX.NIND.CD?end=2015\&locations=PL-UA-SE\&start=1991)

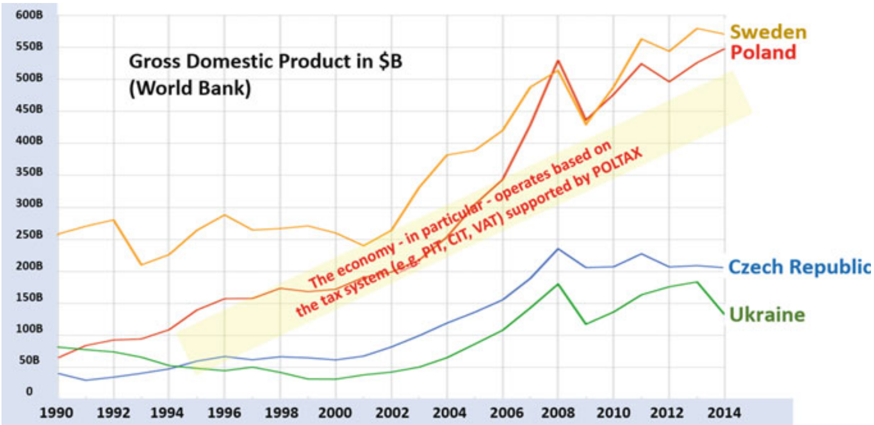


Fig. 16.3 The GDP growth on the basis of the tax system supported by POLTAX (cf. http://data.worldbank.org/indicator/NY.GDP.MKTP.CD?end=2015\&locations=PL-UA-SE\&start=1990\&view=chart)

processing. As planned, two separate centers for data processing were to be created within the network, with the aim of optimizing the efficiency of the system. Therefore, once the POLTAX implementation strategy was accepted in 1991, modern local area networks (LAN) were designed and created in every office, tax chamber and the Ministry of Finance at the turn of 1991/1992. Next, the networks were merged into an integrated network for data exchange, WAN (at first, the connections were made in 1991 via dedicated modems until 1994, when the cooperation with POLPAK[2] was established).

[2]https://pl.wikipedia.org/wiki/POLPAK.

Since the beginning of the 1990s, when the first versions of the above-mentioned POLTAX modules were implemented, the system has been continuously used and developed by the Polish tax services. By assumption, the project was meant to be a long-term initiative.

POLTAX was a conceptual project that was planned in 1990 under the supervision of the author of this book, who developed and to a large extent implemented the fundamentals of the system—its shape, functions, further development—as principles of POLTAX architectural engineering, i.e., the construction, exploitation, and development of the system (cf. Sect. 16.3).

From July 1990 to January 1993, the author of the book was the director of the IT Department under the Polish Ministry of Finance and next during 1995–2000, he held the position of the advisory to the Minister of Finance in terms of IT development within the Ministry of Finance. He was also a member of the IT Advisory Board of the Polish PM.

Some more important conclusions drawn from the beginnings of POLTAX are presented in Sect. 16.4. They were selected from a number of practical experiences, and findings, which are still relevant today. According to the author, in one decade, numerous countries could save billions of (of irrecoverable public fund wastes), significantly improving the quality of their functioning and contributing to their economic growth if adequate political elites (including the members of Parliament and ministers) brought these conclusions to life.

Most of the key technical information presented in this chapter, has been already published in Computer World (e.g., [60, 280]) or, TeleInfo (e.g., [240, 241]), and discussed during special workshops and conferences (e.g., [121]).

The experience gained through commissioning, planning, designing, implementing and developing the project, especially in the 90s of the 20th century, allowed us to collect a considerable experimental material on the basis of which, general principles for a more efficient implementation of such projects could be established. An abstract form of the principles provides the basis for the Framework Postulates for WisTech, presented in Chap. 32.

The experience has also provided a great support in a better comprehension of the IGrC models based on WisTech (cf. Chap. 32). In some sense, FPW presented in Chap. 32 are high-level generalization of the fundamental adaptive CSE principles, presented in Sect. 16.3. More specifically, it means that a thorough analysis of the POLTAX experience led the author to believe that the essence of the principles related to the implementation of such complex projects depends largely on a proper understanding of IGrC related to the implementation of the project. On this basis, it is possible to build and monitor relevant adaptive models for interactive granular computations, process various types of granules, and support them with necessary resources at the right time.

A whole project itself can be treated as a complex process (represented itself by a c-granule) based on interactions between the environment and among c-granules that aim to achieve the trajectories of spatiotemporal c-granules, representing the relevant parts of the project development. Some of the c-granules belong to groups of stakeholders, and project teams, whereas some are simply the c-granules of project agents and/or c-granules of agents involved in interactions with some project components.

Basically, in other case studies described in Part IV of the book, which will be also discussed in this part of the book, similar experience was gained, but with one major difference. Namely, the POLTAX project was associated with the application, adaptation and implementation of a technology that has already existed in the 1990s, and was carried out in a close cooperation with experts who were directly involved in the planning and construction of analogous projects for the legal systems and work culture in France, the United States, the UK, Switzerland, Ireland, and Denmark.

In contrast to POLTAX, other projects (such as AlgoTradix, Merix, Excavio, and UAV) are focused on the creation of a completely new technology (more precisely, a new AI technology), and some of them were even related to the managing the transition from one paradigm of technology construction to another.

16.2 POLTAX: Background, Genesis and Goals

In September 1988, talks regarding the re-legalization of the Solidarity movement began in Poland, thus paving the way for the political and economic transformation in the country. The discussions between the socialist state authorities, representatives of the Polish Solidarity movement, and the Church evolved into official talks known as "The Round Table Talks," which eventually led to a semi-democratic elections in Poland

The elections were held on 4 June 1989 and resulted in the victory of the Solidarity union. The victory enabled the so-called Contract Polish Parliament to appoint Tadeusz Mazowiecki—a representative of the current opposition—as a Polish prime minister. General Wojciech Jaruzelski became the President. With the resumption of the activity, the Solidarity movement achieved some of its goals. These included almost free trade union, the appointment of the prime minister from Solidarity, and one of the objectives of the Solidarity movement from 1980, i.e., the limitation of the political power of PZPR (a communist party, acting as the so called "dictatorship of the proletariat").

It is not possible to describe all the dramatic changes that occurred in Poland, and the USSR in the 80s, and analyze the consequences of these changes for Central and Eastern Europe, as well as other countries, in few sentences.

To better understand the scale of the economical drama, we will use here only one example which shows the level of inflation in Poland after the changes of the 1980s (cf. Fig. 16.1).

It should be noted that after the imposition of martial law in 1981, the situation in Poland was desperate, both in terms of the economy, and general social mood.

A rapidly rising inflation and the lack of stores that could supply basic products, were like metaphorical sparks around an open barrel of gunpowder.

In this context, a partial transfer of the political power from the currently ruling party, i.e., PZPR (a communist party), to the opposition represented by the Solidarity movement, meant assuming the responsibility for the much-anticipated improvement of the economic situation by the Solidarity movement.

At that time in Poland, PZPR (a communist party) was looking for a possible way to weaken the political structures of Solidarity so that they can regain the power. In other words, despite the formal victory of Solidarity in the Polish Parliament and government, it was difficult to bring the actual victory to the managerial and administrational levels of the country. Consequently, hardly anyone could be 100% sure about the success of the future development of the country at the turn of the 1980s/1990s.

There were different forces at various levels of the governmental structure, and the newly elected government had to struggle to take in response to such conflicts.

The context in which these actions took place is of particular importance here as it affected the risk of all the major projects that were undertaken at that time. The Prime Minister, Tadeusz Mazowiecki, was well aware of the ongoing processes. He also had a great sense of responsibility when it came to directing the transformation. The goals were focused on achieving, as soon as possible, the economic stability with a minimal social loss at the same time. Thus, negotiations regarding the reduction of the Polish public debt and the deferment of the remaining debt, inherited from the previous governments, were conducted and an integration process with Western Europe and NATO started.

The economic stability was entrusted to the Deputy Prime Minister, Leszek Balcerowicz, who in 111 days in the second half of 1989, developed a spectacular action plan, known today as the Balcerowicz Plan.

To emphasize the complexity of the process, let us note that the implementation of this plan was initiated by the adoption of 10 law acts on December 28, 1989, which were signed by the President Wojciech Jaruzelski on December 31, 1989. These laws virtually changed the functioning of the economic, banking and financial systems in Poland. An important element of the Balcerowicz plan was a comprehensive modernization of the tax administration, including adaptation of organizational structures to the new demands and conditions (especially, to meet the requirements of the agreements with the European Union Association), as well as the modernization and computerization of the existing tax system, based on manual procedures. The program of activities, including the fulfillment of these objectives and a further development of the tax system in Poland, was introduced by POLTAX.

It is worth noting that, the Balcerowicz plan entailed very great social costs as well, and therefore, it was criticized in the later years. Besides, it was an extremely complex project, given the fact that Poland, with its peaceful nature, had never dealt with a project of such organizational complexity and uncertainty of effects before.

The Balcerowicz plan was fully implemented from the perspective of its formal objectives, such as reducing the galloping hyperinflation, stabilizing the new economic system (based on new taxes such as PIT, CIT and VAT), harmonizing the Polish economy with the European Union (EU) system and negotiating a significant reduction and delay in the repayment of the Polish debt. Figure 16.1 shows a radical reduction and stabilization of inflation obtained as the result of the project implementation.

While characterizing the context of launching a new information technology project for tax services, it should be noted that in the 80s and the early 90s of the 20th century, Poland was subject to very severe restrictions concerning the transfer of technology from the Western countries. These limits are known as the Coordinating Committee for Multilateral Export Controls (CoCom). This was especially true for modern information technology.

For example, between 1988–1989, the author, together with Professor Andrzej Skowron (the Deputy Dean of the Faculty of Mathematics, Informatics and Mechanics at the University of Warsaw), established one of the first "internet" connections in Poland. More precisely, it was a connection to the BITNET network.[3] The connection was based on the electronic modem connection between the Faculty of Mathematics, Informatics and Mechanics at the University of Warsaw and the University of the Balearic Islands. The solution was build using a special computer laboratory based on Pinnacle Computers (Unix/Informix/Motorola environment). The laboratory was founded by a private person, Mr Mirosław Tomalak. One of the most difficult parts of this project was to achieve a necessary administration acceptance (of CoCom and the Polish administration—officially represented by the Rector of the University of Warsaw).

In other words, first of all, it was not easy to educate people in the area of planning and implementing new technologies in Poland, and secondly, Poland was restricted by the embargo on modern information technology transfer. These conditions seriously hindered the possibility of implementing, in an optimal manner, complex IT systems in Poland.

The start-up and implementation of the treasury services modernization program, given the conditions that befell the Ministry of Finance (MF) in the late 80s and the early 90s, required advanced acrobatics. In 1990, the Minister of Finance—Leszek Balcerowicz—entrusted this task to the author of this book who was responsible for the development of the project and its management.

In the summer of 1990, the author of the book established the Department of Information Technology (DI) under the Ministry of Finance.

16.3 POLTAX: Examples of CSE Principles

In this section, we assume that the reader is familiar with the concepts (including principles) presented in Sect. 14.3 (especially with the ABC book principle and its extension in the form of BPCD principles (cf. Sect. 14.3.2.9)).

[3] https://en.wikipedia.org/wiki/BITNET.

16.3.1 Introduction to POLTAX CSE Principles

The POLTAX system engineering was firmly based on appropriate principles of construction, development and exploitation and their adaptive strategies to changes. This made it possible to carry out the implementation of such complex work organization and data processing systems as the consecutive operational versions of POLTAX for almost a quarter of a century. The POLTAX CSE principals were constructed based on the adaptation of BPCD principles (cf. Sect. 14.3.2.9 and Appendix F).

Of course, BPCD principles can be treated in general, as a basis for any CSE project. However, from a practical point of view, it is very important to identify more detailed principles of CSE, based on the specificity of a CSE project. In particular, it is very important to take into account the principles of CSE related to specific characteristic features and expected constraints of a project (cf. the discussion of the "Bermuda Triangle" concept in Sect. 14.4).

Unfortunately, the author had several occasions to look at how complex projects were launched and implemented by project managers and sponsors, who did not take into account the necessity of using any CSE principles. Sadly, all these projects prove that such an approach is not a good example to follow. Typically it is just a "Death Marche" in terms of [545].

The first version of the CSE principles for POLTAX was developed by the author of this book in the first half of 1990. In some sense, it was tested in the very same year, during the implementation of the first version of a pilot project known as DESANT.

On the basis of these experiences, the principles were further developed in the second half of 1990.

Technological details were provided and illustrated on the basis of the second version of the DESANT project. In this way, a much more complex model for the organization of work and data processing procedures was developed as one of the proposals of new organizational systems for tax offices. The systems were based on the first prototypes of the POLTAX software.

The adaptive principles of CSE, developed in 1990, encompassed numerous legal, organizational and technical aspects. Most of all, they ensured a precise synchronization between the construction of POLTAX and the plan for the implementation of the tax system reform in Poland. In the first half of 1991, the POLTAX strategy with a detailed version of the CSE principles was approved. The strategy was approved by a group of representatives that included the potential end-users of the system, namely the administration of the Ministry of Finance (selected in such a way so as to ensure their unrelatedness with those users, who were directly involved in the development of the strategy). The strategy concerning the implementation of POLTAX was also approved within the necessary scope by the Council of Ministers of the Republic of Poland.

A significant part the long-term strategy for the implementation of POLTAX, was formulated in several volumes of technical contract documentation of the BULL company and as a part of in the internal technical documentation, created between 1990 and 1992 by the Department of Information Technology under the Ministry of Finance (compiled in about 150 ring binders of a more detailed documentation). The documents were developed under the supervision of the author of this book, as the director of DI under the Ministry of Finance, through the cooperation of several teams from the Ministry of Finance in Poland,[4] experts from the Ministry of Economy and Finance in France[5] and experts from the BULL company.[6] The works on the development of a strategy for POLTAX were continuously evaluated by an independent QA (Quality Assurance) group, which included experts from the US Internal Revenue System.[7] In this way, experts from IRS also contributed to the final version of the strategy for POLTAX. The principles were developed at numerous levels of specificity.

Within the scope of this book, we focus only on general principles of CSE, related to those aspects of IT engineering, which are of potential concern for each CSE project. We will disregard the aspects related to the legal architecture of the tax system that was being implemented. Some IT-related principles were extremely unique for Poland, such as the principles regarding the standardization of localization to the Polish language: operational systems (UNIX and Windows), ORACLE data base, utility programs (e.g. `Case*Tools`) for software engineers, keyboards, and printers.

We also ignore some significant principles of CSE which were crucial for the synchronization of the tax reform plans in Poland, and the transformation of the work culture in the Ministry of Finance, characteristic of Poland in the 80s, to the newly developed solutions. In particular, we do not discuss the principles of CSE in terms of "work culture legacy management" [281, 376, 516–518].

In this regard, the principles of CSE constitute a part of the *adaptive principles of CSE, supporting the synchronization of communication, design, construction and development of POLTAX within the following 4 basic architecture layers*:

1. *InfraPOLTAX*.
2. *MicroPOTAX*.
3. *MiniPOLTAX*.
4. *MacroPOLTAX*.

A detailed analysis of all these principles (particularly, the explanation of the roles these 4 layers play) could become a topic of a separate book. The principles, along with the techniques for their implementation, were duly recorded by the Ministry of Finance in dedicated ring binders containing the technical and organizational documentation of POLTAX. They were also subject to audits by the Supreme Audit Office (in Polish: Najwyższa Izba Kontroli, NIK) and covered in their reports.

For the purposes of this book, the author made a selection of the most general principles concerning the POLTAX strategy (from the beginning of the 90s). The primary criterion was the *potential usability of principles selected from the CSE*

---

[4] http://www.mf.gov.pl/.

[5] http://www.economie.gouv.fr/les-ministeres.

[6] http://www.bull.com/.

[7] http://www.irs.gov/.

projects that are currently implemented by the organizations, which achieved at least level 2 in a 5-point scale of the Capability Maturity Model Integration (CMMI).[8] The principles were formulated in such a way as to be as resistant as possible to the hardly foreseeable future changes.

The principles, that are to be presented may sound extremely trivial. Undoubtedly, more experienced CSE experts will consider them obvious as these are the principles that should govern every CSE project.

However, in practice, they are not at all trivial. Neither are they easily grasped by those CSE project engineers, who possess little experience. However, even very experienced CSE experts have some difficulty with their implementation. Although many claim that they understand the principles, it is difficult to find a person who would be actually able (in a short period of time) to adapt the necessary details of those principles to the nature of some new CSE project that are being implemented. The situation becomes even more complex in practice, when the principles of CSE are to be continuously adapted to the changing constraints of CSE projects.

The author of this book had a chance to become familiarized with the implementation of complex projects in Poland, USA, France, and Japan. *According to his opinion and estimations, a consistent implementation of the "trivial" principles presented below could lead to a radical increase in the efficiency of public spending in the majority of countries (decreasing the gigantic expenses on unsuccessful projects by at least 50%) and at the same time, contribute to the economic growth of a given country*.

The situation becomes particularly dangerous and threatening (namely, it leads to a massive waste of public funds) when political decision-makers (members of the Parliament, ministers and directors) are not willing to understand the necessity of using these fundamental CSE principles in the implementation of CSE projects.

For example, the author of the book believes that a skillful development and introduction of a legal act, that would enforce only one of these principles, i.e., "HIGHLY QUALIFIED INDEPENDENT VALIDATION AND VERIFICATION," (cf. Sects. 14.3.2.5 and 16.3.2) in a country similar in size to Poland of 2017, would help save billions of dollars that goes to drain due to the unsuccessful implementation of CSE projects, in one decade only.

On the other hand, it is almost impossible to successfully run any CSE project without a proper implementation of the ABC book principle (cf. Sect. 14.3.2.6) and its extension in the form of BPCD principle (cf. Sect. 14.3.2.9). Thus the POLTAX development was based on the appropriate adaptation and evolution the BPCD principles.

[8]http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration.

16.3.2 Examples of Fundamental POLTAX CSE Principles

Taking into consideration the above-mentioned context, and having in mind the ABC book for CSE (cf. Sect. 14.3.2.6) and its extension the BPCD (cf. Sect. 14.3.2.9) are examples of *fundamental POLTAX principles*, we would like to present now some examples of their more detailed forms used for POLTAX:

During the implementation of this complex project (which was highly innovative for Poland of the early 90s) the author, as the POLTAX project manager, focused on the implementation of the ABC book for CSE, its extension by BCDA principle, and additionally the following detailing of CSE principles:

1. **Human/Knowledge Resources Development Principles (HR Development)** encompassing special rules for the Human Resources (HR) development, education, knowledge management, cooperation, communication rules and motivation mechanisms.
   The main objective of this principle is to employ, educate, and motivate the best HR resources required for the development and implementation of a given project. The rules include educational principles in order to provide the necessary expertise for a proper construction, operation and development of the systems created as part of the POLTAX project. The rules have to be consistent with the available project budget and other constraints.
   In each of the 49 tax chambers (existing in the early 90s), a special training room was created, where an advanced training in `Case*Method` and `Case*Tools` of the ORACLE, UNIX, network administration and management, systems engineering, project management, and quality assurance was organized for professionals from various provinces of Poland. Important aspects of this principle were implemented by the POLTAX offset agreements (cf. Sect. 16.3.3).
   In this way, starting almost from scratch, within the firs two years of POLTAX, it was possible to build a strong professional teams in MF and tax administration, which could take over around one thousand IT professionals from the MF headquarters' team and tax offices/chambers. They were well prepared and motivated for a further development and exploitation of POLTAX.
2. **Open Standards** applied to all dimensions of the project (except some obvious security issues). Open standards for technology, especially the open standards for design and construction of IT systems, with a particular emphasis on the UNIX operating environment standards, data exchange network based on TCP/IP, a relational database based on the SQL version of ORACLE, Client Server architecture, etc., were all considered.
   The primary objective of this principle was to focus on avoiding closing of the tax administration in any particular technology and/or licence provider. One of the main measurable indicators of this objective is the cost and/or time to become by MF independent from the particular technology and/or licence.
3. **Continuous External and Internal Stakeholders Communication and Learning** for maintaining ongoing and appropriate interaction contacts, possibly personal, with changing stakeholders of POLTAX and/or their representative agents.

In this regard, it was especially difficult to maintain contacts with the representatives of people who, unfortunately, often changed their roles in the turbulent political times. Therefore, this action also includes the process of identifying, updating and coordinating the activities of the major stakeholders. In the first years of the POLTAX program, special attention was paid to the construction and maintenance of appropriate channels for an ongoing communication, both within the project, which at some stages involved more than 1,000 IT professionals and several thousand tax employees of tax administration.

4. **Concerns and Continuous Identification of Real End-User Requirements** encompassing monitoring the key, and rapidly changing, concerns of stakeholders and framing them in system views. A particular attention should be paid to the ability to predict the future concerns and develop possible actions in response to them. Especially, it is very important to have an understanding of the real end-user's requirements (which are subject to change).
5. **Issues to be Resolved**, including a procedure and rule for the identification, prioritization, processing and communication of relevant issues to be addressed in different system views.
6. **Optimal Solutions**, encompassing conceptual development of optimal solutions to the problems and the verification of the proposed solutions by relevant domain experts. In the case of possible solutions proposed as part of the POLTAX project, the cost/benefit/risk assessment was the main deciding factor.
7. **Integrated Interactive Planning**, including the integration of interactive planning into a single coherent and logical roadmap, encompassing both activities within MF, activities on the intergovernmental level and activities in the Parliament. In 1990, a special software application for integrated interactive planning in various areas of the POLTAX system was implemented.
8. **Flexibility for Future Requirements**, encompassing the development of possible abstract and universal solutions to a potentially predictable changes in project requirements in the distant future.

   To illustrate this principle, let us evoke the history of the development of the basic data model for the POLTAX system. As a matter of fact—after a quarter of the century—it is one of the most important components of the operational part of POLTAX.

   Some experts estimate that in the 21st century, there will be a change in the Polish tax system in Poland every second day.[9]

   In this way, the POLTAX data model has been put to a very difficult test.

   The POLTAX data model was created in 1990–1991, i.e., about a quarter of a century ago. Its development was based on the verification of the accuracy and flexibility of a prototype solution called DESANT ("landing operation" in English). It was the so-called mock-up of the new organization of a tax office, supported by a prototypical computer system.

   The first version of the DESANT system was developed by the Ministry of Finance in Poland in cooperation with a small company, named ProVision. The

[9] http://www.ecddp.pl/download/Relacja_z_konferencji_ECDDP.pdf.

application software for the DESANT system was implemented in the ORACLE environment in the first half of 1990. A little bit more detailed information about the scope of the functionality of the DESANT project can be found in Sect. 16.4, in the part related to the *substantive organizational preparation in MF before the execution of the contract*.

In the second half of 1990, when the model was being developed, a huge effort was put into the construction of the concept ontology (in the form of the entity-relationship diagram), including the expected evolution of the tax system within the next few decades.

The model is based on the experience of the directors who supervised the processes of introducing changes to tax systems in many countries (especially France, the UK, Germany, and the United States). While working on the flexibility of the tax system's ontology, discussions often resembled the debates of logicians and mathematicians, rather than the government officials. They resulted in a spectacular and very complex entity-relationship diagram model, which for a long time was hung on the wall in the room of the DI Director.

This model was implemented and gradually developed in the later stages of the work on the POLTAX system. It has been described as a key element in numerous large volumes of a technical documentation, which contained (in—1991) the vision of POLTAX in the future.

Thanks to its invaluable properties, such as generalization and parameterization, the most important functional elements of the model were easily adapted to a number of legislative changes regulating the functioning of the tax system in Poland in the next two or more decades. In fact, most of them were implemented in the first half of the 90s, including the core part of the POLTAX data model.

9. **Interactive Prototyping and Cyclic Feedback Sessions for the End-Users of the System**—Rapid prototyping and testing of all important project solutions. The most important part of this process is verifying the fulfillment of the end-user's requirements. Well-prepared cyclic feedback sessions (which could be organized using network facilities) are the best tools to do this.

   For example, during the implementation of the POLTAX program, subsequent versions of the software and associated organizational solutions were launched in half-year cycles in order to achieve a better final solution. Initially, these solutions were tested under laboratory conditions (artificial tax offices) and then, implemented in particular tax offices on a larger scale.

   Some examples of the prototypical POLTAX software modules include:

   1. DESANT I (June 1990), as mentioned above in the FLEXIBILITY FOR FUTURE REQUIREMENTS principle.
   2. DESANT II (December 1990).
   3. The prototype of selected POLTAX1.x modules (June 1991).
   4. POLTAX1.x—the module for assigning, cleaning and registering NIP numbers for all taxpayers (December 1991) based on adapted PESEL and REGON databases. This was the first version of the Central National Taxpayer Register.

5. The first version of the POLTAX 1.x (MicroPOLTAX) modules (December 1991/January 1992) and new document processing procedures for tax offices.
6. Deploying of POLTAX1-x (MicroPOLTAX modeles) in all tax offices (Spring 1992) and testing new document processing procedures. Testing and gaining experience for the next versions of the POLTAX system.
7. A prototype of POLTAX 2.x and an illustration of the POLTAX2.x modules' (MiniPOLTAX modules) strategic plan implementation—July 1992.
8. POLTAX 2.x (1992/1994). Development and testing in pilot tax offices. Next, in 1995, the deployment of POTAX 2.x in all tax offices.

10. **PDCA Interactive Evolution**—It refers to the evolutionary nature of the process devoted to the modernization of the tax administration (not only software development for tax administration) within the PDCA cycle. This evolutionary nature manifested itself in the gradual evolution of the system from very simple applications deployed on a large scale in tax offices on personal computers, through more complex ones, achieved by adding UNIX servers in each office, and ensuring their compatibility with those clients, who used PC-LAN. The integration of all servers into a single integrated system, whose activities were to be coordinated by a dedicated WAN network in MF, was the final step. An important aspect of this approach was focusing on the actual user's requirements and implementing them in the form of a rapid prototyping. This was first implemented locally, in designated tax chambers, and then globally.
11. **Backup Solutions for Critical Deliverables**—Any development process of an innovative complex solution could be subject to unpredictable constrains and unexpected POLTAX disasters. Thus, for critical deliverables, it is very important to consider the possibility of producing functional backup solutions (as simple and cheap as possible). It is also a very good idea to have some kind of an internal competition in the project team (between the solution providers). For example, in the case of the POLTAX project, some tax chambers prepared the same POLTAX software modules. The modules were then presented to the end-users (by the Ministry of Finance), who were to judge the winner of this "competition."
12. **Scalability**—It means the ability of any important component of the POLTAX system architecture, to handle a growing amount of work in a viable manner or its ability to be enlarged, in order to accommodate that growth. Scalability is directly linked with legal and organizational changes. It should allow the system to easily adapt to the evolutionary changes in potential organizational models for the implementation of particular stages of the tax reform, according to the schedule.
13. **Security**—Security is an essential aspect for any governmental agency, especially for MF. The data provided by the tax administration is one of the most important security issues for any country.
14. **Highly Qualified Independent Validation and Verification (IV&V)**—Systematic monitoring, evaluation and verification of the progress in planning and implementing the quality assurance IV&V (cf. Sect. 14.3.2.5) mechanisms (cf. comments on the substantive organizational preparation in MF before the

execution of the contract presented in Sect. 16.4). IV&V could be implemented on many levels (i.e., project level, company level, country level) and should be governed by rules and standards similar to the civil engineering law. In other words, in order to build a bridge, a high-way, and/or a home, it is necessary to follow some general rules and standards. Similar situation occurs in the case of CSE projects, especially complex IT systems. The IV&V team of the POLTAX project was built based on very experienced experts, particularly from IRS,[10] EY[11] and Hogan Lovells.[12]

15. **Project Management and Organization Based on Appropriate Adaptation of Case*Method**—see [22].

The above-mentioned principles provide only an overview of all the principles and techniques of their implementation, used from the very beginning of the POLTAX project. They were recorded, in a full version, by the Department of Information Technology under the Ministry of Finance in some of the following documents:

1. About 150 ring binders of a technical and organizational documentation regarding the plans, principles, and techniques for the implementation of POLTAX.
2. Several volumes of a technical contract documentation, presenting the POLTAX strategies from the beginning of the 90s.
3. Records from the weekly reports prepared for the management of POLTAX, containing detailed monitoring of progress in the implementation of particular tasks of the system within the whole country.
4. Information Bulletin issued by DI for tax service employees.
5. Numerous reports from audits made by the Supreme Audit Office (which has been systematically conducted since 1990) as well as reports prepared for the Council of Ministers and the Parliament (including numerous responses to the inquiries made by the Members of the Parliament).

The documentation was duly prepared by the Ministry of Finance and distributed for the intended purposes.

Many aspects of the fundamental CSE principles, used in POLTAX and discussed in this section, were described in the publications from around twenty years ago (e.g., [60, 240, 241, 280]).

16.3.3 POLTAX Offset Agreements and the HR Development Principle

For a proper implementation of the POLTAX project, it was particularly important to hire and consistently develop a highly-qualified IT staff to work for the Ministry

[10] https://www.irs.gov/.

[11] http://www.ey.com/.

[12] http://www.hoganlovells.com/.

of Finance. Unfortunately, in the 80s, there were no higher education institutions in Poland that taught practical skills related to the implementation of complex IT projects, especially the skills based on modern technologies, the transfer of which was restricted by CoCom.[13] Even today, the situation in the domain of the CSE education is not ideal. Luckily, the author of this book was in a convenient situation as he had a possibility to familiarize himself with such teaching programs and acquire these basic skills during his 3-year tenure as an IT professor at the University of North Carolina at Charlotte, Computer Science Department.[14] Being aware of the fact that the Polish educational programs lag behind the needs of the complex IT market, the author undertook some actions, aimed at decreasing this discrepancy. He tried to implement adequate action programs, both at the governmental level (as part of the first government in the 90s) and in the form of the so-called offset agreements, negotiated with IT suppliers for POLTAX. For those who are not familiarized with the concept of an "**offset agreement,**" let us recall its definition, as provided by the Financial Times Lexicon [15]:

> An offset agreement is a type of side deal, sometimes best described as a sweetener. This is an agreement between two or more parties that provides additional benefits and is ancillary to another negotiated contract.
>
> The deal is between a government and company, often a defence company but it can also be used in big civil deals for items such as infrastructure and transport. The company is the one that provides the additional benefit, usually one that is meant to create jobs or wealth for the country's economy.
>
> Companies do this because it boosts their chance to win lucrative government contracts, particularly in developing countries. Countries do this because it makes it easier to justify large expenditures.
>
> A company's offset obligation is usually worth 50–100% of the value of the contract and can be direct or indirect. Direct offsets are linked to the original defence contact. Companies often agree to transfer relevant technological knowhow or use local suppliers to build the equipment they are selling to the government.
>
> Indirect offsets, though prompted by the defence sale, have nothing to do with what the country is purchasing. These can include the company making or drumming up investments in local industries, or helping export a country's goods.

From 1990, the negotiations of such agreements were conducted with all larger IT suppliers as part of the POLTAX project. Below, we mention the results of these negotiations with two big companies, BULL and ORACLE:

1. BULL (cf. Figs. 16.4 and 16.5):
    a. Setting up first academic network computer labs in Poland to teach modern ICT technologies in DPX/UNIX/ORACLE/Case*Tools environments (particularly, the Oracle Case*Method [22] application development life-cycle) at two universities:

[13] https://en.wikipedia.org/wiki/CoCom.

[14] https://cs.uncc.edu/.

[15] http://lexicon.ft.com/Term?term=offset%20agreement.

Bull

1-11, AVENUE DU VAL DE FONTENAY - B.P. 142
94133 FONTENAY-SOUS-BOIS CEDEX
TÉL. : 43 94 60 60 / TÉLEX : 262306 F

M. JANKOWSKI
Director
Finances Ministry

WARSAW (Poland)

October 24nd, 1990

Dear Mr. Jankowski,

This is to confirm our discussions of September 11th, during which we expressed our willingness to enter into a partnership with the Polish Universities.

We tirely believe that this approach is necessary and will enable the promotion of information systems in a very efficient manner within the relevant environments of the Polish National Education System.

In addition this is in line with our strategy of accelerating the training of both engineers and technicians who must soon play a significant role in the implementation of information systems necessary to support the restructuring that the Polish Government has decided upon.

This investment and this cooperation could be considered in several stages :

- Equipment for a "Pilot University" (Warszawa), to qualify the Hardware and Software elements (DPX/UNIX/ORACLE).
- Progressively equip the five Provincial Universities
- The connection of these six Universities together, to exploit the common data base. This would follow the implementation a Telecommunications Network throughout Poland
- Finally, this Polish Universities Network could be connected to the West Europe and US. Databases in order to develop a cooperation with the Scientific Research centers.

.../...

DI-907/6/90

BULL S.A. - SOCIÉTÉ ANONYME AU CAPITAL DE 2.230.522.200 F - R.C.S. PARIS B 642 058 739
SIÈGE SOCIAL : 121, AVENUE DE MALAKOFF - 75764 PARIS CEDEX 16 - TÉL. : 33 (1) 45 02 90 90
ADRESSE TÉLÉGRAPHIQUE BULL SA - TÉLEX : BULL SA 614 050 F - C.C.P. 155 966 JO20.67 PARIS

Fig. 16.4 Bull offset

i. Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw.
ii. Faculty of Electronics and Information Technology, Warsaw University of Technology.

.2

To take in place the two first stages of this project, Bull will provide one DPX 220 equipment to each university and one project leader during this six first months.

In order to go ahead with this large project, I would be obliged if you would nominate a person who has responsibility in the National Education ministry, with whom we can discuss further details.

Please be assured that our group will very strongly support these plans and we are ready to make every effort to help you achieve these objectives in a spirit of full cooperation.

Yours sincerely,

Didier RUFFAT
President and CEO
Bull International S.A.

Fig. 16.5 Bull offset (cont.)

b. Gradual setup of similar computer labs for a network of five higher education institutions outside Warsaw, as needed by and agreed on with the Ministry of Education.
c. Cooperation in establishing, starting, and developing The French-Polish University of New ICT Technologies in Poznań (Poland).

2. ORACLE (cf. Fig. 16.6):

a. 90% discount on all Oracle products for students training and educational purposes.
b. 75% discount on all Oracle products used internally for administrative purposes.

In retrospect, we may say that the successful implementation of the above-mentioned offset programs has significantly contributed to the education of the IT staff in Poland, thus providing specialists, who were potentially available for the implementation of POLTAX.

16.3.4 Political Support for the POLTAX CSE Principles

Of course, the above-mentioned principles seem natural and obvious for the majority of people, who possess experience in the construction and development of complex systems.

ORACLE®

22 October 1990

Mr. Andrzej Jankowski
Director
Ministry of Finance
Department of Informatics
0-916 Warszawa
ul. Swietokrzyska 12

Dear Mr. Jankowski:

Following our meeting of 22 October 1990, I am happy to confirm Oracle Corporation's commitment to Poland and especially to the Ministry of Finance.

University Program

In order to introduce Oracle technology rapidly in your Country, Oracle Corporation is offering their products under special conditions, namely:

o For training and education purposes of the students Oracle will discount their products 90%;

o If products are used internally for administrative purposes, Oracle will discount them at 75%.

The above program will be only offered when the prevention of illegal copying will be guaranteed.

Oracle is ready to set up a special training program for University professors and teachers. Conditions have to be negotiated case by case.

Oracle Presence in Poland

We decided to have our own subsidiary/branch in Poland, and we are committed to do it before June 1991. Naturally, that is dependent on some major issues like financial, taxes and legal.

Phase 1

1 February 1991	Hiring of our first full time employee who will be responsible to set up our office, closing business and hiring future staff (training, technical support, consultants). Setting up relations with polish software houses and hardware vendors.

ORACLE Datenbanksysteme Ges.m.b.H., Schöpfleuthnergasse 25, 1210 Wien, Tel.: 0222/27776-0, Fax: DW 95
Büro Linz: Schloßpark, 4232 Schloß Hagenberg, Tel.: 07236/3331, Fax: DW 30

DVR: 0557161

Fig. 16.6 ORACLE offset

ORACLE®

Mr. Andrzej Jankowski 22 October 1990 Page Two
Ministry of Finance

Phase 2

Based on our results, we may develop more rapidly

Remarks

If the Ministry of Finance could help us by providing offices and clearing some legal and tax issues, we would appreciate this very much and that would allow us to go faster.

Oracle Support for the Ministry of Finance

We will work out together with you and Bull a plan for:

a) Tuning of tax applications written by Bull;

b) Training program for your specialist and special programs for end-users.

c) Participation in the project on a sub-contractor basis.

Sincerely,

Francois Stieger
Vice President

Acknowledged by:

Andrzej Jankowski
Director
Ministry of Finance

FS/jng

ORACLE Datenbanksysteme Ges.m.b.H., Schöpfleuthnergasse 25, 1210 Wien, Tel.: 0222/27776-0, Fax: DW 95
Büro Linz: Schloßpark, 4232 Schloß Hagenberg, Tel.: 07236/3331, Fax: DW 30

DVR: 0557161

Fig. 16.7 ORACLE offset (cont.)

At the same time, every experienced professional is well aware of the following two critical conditions, that determine the success of any complex system development:

1. **Principle of anchoring a project in top management** (cf. Sect. 14.3.2.1). In particular, it means that consistent implementation of the aforesaid principles requires strong support from the decision-makers, that will ensure the coordination and execution of tasks which are often beyond the scope of powers of MF and enter into a complex field of political competition. For example, this happened in the case of the POLTAX project in the Contract Parliament. Luckily for the POLTAX project, these principles were accepted and strongly supported by the Deputy Prime Minister, Leszek Balcerowicz (Fig. 16.7).
2. **Principle of continuous identification, judgment, and adaptation of a family of currently applied principles** (cf. Sect. 14.3.2.2). The fundamental POLTAX principles are relatively general, but as the old saying goes, the devil is in the details. Therefore, in a very vague and rapidly changing environment, these principles may be poorly drafted, overlooking the details, and/or poorly implemented, especially when stakeholders are interested in the failure of the project.

For a project to be successfully implemented, it was especially important to win the support of the decision-makers, as described in point 1. Leszek Balcerowicz was appointed the Minister of Finance twice (from September 12, 1989 to December 23, 1991 and from October, 31 1997 to June 8, 2000). He perfectly understood the necessity of applying the above-mentioned principles of CSE in CSE projects, such as POLTAX. Therefore, the author of this book was in an advantageous situation, having the support from such a person. At the beginning of the 90s, the author of this book was the Director of the Department of Information Technology (DI) at the Ministry of Finance and during the second half of the 90s, he was the adviser to the Minister of Finance and the member of the Information Technology Advisory Board to the Prime Minister. Unfortunately, some successors to the Deputy Prime Minister, Leszek Balcerowicz, in the Ministry of Finance were not eager to support the principles of CSE. What was even worse, they were so busy pursuing their own goals that they did not have enough time to try to understand these principles and become more willing to use them.

The situation occurred in the mid of 1992 and continued to worsen (it especially intensified between 1993 and 1994). It put the POLTAX project in an increasingly difficult and risky situation. To put it mildly, the cooperation between various departments of the Ministry of Finance, engaged in the project, started to decline. There was a risk that the project will go its own way, gradually moving far away from the requirements of its end-users.

Unexpectedly, immediately after the departure of Leszek Balcerowicz from the Ministry of Finance, the position of the author of this book (and consequently, of the very project) in the government administration was reinforced for several consecutive months.

It was caused by the fact that at the beginning of 1992, the author (as the Director of DI at the Ministry of Finance) together with Dariusz Kupiecki (The Director of PESEL—Universal Electronic System for Registration of the Population—at the Ministry of the Interior and Administration) and Kazimierz Krawczyk (the adviser to the Prime Minister) were appointed as the members of the Technical Group to Prime Minister Jan Olszewski. The work of the Group was organized and coordinated by Wojciech Włodarczyk, the minister and the head of the Office of the Council of Ministers. Thanks to the Group, the ideas of the author, concerning his initiative, could have a much wider impact, beyond the Ministry of Finance.

The responsibilities of the Group included formulating proposals of initiatives for the Prime Minister, which were to be presented at the ministerial sessions. The initiatives were aimed at streamlining the Polish government administration and boosting the application of modern techniques of working on the mechanisms of economic development in Poland. The priorities of the Group included improving the flow of information, enhancing the efficiency of decision-making processes by the government administration and supporting the economic planning (with a special emphasis on the use of modern information technology in these processes).

The author, within the scope of the Group, presented several concepts, such as the idea to integrate tax services and customs services in Poland (at that time, Customs Service was subordinate to the Ministry of Economic Co-operation with Foreign Countries) and the idea to appoint the Government Representative for Tax and Customs Reform. According to this concept, political support for the implementation of CSE principles in the POLTAX project was to be reinforced and the project was to be integrated with the functional support from the customs service.

As a member of the Group, the author also developed other initiatives. What is of particular importance, he expanded and reinforced the positive influence of transferring modern information technologies to Poland, as part of the POLTAX project, through a synergy with some traditions of the Polish School of Mathematics.[16] For instance, the author (within the scope of his duties as a government official) once again introduced an initiative of launching a special government program that was to combine the activities of several ministries (education, research, finances and taxes, banking, industry, economy, marketing, promotion activities, etc.), with the aim of creating and implementing mechanisms for the development of small and medium-sized Polish companies, producing software based on the AI technology (of particular concern was its application in medicine, economy, finance, telecommunication, and defense). In simple terms, this was all about positioning the country in a new (at least for Poland) segment of international labor markets.

In spring 1992, Prime Minister Jan Olszewski positively responded to the above-mentioned ideas of the author and began the formalization of the decision-making procedures, with the aim of conducting the feasibility study for these undertakings. However, the procedures were abruptly disrupted due to the departure of the Prime Minister on June 5, 1992, and the unexpected change in the government. As a result, the Government Representative for Tax and Customs Reform has never been

[16] http://en.wikipedia.org/wiki/Polish_School_of_Mathematics,
http://jaszczur.czn.uj.edu.pl/mod/book/view.php?id=1901\&chapterid=11027.

appointed. However, after some time, the Customs Service was transferred to the Ministry of Finance and the process of integrating these two structures was initiated.

It should be emphasized that the synchronization of the cooperation between various organizational units is essential for the implementation of CSE principles (like the ones presented in Sect. 16.3.2). For the implementation of POLTAX, a smooth cooperation with other ministries was needed. In general, such a cooperation is not possible without the support from powerful political decision-makers. Luckily, during the tenures of Tadeusz Mazowiecki and Jan Olszewski, both Prime Ministers and Deputy Prime Ministers were determined to support the implementation of the POLTAX engineering principles. However, with the departure of Oleszewski's government, the situation changed to the worse. The cooperation between DI and other organizational units started to weaken. As a result, near the end of the first half of 1992, the project started to deviate from the fundamental POLTAX principles presented in Sects. 16.3.2 and 14.3.

At the same time, the author noticed that he had troubles explaining the significance (let alone the necessity) of applying CSE principles during the implementation of projects to key political decision-makers. Therefore, in 1992, he formulated a simplified, concise version of these principles, in which he tried to use simplest words possible that were potentially understandable to every "politician." The collection of these rules, is often called by him, as the ABC book principle (cf. 14.3.2.6).

16.3.5 Difficulties with the Implementation of the ABC Book for POLTAX

The ABC book principle, presented in Sect. 14.3.2.6, was rejected by the politicians responsible for the continuation of the tax reform, who were the direct successors of Olszewski's government. For the author, it was particularly surprising given that the majority of these politicians (including the ministers), with whom he had talked about the ABC book before, agreed with him that it should become the cornerstone of the implementation of all complex IT projects (in particular, in the Ministry of Finance). At the same time, **they seem incapable of answering a simple question of why they are unwilling to apply these rules in relation to the computerization of the tax system** that was under modernization.

In order to see through this crucial mechanism, let us analyze the following, hypothetical situation. For example, let us imagine that a Deputy Minister of Finance, responsible for the preparation of laws for a newly introduced "X" tax (or for the modification of the already existing taxes), hypothetically claims that he rejects the ABC book for complex IT projects as it is in conflict with his own tasks and that he refuses to coordinate his tasks alongside other areas of the tax reform. For example, he will not coordinate his implementation plans with the introduction of other taxes. Other activities can only be subordinated to his requirements as he has a good rapport with the Minister. Of course, his stance would be presented in a more "diplomatic way." He would say that *he was instructed by the Minister of Finance to implement solely and exclusively the "X" tax and that all other activities are beyond his competence and area of interest. What is more, he thinks that dealing with other topics could be interpreted as the act ultra vire. It would be in odds with his own principles. Therefore, if the Director of DI wants to discuss some other issues, he should meet with the Minister of Finance himself.* In such a situation, it is difficult to synchronize the implementation plans for all taxes, with the development and implementation of one IT system. Of course, one may say that the Minister is always right, and that the works on accounting and the introduction of other taxes should be suspended, due to a temporary change in priorities.

However, such hypothetical case measures, if introduced, would have increased the chaos and the number of potential errors, as well as generate additional costs along with the frustration of people engaged in the implementation of POLTAX. The author wanted to avoid such situations which could pose a great danger to the tax system that was under reform.

During an opportunate conversation with the Minister of Finance, the author stated that he would continue the management of the POLTAX project, only if the ABC book for the implementation of complex IT projects became re-enforced. The condition was not met by the Minister and therefore, on January 22, 1993, the author left the Ministry of Finance (and resigned from the position of the Director of DI). More details concerning these events, including the context and circumstances, are provided in [240, 241, 280].

The mechanisms presented above touch upon a very important aspect of the "human factor" and its role in the implementation of projects. The very aspect discussed above would be identified by the author as the so-called "window dressing" that led to a change in the CSE principles used for the implementation of CSE projects.

In this case, the crux of "window dressing" is that a project's participant only pretends to apply the project implementation principles. The actual project-related aims are of little significance to her/him.

For example, "window dressing" could be illustrated in the following way: let us imagine that a member of the team assumes responsibility for the tasks she/he is to perform, being aware of the fact that she/he is not going to do it properly. Of course, we do not mean here such trivial reasons for not performing the tasks as being not competent enough. In general, new CSE projects pose such great challenges that numerous participants (if not all of them) do not possess adequate skills at the beginning. Still, it is enough if they are able to make up for these shortcomings as a project progresses. Therefore, to us, "**window dressing**" means a situation in which the **participants pursue goals that are important for them** (and often inconsistent with the aims of the project) and **at the same time, focus on finding the arguments that would justify the failure to fulfill their responsibilities in accordance with the principles that should govern a given project.**

If such a participant feels that the tasks entrusted to her/him are infeasible, she/he should discuss all the doubts with the supervisors. If the doubts are not dispelled during such a conversation, the performer should abdicate the responsibility for her/his assignments. If the supervisors are convinced that someone else is equal to the task, they should shift the responsibility to such a person. Still, the supervisor should first analyze the problems and find the underlying reasons, instead of hiring a person who will agree to become responsible a given task "for a very short period of time."

It should be emphasized that in the majority of cases, *even the most extensive technical, project-related and organizational know-how leads to a disaster, if key people in the project do the so-called "window dressing*."

In the opinion of the author, the human factor, especially the so-called "window dressing" at the level of decision makers, causes that fairly simple and repeatable IT projects (such as computer-aided vote counting during elections on-line car register system, introduction of additional, technically easy to manage properties of ID cards or the possibility of verifying the history of one's social insurance) are still implemented behind schedule in some countries, disregarding a considerable history of generating unnecessary losses and additional social costs in the area of information technology.

Without a doubt, when implementing CSE projects, it is also important to be able to skillfully cope with problems associated with the influence of the "human factor" on a given project (including "window dressing"). In part, mechanisms for managing such problems are related to the mechanisms that help identify, stimulate, develop, satisfy and verify the hierarchy of needs of the project's participants. Some of these aspects might be approached by means of interactive granular computing models based on WisTech, which are currently under development (cf. Chap. 32).

According to the author, the mechanism of introducing changes to CSE principles applied in a given CSE project, is a prevalent practice. Unfortunately, very often the practice is not detected at the right time. As a consequence, the participants of the project become aware of these changes when it is already too late to avoid significant and unnecessary losses in terms of target achievements. A timely detection of changes in the CSE project implementation principles increases the chances of conducting a proper change management process. "Burying one's head in the sand" and pretending that "nothing has happened" may disorganize a project to a greater extent than a suboptimal change management process.

In general, the human factor and associated **mechanisms of changes to CSE principles applied in CSE** projects are underestimated (especially by budding project managers involved in top-level management decision-making processes). The author noticed that the form in which the negative consequences of the "human factor" become more explicit is to a large extent dependent on a country-specific work culture. Therefore, the consequences vary for the projects implemented in different countries such as the USA, Japan, France, Germany, the UK, and Poland. The experience of the author suggests that mechanisms for diagnosing and managing such risks are most successfully implemented in Japan. The topic will be analyzed in more details in Sect. 16.3.6.

Going back to the mechanisms of changes related to the CSE principles for POLTAX, developed in 1990–1991, it should be noted that the Ministry of Finance (MF) put the POLTAX project in a very difficult position after rejecting, in the second half of 1992, the CSE principles; all the more so, because the Minister, after accepting the author's dismissal (motivated by MF decision to reject of CSE principles), did not appoint a new Director of DI in January and February 1993, so there was no one to take the responsibilities for the management POLTAX.

As of July 1993, the position was held by Professor Antoni Kreczmar from the Institute of Informatics at the University of Warsaw.

Director Kreczmar put a lot of effort in trying to implement the POLTAX project despite the lack of compliance with the ABC book principle for complex IT projects (and in consequence despite the lack of compliance with the BPCD principle). To be more exact, the implementation of POLTAX was carried out on the basis of organizational and technical inertia, resulting from the implementation of the strategy that had been already developed. At the same time, the implementation of the project was based upon the CSE principles that were more and more distorted.

Consequently, despite huge efforts of Director Kreczmar, the cooperation between IT specialists and competent departments left a lot to be desired.

Of course, in such a situation, IT specialists become the so-called "scapegoats" and are out of work without any fault of their own.

Unfortunately, that was the case of Director Kreczmar who was dismissed at the end of October 1994. In a Polish edition of Computerworld from November 14, 1994, Kreczmar comments on the decision about his dismissal in the following manner[17]:

> Unfortunately, the decision about my dismissal from the position of the Director of DI – says Kreczmar in the October edition of the Public Information Bulletin issued by the Department of Information Technology under the Ministry of Finance – was made upon the long term analysis of my current achievements and in response to the new challenges that I was not able to handle with them. [...]
>
> I fell like a sportsman who did not lose the distance to the lead on an important leg of a relay race and handed off the baton to the next runner. I might have made some mistakes, but my leg was a very long curve and I was striving to turn the corner. [...] My successor should continue running to the finish as quickly as possible.

After the dismissal of Kreczmar, the responsibilities of the Director of DI at the Ministry of Finance were taken over by Michał Zalewski. Luckily, Director Zalewski managed to convince the decision-makers to apply at least some of the principles presented above. The new director and the Minister of Finance also asked the author of this book to once again support the works devoted to the implementation of POLTAX (as an adviser). As a result, in 1995, the Minister nominated the author of the book for the IT adviser to the Minister of Finance. The Deputy Prime Minister and the Minister of Finance, Grzegorz Kołodko, decided to return to some of the CSE principles developed for POLTAX. Unfortunately, many principles that have been already mentioned in the book were not accepted by him (in particular, the IV&V principle which has not been implemented in MF till date). However, the main elements of the POLTAX strategy, defined at the beginning of the 90s, have

[17] http://www.computerworld.pl/news/302891/Zmiany.w.Departamencie.Informatyki.Ministerstwa.Finansow.html.

been to a large extent implemented in MF. Today, the POLTAX system effectively supports the operations of the tax administration using the modules, which were designed and deployed in the 90s of the 20th century.

16.3.6 POLTAX CSE Principles and Japanese Approach to Complex IT Projects for the Ministry of Finance in Japan

Between 1994–1999, the author was the Vice-Rector for Research and Didactics at the Polish-Japanese Institute of Information Technology.[18] At that time, he was able to learn and participate in numerous projects conducted in accordance with the work culture characteristic of Japan.

In 1995, the author successfully completed the Individual Training in Computer Science Education, organized by the Japan International Cooperation Agency (JICA). As part of this cooperation, the author visited the Ministry of Finance in Japan in 1995. During this visit, he was able to share his experience with the corresponding officials who managed the construction and development of projects similar to POLTAX. That gave him the opportunity to become acquainted with the ADAMS project (**Accounting Affairs Data Communication Management System**), including its history, current status and implementation plans.

The main task of the ADAMS system was to maximize the efficiency of the governmental administration in terms of budgeting, budget management and budget settlement, performed by the governmental institutions in Japan. The key part of the data model for the ADAMS system has been a detailed representation of activities related to the accounting and budgeting system of the Japanese government. A general data flow diagram for the current version of the ADAMS system is presented in Fig. 16.8.

For the author, the visit to the Ministry of Finance in Japan was a genuine learning experience and it broadened his knowledge as to the implementation of complex IT projects for the governmental administration in countries with different work cultures. In particular, the visit enabled the author to realize that there is a great difference between the principles listed, proposed, and used for the purpose of POLTAX (cf. Sect. 16.3.2) and the principles applied in similar projects in Japan.

It is impossible to summarize all these differences in this book, but we will try to present at least the most important (according to the author) ones, which are related to the techniques for managing the "human factor" discussed above. When it comes to various projects, Japanese people attach great importance to the concept of long-range planning along with the development of human resources that could support the implementation of a given project.

To put it simply, according to the Japanese strategy, a project similar to POLTAX should be preceded by numerous preparations to ensure a proper verification of can-

[18] http://www.pja.edu.pl/.

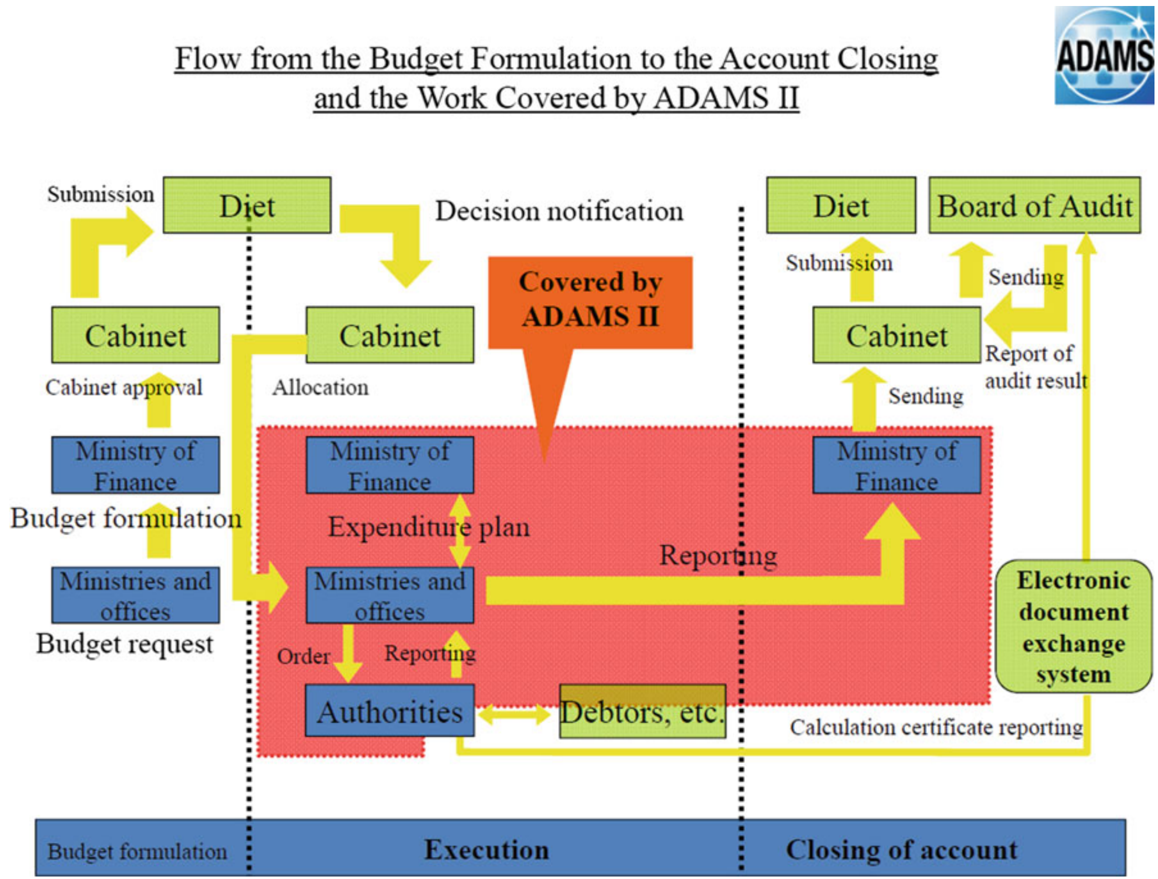


Fig. 16.8 ADAMS II—General data flow diagram (*source* Ministry of Finance in Japan–April, 2015)

didates (especially representing the end-users) as project's leaders/key participants. This particularly applies to providing special education to people who will represent the interests of the end-users of a given IT system. The candidates should be selected from practitioners, who possess considerable experience and know-how in particular areas of applications.

One of the techniques of the early-stage sourcing for the purpose of further verification and employment in the project was providing potential employees with computers for education and training (if possible, in an entertaining way, for instance in a form of a "game").

Such "educated" practitioners were then gradually included in processes that were being launched, in which they had to formulate and answer more and more detailed questions concerning the new organizational and legal solution concepts.

It was equally important to come up with a visualization of a new system solution, in a form of a mockup, and to perform the analysis connected with increasing the efficiency of the new organizational solutions. These activities also prepared the ground for proposing a given course of action to legislators (including the Parliament) and synchronizing the progress in the introduction of legal amendments with works devoted to the software and administrational organization.

The development and implementation of consecutive versions of the ADAMS system lasted (by very rough estimates) about 8 years. Figure 16.9 presents a brief history of the ADAMS system as of April 2015.

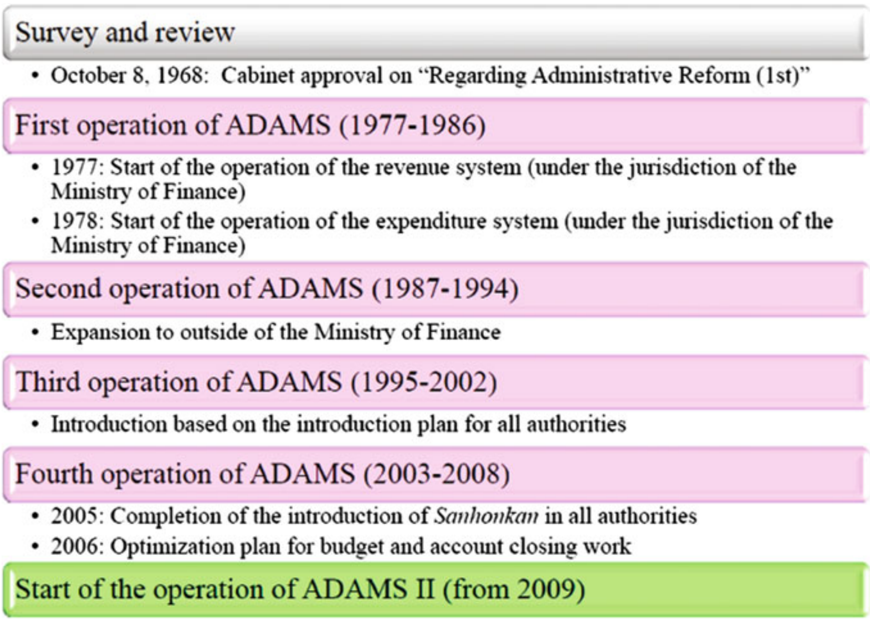


Fig. 16.9 Brief History of the ADAMS System (*source* Ministry of Finance in Japan–April, 2015)

As can be seen in Fig. 16.9, works on the first release of the ADAMS system were approved in 1968 (as part of the governmental administration reform). The first version was implemented in 1977.

In 1995, when the author met with the employees of the Ministry of Finance in Japan, they had been already working on a version of the ADAMS system that was to be implemented around 2003. At that time, the author was surprised that the Ministry was planning to develop an IT system that was to be implemented in 8 years (which at that time seemed a very long time horizon for the author). Therefore, the author asked his Japanese friends, half-joking, half-seriously, how could they predict an upcoming 8-year perspective in terms of the capacity of computers, the capacity of data flow, and the capacity and speed of the hard disk?

His friends from the Ministry of Finance in Japan answered, also half-joking, half seriously, that according to their state of knowledge, within the given time horizon, it was still possible to make extrapolations regarding the development of hardware and its technical parameters on the basis of the Moore's law (constrained by the quantum mechanics).[19] Twenty years later, one has to admit that they were right.

On April 30, 2015, the author, after 20 years, sent an email to the Ministry of Finance in Japan with a description of his visit in 1995, asking for the information about the current state of the ADAMS system. The answer came about 2 days later. It included Figs. 16.8 and 16.9.

On the basis of the reply, the author learnt that the fourth version of the ADAMS system, about which he talked in Japan in 1995, was implemented 8 years later in 2003. It was used until another version, ADAMS II, was implemented. According to Fig. 16.9, ADAMS II was launched in 2009.

The above reflections should not be treated as an in-depth comparative analysis of the methods used for the implementation of complex IT projects in Europe and Japan. The intention of the author was to emphasize that there are certain differences between these cultures, especially when it comes to managing problems associated with the so-called "human factor," time horizon, and the accuracy of project planning.

At the end, the author would like to share one more important (in his opinion) observation regarding the differences between the organization of projects such as POLTAX and ADAMS.

While creating the organizational structures of teams, responsible for planning, construction, exploitation and development of POLTAX, the author (due to economic reasons as well as his intention to reduce the number of potential sources of internal conflicts within a team) tried to minimize competency overlapping (in terms of delegating, making decisions and performing tasks) as well as responsibility overlapping between various people in a team.

In general, this meant that overlapped competencies and responsibilities between various participants of the project were limited to the requirements related to the attempts to ensure work continuity (e.g., as one of the aspects of knowledge management) and the highest quality possible.

---

[19] http://en.wikipedia.org/wiki/Moore's_law.

Similar feeling concerning the minimization of competency and responsibility overlapping were observed by the author in the majority of projects implemented in European countries. He noticed that in a number of these projects (including POLTAX), a great emphasis was put on determining the roles and responsibilities of employees in the most accurate way possible. To use a metaphor, **we can imagine that each participant of a project (and her/his co-workers and/or students) is in a separate box with well-defined edges (at particular levels of organizational and managerial structures)**. In this model, the majority of European countries prefer teamwork to be handled at a lower organizational level. In practice, higher organizational levels are those where the hierarchical relationships are most predominant. According to the author, this metaphorical organizational model is very popular in Europe, but differs from the model that is used in Japan.

The predominant organizational model of projects implemented in Japan is oriented towards teamwork (including two-person teams) not only at the lower levels of organizational structures, but also at the top. What is more, in Japan, employees are not contained within boxes with such well-defined edges as in Europe. The boxes that define the scope of responsibilities and competencies very often overlap. Thus, the decision-making process is much more complex as it involves a greater number of co-workers, who are all entitled to take a stance regarding a given decision. In case it is impossible to reach consensus within a pre-determined period of time, the problem is also escalated to the higher managerial level.

Once more, let us emphasize that the above remarks are based on the personal observations of the author, mainly in terms of complex IT project engineering. More information regarding the implementation of CSE projects in Japan may be found in [219, 255, 266, 476].

16.3.7 Today and the Future of CSE Principles in POLTAX-Type Projects

Certainly, today—after a quarter of a century—we have a much broader and deeper knowledge about how to plan and develop complex systems.

Especially essential part of the engineering of CSE systems are principles of architecture engineering, This particular knowledge even went through internationally recognized standards such as ISO/IEC 42010.[20]

Various recommendations and criteria for assessing the quality of the rules which should be observed when planning and building an architecture, were created. The examples of such general criteria from the TOGAF[21] architecture framework include:

---

[20] https://en.wikipedia.org/wiki/ISO/IEC_42010.

[21] http://www.opengroup.org/subjectareas/enterprise/togaf.

Understandability. The underlying tenets can be quickly grasped and understood by individuals within the whole organization. The principle is meant to be clear and unambiguous, so the risk of violations, whether intentional or not, is minimized.

Robustness. It enables one to make-good quality decisions regarding architectures and plans that are to be devised, and enforceable policies, as well as standards which are to be developed. Each principle should be sufficiently definite and precise to support consistent decision-making in complex, potentially controversial situations.

Completeness. Potentially, every important principle governing the management of information and technology in a given organization is defined. The principles encompass every perceived situation.

Consistency. Strict adherence to one principle may require a loose interpretation of another principle. A set of principles must be expressed in a way that ensure a "balance" among their respective interpretations. Principles should not be contradictory in a sense that adhering to one principle would violate another. The language of a principle should be carefully formed so that a consistent, yet flexible interpretation can be allowed.

Stability. Permanent principles should be stable, yet capable of accommodating to changes. An amendment process should take place when adding, removing, or altering principles after they are initially ratified.

Without a doubt, it is recommended to update the principles that govern the development of a complex system once in a while.

It should be emphasized that TOGAF [373] is only one of many approaches to determining CSE principles. It is worth mentioning other approaches as well.[22] However, many practitioners believe that these approaches are not flexible enough and thus, impose too great constraints on the implementation of innovative CSE projects. However, when it comes to domain projects (e.g., medical, industrial, construction), they tend to ignore crucial aspects of a rapidly developing domain knowledge. In this context, there is a need for establishing a new approach among the practitioners. According to the author, such approaches (at a high level of abstraction) might be developed on the basis of interactive granular computing. The topic is discussed in greater detail in Chaps. 32–35.

16.4 POLTAX: Brief History of the First Decade

Project launching. In June 1989 (right after the "Round Table Negotiations" and before the establishment of a new government by Prime Minister Tadeusz Mazowiecki, the MF management decided to issue a tender for the purchase of a computer system supporting new tax services: PIT, CIT and VAT. The tender for a computer system in the Ministry of Finance was under supervision of Director Józef

[22] http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html.

Piskorz. The tender was announced in August 1989 (i.e., during the establishment of the Contract Parliament and the beginning of the Tadeusz Mazowiecki's office as the first noncommunist Prime Minister in Poland after World War II).

It is remarkable that in 1989, MF was virtually without computers. In addition, the organizational structure of about 320 tax offices, subordinated to 49 tax chambers, was based on the disordered and fragmented operation procedures. Furthermore, their workflow was poorly documented and it varied from one tax office to another. At the same time, the new tax system was implemented with substantially different objectives and conceptual operational framework from the system that was developed over fifty years ago in the so-called centrally planned socialist economy. To this, one should add the lack of universities in Poland, which in the 80s could train professionals that were needed for the design, planning, construction, and development of large, complex IT systems.

Substantive organizational preparation in MF before the execution of the contract. Before selecting the final contractor for supporting the implementation of the projects and signing the contract on July 1990, a preliminary specification framework concerning legal, organizational, functional and technical aspects of the new tax system was developed. On this basis, by the end of June 1990, MF, in cooperation with the staff from ProVision, a Polish IT company, built a simple model of a system dedicated for a typical tax office. The project that led to the creation of the mock-up was launched in the second quarter of 1990 under the name "DESANT." The mock-up included legislative, organizational, functional and technical specification frameworks.

DESANT was implemented in a Unix/Oracle environment and the mock-up model was created for the organizationally modernized tax offices. This included some graphic components for user interfaces, new organizational structure procedures, some new accounting rules and new proposals of key legacy management factors [281, 376, 516–518].

The mock-up along with the documentation were demonstrated to the prospective users, including the heads of tax offices, tax chambers directors, and members of the MF management. It is worth noting that in June 1990, optical scanners produced in Poland were successfully used in the DESANT software for data loading and taxpayer identification, whereas the electronic transfer of documents to the system was envisioned for the future. This mock-up was a perfect tool to analyze the requirements of the end-users, which were critical in terms of finding an adequate partner that could support MF in the implementation of the project.

On the basis of the analysis, in the fall of 1990, the first version of a prototype software of POLTAX was prepared in cooperation with the Bull company. In December 1990, this prototype application software system was demonstrated to all directors of tax chambers in Poland. The aim of MF was to see through the actual and practical requirements of tax offices. This knowledge was the basis for the analytical studies conducted from 1990 to 1991. Several volumes of professional, technical documentation, describing the optimal strategy and architecture for the implementation of POLTAX, was compiled. An executive summary of the POLTAX strategy was presented to the Prime Minister and the Council of Ministers in July 1991.

Along with analyzing of the conceptual model of the organizational and technical system, a comprehensive, detailed plan for a strategic tax reform was developed in 1990, including legal, organizational, administrative, and technological aspects. The aim of the plan was to coordinate all major courses of action. It was approved by the MF executives in July 1990. As part of this plan, legislative procedures for the introduction of new taxes were initiated and in a short time, a small group of people turned into a very powerful team, consisting of more than one hundred of professionals with an extensive IT experience from Poland and abroad.

At that time in Poland, no one had any experience when it came to legal aspects of contracts commissioning the creation of a complex IT system. Therefore, numerous contacts with experienced foreign experts were established in order to support the negotiations and secure the interests of MF in the final contract.

In this process, the most important role was played by Bob Kenney,[23] employed by MF, who is an eminent and experienced specialist in contracts for complex systems. During that time, he was a partner of one of the most prestigious law firms in the world: Hogan and Lovelles.[24]

The contract with Bull had a very complex legal and financial structure. The original contract covered all phases of system operation: from strategic planning to implementation, which was planned after the final acceptance of the system. All later analyses of the contract showed that it had been prepared in a professional manner and proved very beneficial for MF. The experience with the legal aspects of POLTAX allowed Bob Kenney to grasp the nature of our legal requirements. This helped him to prepare the first version of the Law on Public Procurement which, after some amendments, was approved by the Parliament.

In retrospect, one must say that unfortunately, many legal solutions introduced as a part of the contract with Bull and proposed for the implementation in Poland have been abandoned today. The abandonment of IV&V principle, implemented under the contract with Bull, proved to be the most detrimental.

Roughly speaking, the essence of the IV&V mechanism (at the project's local level) is based on the fact that in the case of large IT projects that are publicly funded, the customer is obliged to set aside a few percent of the total contract value (from 2 to 5%), depending on the characteristics and risks of a given project, which would be deducted from the contractor. In accordance with the pre-established procedures, these resources should be used for monitoring the correctness of the project implementation, as well as the level of emerging risks and proper functioning of the quality assurance mechanisms. Besides, in case a project fails due to the negligence of the contractor, the company will suffer the consequences.

The IV&V principle can be implemented in many different ways. Of course, it directly involves the benefit-cost ratio. It makes no sense to paralyze a given project by a very complicated quality process, which does not have a significant impact on the final outcome. On the other hand, in practice, a complete elimination of these mechanisms in independent project monitoring translates into very great losses.

---

[23] https://www.linkedin.com/in/robert-kenney-01683633/.

[24] http://www.hoganlovells.com/.

For example, in the 90s, the representatives of MF made numerous attempts at implementing IV&V principle in all important projects conducted in Poland. Unfortunately, these attempts were unsuccessful and in a consequence, led to enormous financial losses and delays in the implementation of major projects in many departments in Poland. A very interesting form of implementing IV&V principle can be observed at NASA.[25]

In the case of the POLTAX project, during the first years of its implementation, the IV&V team consisted of experienced designers and professionals from IRS (Internal Revenue Service in the U.S.), supported by the Swiss Government which, as a result of the internal tender, sent a group of consultants from Ernst and Young to MF.

Unfortunately, in 1993, one of the Ministers of Finance rejected many CSE principles that were fundamental to the development of POLTAX. In particular, he decided to resign from the IV&V principle.

Luckily, in 1994, another Minister of Finance recognized the importance of CSE principles (in particular, the BPCD principles). Unfortunately, many of them have not been introduced till date. For example, according to the knowledge of the author, the IV&V principle as described above has not been implemented so far. At the same time, in the recent decades in Poland, the majority of finance ministers (at the end of their term) recognized the poor quality of IT complex projects, implemented in the Ministry of Finance.

Choosing the Bull company. On the basis of the tender evaluation, in the first half of 1990, two companies, Bull and IBM, were qualified for the final phase of the tendering procedure. Others failed to sufficiently fulfill the following three criteria of MF: the expected range and quality of software services, high-quality technical equipment and favorable financial conditions.

Before making any final decisions, experts working on behalf of MF analyzed in details the technical proposal of the Bull company, which was based on the UNIX/ORACLE environment (including assessing the quality of Bull's hardware). Technical expertise of their solutions proved to be satisfactory.

It is worth mentioning that Zenith personal computers delivered to Poland by Bull were for many years one of the best-selling computers for the U.S. administration. IBM was requested to prepare the offer based on the UNIX operating system.

Rather than providing a new offer based on UNIX, IBM experts and top managers consistently repeated their arguments, justifying the superiority of solutions based on the IBM OS/400 operating system over other solutions (including UNIX). To MF, solutions using the AIX operating system developed by IBM were particularly detrimental. The IBM experts mentioned the following disadvantages of the UNIX operating environment in comparison with OS/400: *UNIX has much lower performance, lack of adequate system security and reliability, weaker tools for software development and application of project. It requires specialized expertise in maintenance, lacks of ools for application development, and generates high costs of software maintenance*.

---

[25] http://www.nasa.gov/centers/ivv/home/index.html,
https://www.nasa.gov/centers/ivv/about/index.html.

It should also be noted that the open UNIX systems were recommended for the computerization of the government in Western Europe and the USA. The main advantage of the solution proposed by Bull, in comparison to that of IBM, was that the openness of systems could be guaranteed so that in the future, MF would not be dependent on one supplier of information technology. On this basis, MF made its final decision and chose the bid submitted by Bull. Poland was supported in the modernization of the tax reform by France, which was the first country to introduce VAT on the basis of contractual mechanisms. It is worth noting that sometime later, in an analogous tender for the computerization of the tax system in the Czech Republic, IBM won by offering solutions based on UNIX. In the summer of 1995, in Augustów (Poland), at the international symposium organized by the Institute of Computer Science of the Polish Academy of Sciences, Warsaw University of Technology, and Polish-Japanese Institute of Information Technology, IBM experts presented to the POLTAX managers the history of computerization of fiscal offices in the Czech Republic, where the first piece of software was implemented after two years, using open standards. The presentation showed that this success was possible owing to an effective process management, comprehensive modernization of the tax system in the Czech Republic, and the involvement of native experts in information technology.

Thanks to the fact that the POLTAX system architecture was based on open system standards, MF became technologically independent from external information technology companies when the contract with Bull ended.

The tender for a purchase of a computer system for MF, which led to the choice of the France-based company, and not the US-based company, caused consternation among some political circles in Poland as well as overseas. The case was investigated by a number of committees and regulatory bodies (for example by the Special Committee of the Polish Parliament). On these hearings, MF was represented by the Deputy Minister Stefan Kawalec, the author of the book (as the director of IT department), and the adviser to the minister—Jacek Karpiński (the constructor of one of the world's first minicomputers—K202 (cf. https://en.wikipedia.org/wiki/K-202)). In the international context, it was fortunate that the U.S. Treasury came with a technical mission to MF during the final stages of the tender. This was at the time when MF was making the decision concerning the computerization of the tax administration.

Since the mission included experienced specialists in the field of computerization of the tax administration in the U.S., they were also invited to support MF during talks with different companies and to share their opinion. The final report from the mission was transferred to MF by the management of the U.S. Department of the Treasury in autumn 1990. The following words of the experts from the U.S. tax services, taken from the official report, confirm the validity of the decision made by MF:

> Bull had been picked over IBM, however, because they presented an "OPEN" solution, while IBM's was a proprietary one. **We agreed this was a wise course**. IBM had been apprised of this desire, but didn't amend their approach.

This opinion was signed by John Robson (the Deputy Secretary of the Treasury, United States[26]).

The contract. After the approval of the POLTAX implementation strategy by the government in July 1991, the value of the contract was calculated at 442.9 million FF. The amount was beneficial to MF and it hardly increased till the end of the contract term.

The final value of the contract can be divided into two types of expenditures. The first one included hardware components, software system, utilities, and structured cabling network for all tax offices, and tax chambers as well as the MF building (373.1 million FF). The costs associated with labor, training, documentation and Polonization constituted the second part of expenditures (69.8 million FF). It should also be emphasized that the value of the contract was negotiated and it was agreed that in order to receive the payment, the contractor must deliver a fully integrated system, capable of satisfying the requirements of the end-users in accordance with the pre-determined strategy. The intention was to avoid splitting the contract payment into several independently treated payments for each piece of hardware, software, and other services.

This was reflected by relevant mechanisms of paying for the packages of received goods and services. For example, 5% of the contract value (about 22 million FF), covering both hardware and software as well as services, have been earmarked for the proper execution of the strategy development phase (which took about a year), while another 5% (about 22 million FF)—as a payment for the Final Acceptance of the system.

In addition to these general indirect mechanisms, which enforce the construction of an integrated system, mechanisms of paying directly for the provision of basic packages of services and products at different stages of the software development were implemented in accordance with the `Case*Method` described in [22]. The development standard of the `Case*Method` software was supported by the ORACLE's Computer-Aided Software Engineering Tools (`Case*Tools`). At the beginning of the 90s, a special training in the `Case*Tools` environment was organized for all 49 tax chambers.

The Bull contract guaranteed a greater flow of the modern information technology to Poland. For example, pursuant to the contract, around one thousand of professionals in Poland were trained in the development of information system and equipped with modern CASE Tools. Moreover, structured wiring was implemented in all tax offices and computer equipment was provided, along with a technical documentation and modern IT environment and network, polonized by Bull and based on UNIX/WINDOWS/TCP/IP/ORACLE.

In order to better visualize the scale of the supply contract, one can imagine about two hundred large TIR lorries importing hardware, cabling and software for all tax offices and the installation of approximately 18,000 outlets for connecting

[26] http://en.wikipedia.org/wiki/United_States_Deputy_Secretary_of_the_Treasury.

those computers to a network of the electronic information exchange. In addition, there was the installation of approximately 1,200 km of a data secure transferring cable and another 1,200 km of a power cable during the business hours of a tax office. In this way, the supply contract led to the establishment of an information infrastructure which gave the opportunity to exchange information between all tax offices, tax chambers and MF—with modems installed in each minicomputer. Based on this platform, a private Wide Area Network (WAN) for the Ministry of Finance and tax administration was launched.

With the help of the technology transfer, guaranteed under the contract with Bull, it became possible to develop some software modules of the POLTAX system by regional IT engineers, working in tax chambers and tax offices. For this purpose, they used very modern methods and tools.

In 1992/1993, the next iteration of software for registering taxpayers and recording as well as processing their declarations in relation to PIT/CIT/VAT was successfully implemented within the framework of MiniPOLTAX.

End of contract with Bull.

According to the original contract, in 1995, Bull was to receive the certification of the Final Acceptance of the system.

However, there occurred a fundamental problem.

The system supplied by the Bull formally met the official software requirements that were once defined by the Ministry of Finance (in particular, for creating a module of a new accounting system for tax offices), but, unfortunately, the “official requirements” provided by DI to BULL in 1993 did not comply with the end-user requirements. This situation was a direct result of the cancelation (by the Ministry of Finance in the spring of 1993) of numerous CSE principles that were to govern the implementation of POLTAX (mentioned in Sect. 16.3).

Thus, the software provided by Bull did not meet the actual requirements of the end-users that worked in tax offices (especially when it comes to a new accounting system). However, it satisfied the “official software requirements” of the Ministry of Finance. Nevertheless, the system was not fully adjusted to be deployed in all tax offices, tax chambers and the Ministry of Finance itself.

As stated in the contract, the final acceptance of the system could only be made on the basis of the results of the tests. Because Bull prepared a software that was in line with the requirements that were officially approved in 1993 by MF, the company demanded a final acceptance of the system despite the fact that the entire system could not be implemented.

Negotiations between Daniel Le Coguic (from BULL) and the author of the book (as the adviser to the Minister of Finance) were undertaken and as a result a compromise was signed on August 1, 1995, satisfying both parties that terminate the contract. As a result, MF was given the rights to use and modify the contract and its related products (including the software). In addition, the Ministry received a group of selected professionals from Bull who were at its own disposal and continued working on the development and stabilization of the software till the end of 1995. Without increasing the value of the contract agreed in 1991, Bull upgraded the computer equipment worth 10 million FF at Polish tax offices. Thus, the contract was amicably terminated for both parties.

A relevant and complete version of the accounting software system (that met the requirements of both MF and the end-users) was developed by MF between 1992 and 1994 outside the contract with Bull, with the support from the local tax offices (POLTAX2B by the Kraków-based tax chamber and POLTAX2A by the Bydgoszcz-based tax chamber). In 1995, after the analysis of pilot operations in selected tax offices, MF decided to deploy POLTAX2B in all tax offices. The system has been in use till date (with appropriate modifications following some legal amendments).

Local Area Network (LAN) and Wide Area Network WAN). From the very beginning, the long-range strategy of POLTAX was aimed at constructing an integrated data processing system based on wide area network. This part of the strategy was referred to as MacroPOLTAX. However, in 1990, there were virtually no computers in the Ministry of Finance and its subordinate offices. To all intents and purposes, the majority of tax offices lacked a good quality power system. Therefore, while devising the LAN structured cabling plan for approx. 320 tax offices, 49 tax chambers, and the central office of the Ministry of Finance, it was necessary to design an off-grid power system as well.

In the first half of the 90s, as a part of the contract with BULL, modems on all UNIX-based servers (in tax offices and tax chambers) were configured. The modems were to be used for transmitting information to the processors of these servers. The connection was then established via a telephone network. The data transfer rate was extremely low. In the second half of the 90s, a much more advanced WAN architecture was developed. It was based on three layers: core layer, distribution layer and access layer. At the beginning of the first decade of the 21st century, the network was developed in cooperation with TP S.A. by means of the "Polpak-T" packet network, using Frame Relay technology. In 2002, the Central Customs Office was connected to the network.

In August 2006, Wide Area Network supported the implementation of the first phase of e-POLTAX, a subsystem thanks to which it was possible to send tax declarations to POLTAX via the Internet.

In 2007, Wide Area Network was reorganized in MPLS technology, which fulfilled the requirements of POLTAX in a more efficient way.

16.5 POLTAX: Main Results

Few people are aware of what really happened during the implementation of POLTAX at the beginning of the 90s, how great obstacles had to be overcome, and how big consequences the introduction of POLTAX has had for the future of the country.

Without a doubt, the biggest positive consequence was that the implementation of consecutive versions of POLTAX at the beginning of the 90s resulted in the introduction of a new tax system in Poland, which contributed to the stabilization and significant increase in the tax revenue, paved the way for the economic growth of the country (cf. Figs. 16.2 and 16.3), *and brought the harmonization between the Polish tax system and the legislation in Western Europe.*

To fully comprehend the above paragraphs, one has to be able to grasp the consequences of trying to implement new taxes (PIT, CIT and VAT) by means of "manual procedures," using counting frames at the beginning of the 90s. In order to imagine these consequences, one should start from estimating how many counting frames and manual workers would be needed for that purpose and how much time would be spent to manually record and process declarations of approx. 25 millions of personal income taxpayers (including about 19 millions of those, who submit tax returns directly to a tax office), as well as declarations of corporate income taxpayers and value added taxpayers. It seems that the very act of generating, verifying, updating and using all the declarations of the common Tax Identification Number (NIP) would be infeasible by means of manual procedures. If we add to that posting to general and subsidiary ledgers, it is hard to imagine using such a system of tax revenue for making estimations, controlling, distributing, and forecasting.

Another *important consequence of the implementation of POLTAX was its paving the way for a better integration with the European Union and for a stable economic growth of the country. Besides, everyone who is up to date (i.e., 2017) with the situation in Ukraine and its struggle to achieve integration with EU may guess that launching, developing, and implementing such a project as POLTAX required great equilibristic skills, whose full presentation would force us to go much beyond the scope of this book.*

It is important to emphasize that since 1995 (after signing the agreement terminating the contract with BULL S.A.), the Ministry of Finance has been exploiting the POLTAX system in all tax offices in Poland. The technological design of the system provides the Ministry of Finance with a freedom of choice when it comes to IT suppliers. In particular, it is possible to develop POLTAX by internal project teams, without a need of outsourcing the task.

Currently, in 2017, the POLTAX system is up and running in all tax offices, supporting the key functions of the Polish tax system. In particular, it consists of the following modules, implemented and employed in the 90s, that constitute the following key subsystems of POLTAX:

Registration—registration of business entities on the basis of NIP application forms, collection and maintenance of data concerning business entities and their activities. Creation, implementation, and exploitation of the Central National Taxpayer Register (in Polish: Centralny Rejestr Podmiotów Krajowej Ewidencji Podatników). This is the core data subsystem in POLTAX.

Declarations— register of declarations, decisions and other documents related to particular taxes (e.g., PIT/CIT/VAT) and their consequent administration.
Accounting— register of taxed and untaxed budgetary receivables (attributions, deductions, payments, returns); tax settlement and revenue sharing between particular budgets and other creditors.
Notices— register of penalty notices and fiscal criminal cases.
Audit— audit planning, selection of entities to be audited, automatic support for tax, audit procedures administration and record keeping, reporting.
Debt Collection— supporting tax offices in debt collection.
Foundation— supporting management of ontology and business rules, used by succeeding versions of the POLTAX software.

The POLTAX system stores all the data related with taxes for all taxpayers/tax bearers. By means of functions available to all the users, tax officers have a full access to taxpayer databases and may obtain information for audits. A tax office, having an access to all the data, may check whether a given taxpayer meets her/his tax obligations. The POLTAX system, used in tax offices in Poland, has significantly improved the tax collection rate and the tax debt register (subject-oriented data warehouse) has helped to maintain control over tax arrears and carry out the collection procedures.

The integrated database of the POLTAX system is based on an extremely flexible data model, designed at the beginning of the 90s. The model allows for a consistent development and expansion of the system's functionality, in compliance with the ever-changing tax law, as well as the development of interfaces for POLTAX. Such interfaces, for example, help exchange data via the Internet and feed numerous dedicated data warehouses with reference of data from POLTAX.

What is particularly important is that interfaces ensure a smooth expansion of an existing IT system to a new system. At the moment, a system named e-Podatki (e-Taxes) is being gradually developed and implemented in Poland. It is expected that maybe after 2017, the system will possibly supersede the POLTAX system.

Taking into account the context against which the POLTAX program was developed in the early 90s, it should be noted that it represented a technological breakthrough in the Polish governmental administration and in the following years, it has been a reference point on the basis of which other projects were planned and designed in Poland.

At the same time, the POLTAX project contributed to the transformation of the tax system in Poland, from a centrally planned economy to a market-oriented economy. The main result of the project was the creation of an operational system, described above, which is subject to evolution according to the general principles of architecture evolution. There are several positive consequences that resulted from the implementation of the operational version of the POLTAX system. Probably, the most important ones include:

1. Comprehensive reform of the tax system, including the effective implementation of the following taxes: PIT, CIT, VAT (cf. Figs. 16.2 and 16.3).
2. Implementation of the strategic objectives laid down by the Cabinet of Tadeusz Mazowiecki, aiming at stabilizing the economic situation in Poland (particularly,

supporting the dramatic reduction of hyperinflation) and paving the way for the European Union Association Agreement, which, among others, brought about the harmonization of the tax system.

3. Flexible solutions used in the architecture of the POLTAX system, developed in 1990 and 1991, which, in the consecutive years, ensured a very smooth adaptation of the system to many legal and organizational changes. In fact, the architecture and the data model, after about a quarter of a century, are still in use.
4. Openness of the system architecture, which was particularly sharply criticized in Poland in the early 90s of the 20th century by a number of competency centers associated with major IT companies, and which stimulated parliamentary questioning and parliamentary speeches. From the perspective of almost a quarter of a century, the open system has proved to be a great advantage of POLTAX allowing MF to avoid establishing a lasting bond with only one IT provider.
5. The implementation of the POLTAX program in 49 tax chambers in Poland entailed intensive computer training in modern IT technologies, owing to the nationwide scale of the project. This significantly contributed to one of the largest in the 20th century transfers of information technology to Central Europe, which, until 1994, was hindered by the CoCom technology transfer restrictions.
6. The POLTAX program helped to educate numerous professionals in Poland in the field of modern technology, based on ORACLE's `Case*Method`, concerning the design and construction of software systems and relational databases, as well as later variants of this method (also from other companies). In addition, at that time, a modern IT platform architecture based on the UNIX operating system, with structured data exchange networks and their integration into a single system based on TCP/IP, began to be used in Poland for the first time on such a large scale.
7. In the post-communist Poland, within the framework of the POLTAX project, the first IT offset projects were launched. In particular, the Bull company provided hardware and software for modern computer labs at universities in Poland, facilitating the practical teaching of modern programming, database building, planning, and administration of computer networks. In addition, Oracle offered free licenses to all their products for educational purposes in Poland and a 90% discount on the prices of licenses used for the purpose of administrational support at universities and other educational institutions in Poland.

In retrospect, it is clear that it would have been impossible to achieve all of this, if a proper development strategy for POLTAX had not been formulated and implemented at the beginning of the 90s.

At some point, the author realized that the strategy, which had been intuitively implemented 25 years ago reflects some very general mechanisms related to interactive granular computation models based on WisTech, which were developed by the author in cooperation with prof. Andrzej Skowron later on. The work is presented further in the book.

Without a doubt, in the 90s, the author did not understand the crucial role of IGrC for CSE projects. To a large extent, he was driven by his experience in the implementation of IT projects, gained while working in the Department of Computer Science of the University of North Carolina at Charlotte, his passion for algebraic methods in logic, and his interest in the techniques of constructing models of mathematical theories.

Now, these intuitions might seem clear. The most important intuitions, connecting POLTAX-related experiences and IGrC models based on WisTech, presented later in the book, are the following:

1. *The dynamics of a project in all its aspects is a consequence of various types of interactions between c-granules (owned by project participants, stakeholders, project teams, and external persons) and physical objects. The interactions between c-granules of teams of people (these c-granules—i.e., established by a society of agents—are called meta-social c-granules, cf. FPW-3 in* Chap. 32) *representing various cultural habits, are particularly crucial for project planning and management. In these interactions, c-granules representing project plans, actions, and other different concepts, as perceived and understood by the participants of a given project, play a key role.* Practically, every person has a slight different perception of each such c-granule. In this sense, a whole project is, de facto, also a c-granule.
2. It is very important *to pay great attention to the evolution of interfaces, which facilitates communication between the participants of a given project, who use their owns jargons* (e.g., accounting jargon, programming jargon and legal jargon are totally different). However, it is also crucial *to skillfully manage the development of communication between project participants and the software* (both utility software, e.g., `Case*Tools`, and application software). In granular computations, this is related to the *approximation of c-granules of one of the agents by means of c-granules of another agent*. This kind of approximation (of agents c-granules) processes lead to the establishment of meta-social c-granules.
3. Current context against which a given project is implemented may hinder and distort cooperation between project participants. The context is unforeseeable. However, it is often possible to parameterize potential changes and be prepared for some unfavorable circumstances. It is difficult to construct specifications of context and related changes, using a precise language. It is important to acknowledge the fact that *these specifications must be constructed in a natural language, using complex vague concepts*. However, in general, when it comes to practical applications, it is enough to provide the approximations of these concepts. Therefore, it is important to skillfully improve these approximations and manage any changes that have been introduced by them.

4. When implementing a project, it is extremely important to *first, establish the ontology and next, manage its process of development*. The ontology should include not only concepts and relations between these concepts. It should also encompass principles governing the aggregation of concepts. The aggregation can be conducted on a grammatical level (e.g., how to construct formalized language of mathematics using logical connectives) or by means of defining new concepts through the aggregation of semantic components, taken from some other concepts. In this sense, we may have explicit aggregations (using logical connectives) and implicit aggregations (using semantic components of other c-granules).
5. *When modeling granular concept or interactive granular computations, one should not ignore their physical nature*. Generally, it is a very deceiving practice. For example, while discussing some data attributes, it is important to remember about the mechanisms of access to their physical sources and about physical mechanisms, by which attribute values can be computed.
   For example, let us imagine that the Minister of Finance is interested in the value of an attribute assigned to every taxpayer, which is either "true" or "false." The value is "true" when a given taxpayer evades at least 1 million euro in taxes a year. Of course, it is very difficult to provide an exact computation of this attribute. However, it turns out that the problem is similar to the one whose solution is presented in Chap. 18, regarding the Merix project.
   The Merix project is aimed at the real-time detection of credit card frauds. According to the author, such significant tax evasions as described above are easier to detect than credit card frauds. Firstly, tax evasion detection does not require such demanding computations and real-time decision-making processes. Secondly, the data concerning taxpayers are much more complex—the data sets are a quarter of a century old. Furthermore, numerous hypotheses regarding the construction of an approximation for such an attribute may be physically verified in time. Moreover, computations of the "tax evasion" attribute can be complemented with some simple adaptive and interactive processes, which may improve the accuracy of the approximation.
   Therefore, there should exist a possibility to generate a much more accurate approximation of the attribute than it was in the case of the Merix project. However, in order to construct such an approximation, a series of physical experiments on real data has to be conducted. Without these experiments, it is impossible to construct these types of classifiers.
   However, once a precise approximation of the "tax evasion" attribute is constructed, it may turn out that the lawyers of such tax evaders are able to prove that these taxpayers "do not pay taxes" in accordance with the applicable law. Such interactions resemble an endless game, like the metaphorical cat-and-mouse game. Unfortunately, in this game, the roles may be reversed. In such a situation, it is additionally important to ensure the safety of all employees of the revenue service. It is an important aspect of an interaction in a physical world, which should not be overlooked in computations of such an exemplary attribute.

6. It is convenient when *a team involved in the implementation of a given project understands the principles behind an interactive granular computation model that is being used*. For example, such a model can be based on FPW postulates, presented in Chap. 32, or on their proper adaptation and/or modification. *A systematic and interactive verification, along with the evaluation of assumptions are necessary for introducing any corrections to them.*

16.6 POLTAX: Conclusions

From the perspective of a quarter of a century that has elapsed since the birth of the POLTAX project and taking into account some of the aforementioned aspects, it is very difficult to succinctly summarize the most important conclusions that could be drawn from the implementation of the project, especially given that our knowledge and understanding of the risk management mechanisms in all types of complex projects related to organization and technology, was enriched in the last quarter of a century.

In the opinion of the author of this book, the POLTAX project could be implemented mainly thanks to the unique atmosphere of the working environments. This unique atmosphere was characterized by positive and constructive relationships among people working on the project. These relationships were primarily based on mutual respect between people of high personal culture. Moreover, the decision-making process was based on rational arguments, resulting from the in-depth analyzes of possible costs and effects of various alternative solutions. It should be strongly emphasized that the working atmosphere in the office (which was MF) at that time in many ways resembled the constructive atmosphere and enthusiasm, related to the formation of the Solidarity movement, which consisted of over 10 million members. Certainly, the mechanisms and atmosphere of a peaceful social movement are worthy to have a separate analysis that could provide a lesson for the future.

These conclusions show the necessity of taking into account not only knowledge, as an important factor in building complex systems, but also the interaction between people of high culture, closely synchronized with the network systems in which the adaptive judgment (represented by the organizational structure of the project – adapting to the rapidly changing conditions of the project) controls the changing priorities and needs of both the system that is being built and those participating in its implementation and development.

In other words, referring to the metaphorical Wisdom Equation defined in the meta-equation (8.1), we are talking about the three pillars of the concept of the WisTech program, namely: interaction, adaptive judgment, and knowledge [244, 245, 247, 253, 448].

There are many participants that have contributed to the positive effects of implementing the POLTAX program. Certainly, various participants and observers of the program have different opinions. From the perspective of the people who created and directly managed the project from the inside, it seems that after so many years, the most important environments, that helped achieve the above-mentioned positive results include the following:

1. Environment consisting of the immediate surroundings of Prime Minister **Tadeusz Mazowiecki**, especially Deputy Prime Minister **Leszek Balcerowicz**, who worked very closely with Mazowiecki and was responsible for MF, as well as close associates of Balcerowicz in MF who dealt with taxes, i.e., Deputy Minister **Stefan Kawalec** and the advisers: **Robert Koński, Witold Radwański**, and **Jacek Karpiński**. In the first half of 1992, Prime Minister **Jan Olszewski** played a crucial positive role in the stabilization and development of POLTAX (in particular, he paved the way for the integration of the tax system and customs system in Poland).
2. The **Bull company and the French government, without whose support the POLTAX would probably have never been created in Poland**.
   At the initial stage, this environment was established and coordinated by the Vice-President of Bull—**François Genet. He was a man of an extremely high personal culture, with diplomatic skills, expertise and deep understanding of the political context in Europe in the 90s of the twentieth century. In retrospect, one can safely say that through his effective involvement in the implementation of the POLTAX project, Bull had a significant and positive direct influence on the further development of the situation in Poland**.
   The French and/or Bull environment was substantially supported on an ongoing basis by the expert from the French Ministry of Finance—Chris L'Esteve. Moreover, the French and/or Bull components were following:

   a. Bull's Vice-Presidents: François Genet, Didier Ruffat, and Daniel LeCogic,
   b. the initial Bull project directors: Serge Depuy, Alain Acton, Chris Edwards, and Eamon Cooper,
   c. the software development and documentation managers: Nick Aggarter, Waldemar Birk, Jimmi Rammy, Rene Koelblen, and Konrad Makomaski,
   d. the system and network manager: Jacques Corvaisier,
   e. the first chief of the Bull office in Poland—Tadeusz Zondziuk.

3. The Polish business environment also supported rapid prototyping of software solutions for network and system platforms, as well as the verification and validation of some project products. In this regard, it is important to mention the following companies: ProVision of **Andrzej Dobosz** and Unidatax of **Mirosław Tomalak**.

4. External lawyers from Hogan and Lovelles, coordinated by **Bob Kenney**[27] who is de facto an architect of extremely valuable and important legal solutions, including a complex structure of the contract with Bull played a great role. Thanks to these solutions, this project could be successfully implemented. Bob Kenney's unique experience in implementing large IT projects in the U.S., which he skillfully adapted to specific conditions in Poland of 1990, was another important factor.
5. Domain experts from **IRS** who, under the assistance of the **U.S. government** (especially Jim Giannamore), supported MF in their high-level expertise in the field of designing and building IT systems for the tax service. They were also the ones who, in the early years of the program, recognized the importance of the IV&V principle.
6. The environment of different countries, which were very positive about supporting the emerging modern administrative structures and the transformation of the economic system from a centrally-planned economy to a free market economy. The attitude of the governments from such countries as **Switzerland** (including funding the activities of tax and IT experts from Ernst and Young, conducted as part of IV&V), **Denmark** (providing financial support to volunteers and consultants and organizing a very interesting reference visit, illustrating the organization and functioning of MF and tax authorities in Denmark), and Ireland (supporting the conceptual and organizational work) deserve particular distinction.
7. MF and tax administration experts were involved in and supported all types of works related to the construction and development of the POLTAX project. Not to mention all the people who contributed to the numerous achievements of the POLTAX project through their hard and very good work. When the project started to be developed, there was a unique atmosphere of work, harking back to the best days of the Solidarity movement in Poland in the 80s. This positive and constructive atmosphere of work was a major factor that enabled the formation of the IT team, consisting of several hundred IT experts in MF and its subordinate organization units in a very short period of time.

   Many trusted employees from the Department of Information Technology, tax offices, and tax chambers, who worked in a close collaboration with the author, especially contributed to the project.

   In particular, the support from **Janina Derwiszyńska** was extremely important for the success of the project. In addition, also deserves mention a very large contribution to the POLTAX project of such people as: **Roman Bącal, Marek Prussak, Michał Zalewski, Grzegorz Wojnarowski, Józef Adamus, Ludwik Lewandowski, Waldemar Szumieł, Wojcich Gargas, Roman Nowicki, Roman Zawodniak, Zbigniew Żak**, **Piotr Stodółkiewicz**, and **Antoni Michalski**.

[27] https://www.linkedin.com/in/robert-kenney-01683633/.

8. Once again, it should be strongly emphasized that all the positive effects of the project POLTAX project are the results of the cooperation of many complex dynamic network teams (not all of them are listed above). There were also many other valuable participants in the project (especially from: Ministry of Finance, tax chambers, tax offices, and from the company Bull), who contributed to the success of the project.

The author would like to end this chapter by reemphasizing the importance of the human factor in CSE projects, which cannot be reduced only to the so-called "window dressing," that has been already discussed (cf. Sect. 16.3.5).

It should be highlighted that an active participation in projects like POLTAX cannot be detached from one's personal life. In fact, one should acknowledge the fact that in such a case, her/his family and privacy will be largely affected. It is important to remember about this sacrifice when deciding to take up such a challenge.

Speaking about the professional life, POLTAX was undoubtedly the most important yet the most difficult project in the life of the author of this book. He devoted almost 10 years to this project, working very hard, under an extreme pressure. Today, the author derives great satisfaction from the achievements of the project, both when it comes to supporting the implementation of the tax reform in Poland, which paved the way for its economic growth, and contributing to a massive transfer of the most recent information technologies to Poland. Therefore, the author is grateful to Leszek Balcerowicz and Stefan Kawalec, who had faith in him and entrusted him with such a gigantic responsibility.

The author would also like to emphasize that many other participants of the project (both from MF/tax chambers/tax offices and the Bull company) agreed that this was the most important and, at the same time, the most difficult project of their lives—and, what is most important, these were not only Poles.

For example, two members of the managing team from BULL, most distinguished in the context of the POLTAX project, François Genet (France) and Eamon Cooper (Ireland), shared with the author their joy and gratitude for the possibility of being a part of such an incredibly important project. During the final stages of their lives, they arrived to Poland to say goodbye to the author.

At this point, it is worth mentioning that together with Eamon Cooper (representing BULL), we (i.e., the Ministry of Finance in Poland) have prepared a very interesting and important IT project devoted to **Government Finances: Research and Education Competence Center**. One of the objectives of the project is to transfer and develop high-quality IT know-how in Central and Easter European countries (especially in such Ministries like the Ministry of Finance, Economy, Tax, and Custom Administration). Unfortunately, this very important project still awaits the adaptation to the current (much bigger) demands, without its initialization and development is impossible.

Of course, the subjective summary presented above does not pretend to fully summarize the POLTAX project. Its aim, above all, was the presentation of the most important issues and mechanisms that can help in the implementation of other complex systems. It is worth noting that the conclusions discussed above, as well as many more insights drew from the POLTAX project, which could not be mentioned here because of space constraints, remain equally relevant today and can be used in the future projects.

Chapter 17
AlgoTradix

17.1 AlgoTradix: Background, Genesis and Goals

Analogously, to two meanings of POLTAX (i.e., as an IT system and as an IT project, cf. Sect. 17.1) AlgoTradix has also two meanings.

AlgoTradix as an IT project, is the name of a software development project which was implemented between 2006–2011 by a group of companies organized around AdgaM Solutions under the name of AdgaM Group. The author of this book was involved to a great extent in the management of AdgaM Group.

The AlgoTradix project was concerned with the design, construction, implementation and development of various technologies for the implementation of algorithmic trading. The concept of algorithmic trading is understood here very broadly. Intuitively, it means software which analyzes changes in international electronic financial markets and on this basis, triggers the purchase and/or the sale of products and goods available in these electronic financial markets. Basically, we assume that the software is working autonomously and is monitored by a set of instruments for risk management in interaction with experts who supervise the proper functioning of the system. In other words, by algorithmic trading, we understand a complex interactive system processing various types of information and knowledge, with the involvement of human beings, electronic markets, and computer systems. The essence of algorithmic trading is an automatic support of adaptive judgment about electronic financial market risk factors.

Taking into account the fact that the management of AdgaM Group originated from academia and research, it was very natural that their long-term goal was the organization and management of a private educational and research center, that would focus on the development of AI technology, especially on its applications in clinical decision support. According to the author, the implementation of such a project, within the framework of state institutions and the so-called aid programs, requires involvement in both administration and bureaucracy, which called for a greater effort than that devoted to research. At the same time, such projects were largely doomed

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_17

to failure due to the lack of flexibility in terms of change control procedures, which is characteristic of research projects. In other words, at the beginning of the implementation of a research project, especially one that is related with the production of an innovative technology, there is not enough experiential knowledge, acquired from a number of experiments which should be performed in the case of any complex research project. As a result, the initial assumptions, as well as the methods and tools needed for a given project, may completely change. Any change in the scenario, in turn, requires the flexibility of change control procedures during the implementation of the project. The experience of the author in the implementation of the so-called innovative projects, financed by public funds, shows that the enthusiasm of a team, working toward the goal of the project could get destroyed by a bureaucracy, which demands too much time for documenting and reporting in order to meet the expectations of the sponsor. What is worse, when projects, even written by experts, counted globally, get rejected by the government clerks on the ground of "lack of competence." In other words, the practice shows that the current system of developing innovative economy, present in many EU countries, is inefficient. Therefore, it requires a great effort from scientists to actually seek to make an essential contribution for increasing the competitiveness of the economy of a given country as well as improving the quality of life of ordinary citizens.

Anyone who knows a little about the Lisbon Strategy (cf. https://en.wikipedia.org/wiki/Lisbon_Strategy) understands the complexity and difficulty of the dilemmas associated with the implementation of research programs in EU. These problems are particularly acute in economically weaker countries and escalate during successive waves of a global economic crisis. On the other hand, one can see how well private research programs can be managed, when the implementation of a research program is approached in a comprehensive manner and its participants are at the same time given the freedom to design appropriate changes according to the current high-tech manufacturing trends—of course, within the pre-determined scope of available resources. This can be illustrated by the achievements of both the American technology giants, such as Microsoft, Google, Oracle, or Apple, and a number of countries outside Europe, which through the creation of appropriate conditions, made in the second half of the 20th century a technological leap. In this regards, such countries as Japan, South Korea, Singapore, and Taiwan are of particular note. In these countries, an especially important catalyst for a technological progress was undoubtedly making their governments realize how important the role of innovative technology is in the modern world. This provided a favorable ground for launching effective mechanisms of cooperation between the private sector and the public sector, in order to achieve the leadership position in selected market sectors. At the same time, an extremely high level of synergy was achieved on one hand, by selecting market sectors with the demand that generates great sales opportunities in international markets (which contributed to an increase in financial resources) and on the other hand, by contributing to the creation of an infrastructure for the future wave of development in the country. The example of the Lisbon Strategy can be used as an illustration of the lack of such mechanisms that would integrate the EU policy in the field of cooperation with the aim of improving the competitiveness of the EU.

Taking into account the above considerations, it is clear that it is not easy to complete, even initiate, a project that would yield a completely new technology in a such country like Poland. If we assume that the implementation of any project requires dramatic changes in the experimental environment, human resources and access to external sources of data, then a project can successfully run, in the opinion of the author of this book, only with the financial support of private investors who accept the risk (mainly by Piotr Dziubański and the author of the book). Therefore, the intention of the author of the project was to take the risk associated with the production of technology using their own funds as well as the support of relevant cooperating investors. In this way, the idea of developing software which employs AI to "automatically make money" in international electronic financial markets was born.

Basically, the essence of the problem of building software that could support automated trading in international financial markets is very easy to formulate, and boils down to adaptive judgment of an answer to the following question:

Do we expect a market crash?

In other words, if the market behaves quietly and moves on a stable trend (upward, downward, or sideways), then, theoretically one should be able to make money without much risk. Such a peaceful idyll of making money can be interrupted by the collapse of the market, which can result in a sharp change in the upward trend. If we are able to withdraw ourselves from the market in proper time, we can avoid unpleasant consequences of the rapid market turbulence. Unfortunately, the very concept of a "market crash" is not a precise term that could be easily defined on the basis of clear and unambiguous criteria. Rather, it is a classic example of a vague concept, i.e., it is not possible to precisely define the boundaries on the basis of which one can decide whether a market crash has already arrived, or whether the current market behavior pattern is leading to such a crash, following the line of past. In the last around 400 years or so of close market scrutiny, from the Tulip Mania Crash (1636–1637) to the South Sea Company Crash (1720), Stock Market Crash (1929), LTCM Trading Crash (1998) and contemporary manifestations of a financial crisis, people are still unable to find effective tools to predict an economic slump.

In Fig. 17.1, one can see a strong resemblance between different market crashes from the last 400 years.[1]

Furthermore, by observing the crush-like behavioral patterns of markets in terms of the S&P500 indicator, one can see a very strong similarity to the market turbulences which accompany with a market crash (cf. Fig. 17.2).[2]

The similarity in the behavioral patterns of markets relative to the time, during which a market crash occurs, is illustrated in Fig. 17.2. This provided a strong stimulation for the construction of various approaches to the algorithmization of the decision-making processes related to risk assessment and decision-making in terms of when and what to buy and sell.

---

[1] http://www.calculatedriskblog.com/2008/10/comparing-stock-market-crashes.html.

[2] http://www.calculatedriskblog.com/2008/10/comparing-stock-market-crashes.html.

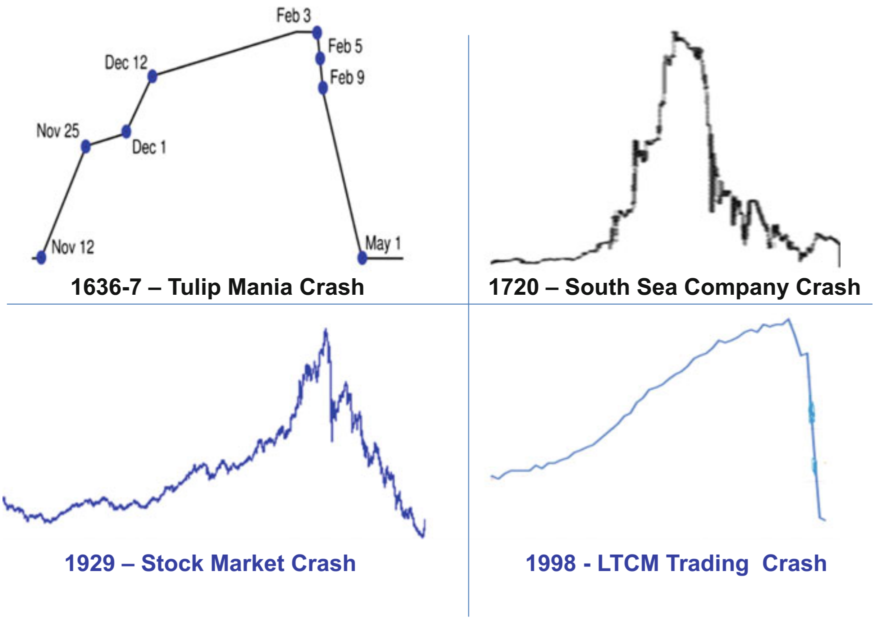


Fig. 17.1 Large crashes over the last 400 years

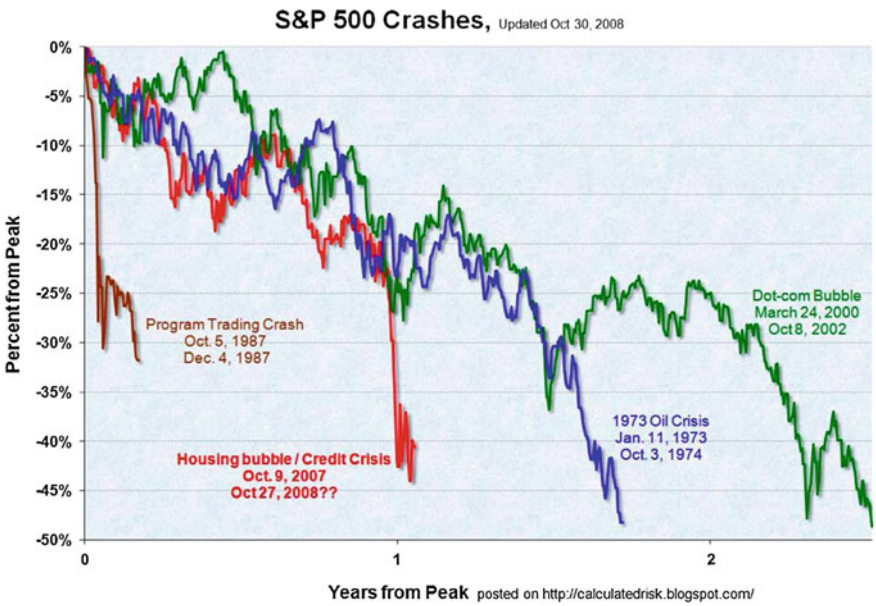


Fig. 17.2 Behavior of the large crashes over the last 400 years

Throughout almost the entire history of AI, a variety of techniques were created to support generations of winning strategies in different types of games (in particular, the market trading). In the 80s of the 20th century, numerous companies declared the use of different types of AI techniques to support the decision related to purchase and sale in the financial markets. However, many of these companies achieved spectacular success, whereas others have unfortunately failed.

Especially the achievements of the Renaissance Technologies company, founded in 1982 by a mathematician, James Harris “Jim” Simons,[3] are worthy of a great deal of admiration. Renaissance Medallion Fund has been in business for about 30 years and still enjoys spectacular successes. For example, since 1989, the company’s $5 billion Medallion Fund has averaged 35% annual returns after fees. Owing to these results, the Renaissance Medallion Fund is one of the best hedge funds in the world. It should be noted that the Renaissance Medallion Fund has also had extremely good results in the financial markets since 2007, during a deepening crisis. This in one hand causes a remarkably good result as the observers of the financial markets started having an enormous interest in the company, on the other the company itself was shrouded in mystery as the owners were reluctant to disclose the technology that is in use. Therefore, one cannot expect reliable and verifiable information on the methods used by the company, which would be available to public sources. For example, websites were extremely brief and only contained information regarding open vacancies for Ph.Ds. in Computer Science, Mathematics, Physics, Statistics, or related disciplines.

The information available on the Internet includes a portfolio of Medallion Renaissance Fund. For example, during the second quarter of 2012, the fund had a total of 3356 positions. Some of its top holdings as of June 30, 2012 are the following: NKE, GS, IBM, CRM, AMZN, F, TMO, FAST, MDT, BLK, HCN, SE, TXN, SBAC, CMCSA, PCP, LNKD, JOY.[4] On the basis of various statements of those involved in this fund, one can only guess, more or less accurately, what techniques are used by them. Probably, one of the strategies takes advantage of the inefficiencies in the execution of large transactions. One of their algorithms probably determines whether a very large order is executed and then front-runs it. As a result, Medallion bears high transaction costs and high expenses.

However, it should be noted, that in some sense this approach to algorithmic trading is very close to rough border of transparent and/or non-transparent activities. Some algorithmic trading transactions can use specific domain knowledge available for experts from financial institutions, law firms, accountants, political environments, and/or others. The knowledge can be used to design and implement complex financial structures to take advantage of and, at times, abuse or violate tax statutes, securities regulations, and accounting rules. In particular, on July 22, 2014, Simons was subject to bipartisan condemnation by the U.S. Senate Permanent Subcommittee on Investigations for the use of complex barrier options to shield day to day trading (usually subject to higher ordinary income tax rates) as long-term capital gains.

---

[3]https://en.wikipedia.org/wiki/James_Harris_Simons.

[4]http://stockpickr.com/pro/portfolio/renaissance-technologies/.

> Renaissance Technologies was able to avoid paying more than 6 billion in taxes by disguising its day-to-day stock trades as long term investments,

said Senator John McCain (R., Ariz.), the committee's ranking Republican, in his opening statement.

> Two banks and a handful of hedge funds developed a complex financial structure to engage in highly profitable trades while claiming an unjustified lower tax rate and avoiding limits on trading with borrowed money,

said Senator Carl Levin (D., Mich.).[5]

In the Report of the U.S. Senate Permanent Subcommittee on Investigations of July 22, 2014, one can read[6]:

> This investigation offers yet another detailed case study of how two financial institutions - Deutsche Bank AG and Barclays Bank PLC - developed structured financial products called MAPS and COLT, two types of basket options, and sold them to one or more hedge funds, including Renaissance Technologies LLC and George Weiss Associates, that used them to avoid federal taxes and leverage limits on buying securities with borrowed funds.

Probably the most spectacular failure of algorithmic trading was the collapse of LTCM (Long-Term Capital Management). LTCM was founded in 1994. The members of LTCM's board of directors included Myron Scholes and Robert Merton, who shared the 1997 Nobel Prize in Economic Sciences for a "new method to DETERMINE the value of derivatives." Initially successful, with annualized returns of over 40% (after fees) in its first years, in 1998, it lost \$4.6 billion in less than four months, following the Russian financial crisis which required a financial intervention by the Federal Reserve (i.e., ten years before the Lehman Brothers bankruptcy case). The fund was liquidated and dissolved in early 2000.

The above two examples of algorithmic trading projects explicitly show the importance of domain expert knowledge. Identification, representation, and implementation (in software) of valuable pieces of domain knowledge is one of the key critical success factors for any CSE project. The simple computational models for such complex applications just does not work. In particular, for algorithmic trading applications it is not sufficient to find a complex mathematical equation (or even implementation of all equations of today version of "financial mathematics"). Additionally, it is important to use knowledge about identification of market risk factors (especially political, legal, behavioral, and economical factors) and next it is necessary to implement appropriate risk management algorithms. Of course, algorithmic trading history is much richer than these two interesting examples. However, it is not our goal to present the whole history, but to provide a broader context and emphasize the diversity of the AlgoTradix project, especially when it comes to its development and implementation.

---

[5] https://www.hsgac.senate.gov/subcommittees/investigations/hearings/abuse-of-structured-financial-products_misusing-basket-options-to-avoid-taxes-and-leverage-limits, https://www.bloomberg.com/news/articles/2013-07-01/simons-strategy-to-shield-profit-from-taxes-draws-irs-ire.

[6] https://www.hsgac.senate.gov/subcommittees/investigations/hearings/abuse-of-structured-financial-products_misusing-basket-options-to-avoid-taxes-and-leverage-limits.

17.2 AlgoTradix: Examples of CSE Principles

In this section, we assume that the reader is familiar with the concepts (including principles) presented in Sect. 14.3 (especially with the ABC book principle and its extension in the form of the BPCD principle), and principles specific to CSE projects aiming to develop new AI technologies, especially *Crocus framework principle* (cf. Sect. 35.2 presented on the Fig. 35.1.

The main difference between POLTAX and AlgoTradix lies in the fact that the POLTAX project was based on the already existing technologies and their adaptation to the conditions in Poland in the late 80s and early 90s. However, the AlgoTradix project is in its essence based on the creation of new technologies that could potentially be applied in algorithmic trading. This profound difference resulted in the formulation of different specific principles for AlgoTradix, as compared to the principles used in the POLTAX project. However it should be underlined, the BPCD principle has an universal character and should be applied also to the AlgoTradix project. The most important specific principles related to the implementation of AlgoTradix can be grouped into the following three areas of principles:

1. PDCA development of machine learning technology ("ACADEMY for algorithmic robots");
2. Risk management adjusted to algorithmic trading specificity (i.e., specific domain knowledge for algorithmic trading);
3. Efficient and secure PLATFORM development.

One can group these specific principles in the form of a triangle, as shown in Fig. 17.3.

Intuitively, ACADEMY denotes an environment in which a new technology, aimed at implementing the idea of algorithmic trading, is being tested. The main substantive core of this process is the design and verification of different approaches to machine learning. The machine learning technologies were developed according to *Crocus framework principle* (cf. Sect. 35.2 presented in Fig. 35.1).

Machine learning includes both learning from multivariate time series reflecting the changes in the markets and all economic statistics related to the functioning of these markets, including the analysis of texts appearing in the news that could have an impact on the changes occurring in the markets. In this aspect, the project corresponds to another case study—Excavio (cf. Chap. 19).

It should be noted that applying appropriate measures to verify the robustness of a given technology under "forecast" of the market disturbances is no less important for the success of the project than the method of machine learning. In other words, the patterns of behavior observed in the last 10 years may prove to be quite inadequate for a correct behavior of the algorithms in the future. Therefore, the research in machine learning should be extended with intensive research related to stress tests or the verification of the technology behavior under the "forecast" of market disturbances. In general, the "forecast" of market disturbances are generated using the Monte Carlo approach, based on our understanding of potential market changes and market crash

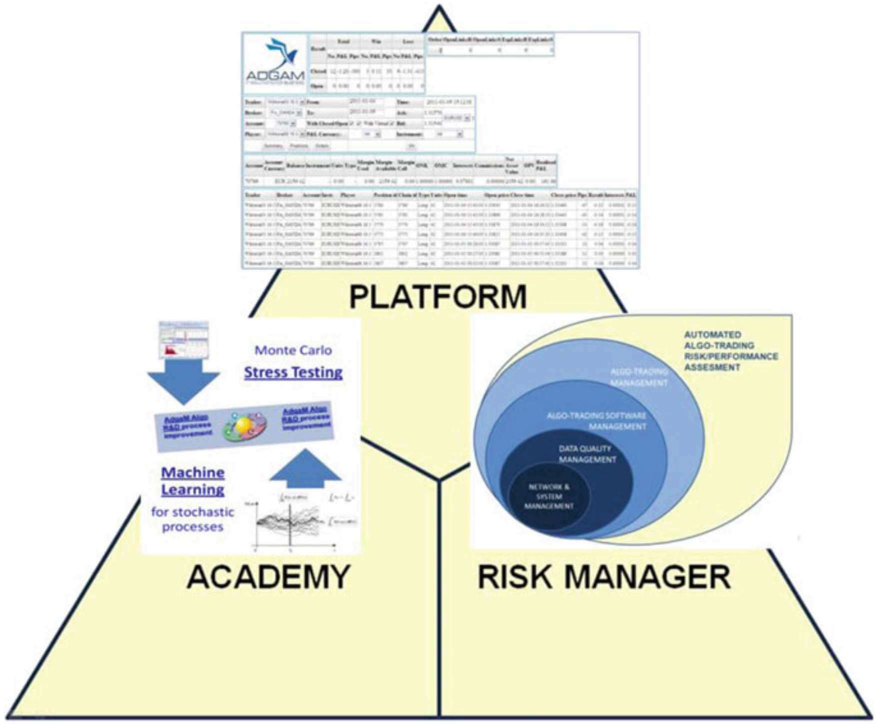


Fig. 17.3 Triangle of principles of the AlgoTradix project

mechanisms in the future. This is a crucial factor of the domain knowledge, necessary for machine learning and the technology testing for algorithmic trading.

With even the best components of machine learning and Monte Carlo stress tests, we are still vulnerable to random events if we do not take into account the most difficult challenges, including a smooth transition from one machine learning paradigm to the other through a skillful process control in a form of an appropriate adaptation of the Deming cycle, known as the Plan, Do, Check, Act (PDCA) cycle. In other words, the essence of the ACADEMY architecture lies in an interactive modification of machine learning and paradigm testing in such a way that on one hand, the best possible stress tests are discovered and on the other hand, previously collected experience is used to produce new paradigms in the process of shifting paradigms. The three components of ACADEMY (machine learning, stress tests and PDCA) are shown in Fig. 17.4.

Risk Manager is another component of the architecture, which is next to the ACADEMY in Fig. 17.3. It consists of software and software-supporting procedures, used online to monitor various types of risks associated with a given operation and transaction that is being executed. This system is a multi-dimensional analysis of the

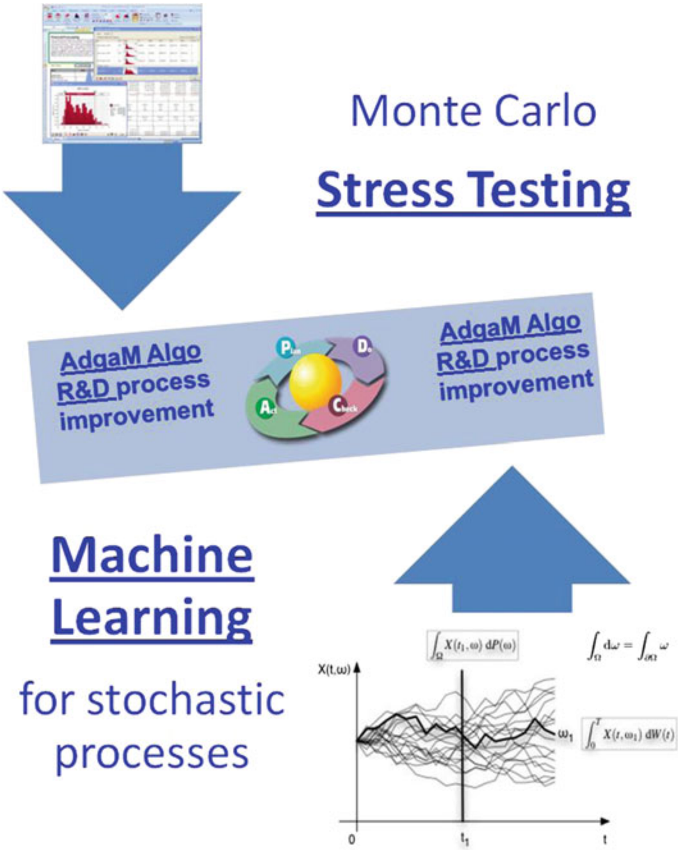


Fig. 17.4 ACADEMY components

risk, encompassing all the essential aspects that may affect the risk. A typical set of such aspects is illustrated in Fig. 17.5.

Figure 17.5 is a very general representation which, on its own, is insufficient to clarify a number of important architectural decisions related to risk management in e-commerce. Therefore, it is necessary to further refine the concept. The implementation of certain aspects of the ISO 31000 standard and related standard recommendations constitutes of an example of the details connected with the automated risk management system architecture.

To illustrate this approach, let us imagine that we automatize the risk assessment process described by the ISO 31000 standard recommendations. Figure 17.6 illustrates a general scheme of the risk management inference framework and Fig. 17.7 provides some more details to this general scheme (i.e., risk sources and mitigation controls).

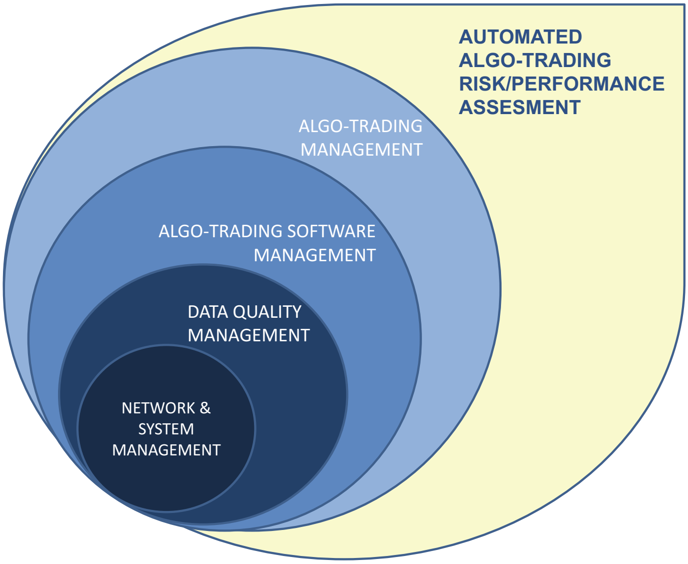


Fig. 17.5 “Teardrop”: Automated algo-trading risk/performance assessment

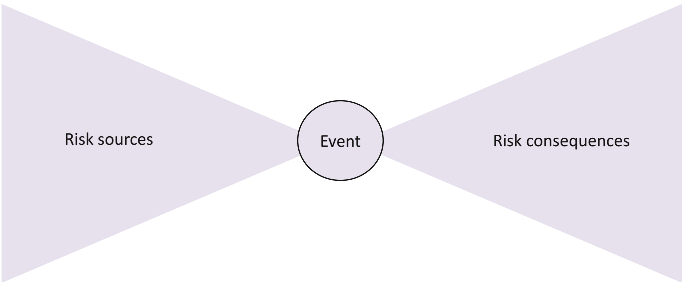


Fig. 17.6 Visualization of the risk management inference framework by the bow tie diagram according to the ISO 31000 standard recommendations

In Fig. 17.8, information about inference mechanisms, based on the original bow tie diagram, and the analysis of a context and sensor data are shown. Moreover, as the result of the inference, output sensor actions and scenario recommendations are produced.

The diagram should present the arrangement of intelligent software robots, performing risk assessment and drawing conclusions regarding possible necessary actions, on respective vertices. When it comes to the construction of such algorithms, there is a strong preference on our part for the approach that is based on interactive granular computations, performed by hierarchical structures representing rough sets and related logical structures. Schematically, this can be illustrated as

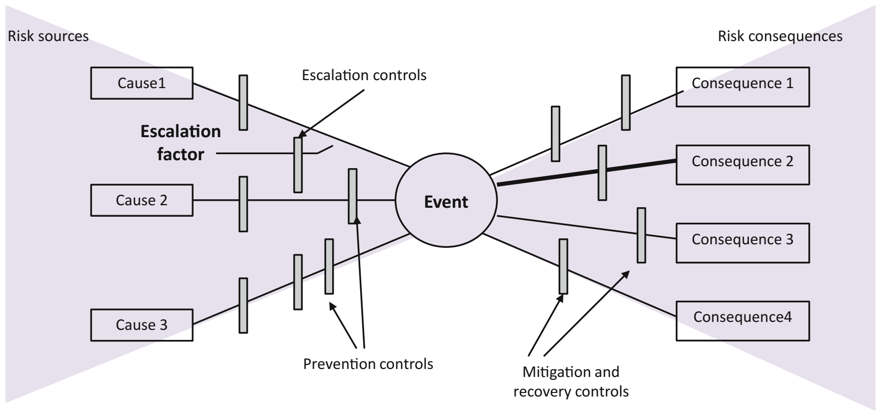


Fig. 17.7 Inference engine description by Fig. B.8 from the ISO 31000 standards—Bow tie diagram for unwanted consequences

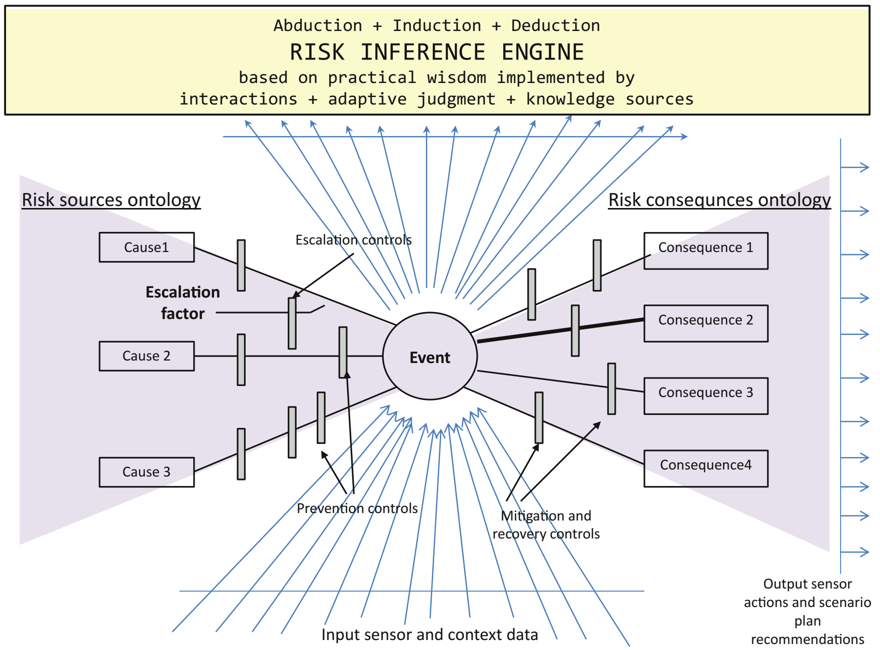


Fig. 17.8 Extended bow tie diagram based on ISO 31K (additionally inference engine and optimal selected action path for risk treatment are shown)

in Fig. 17.9. We treat each node as software-intelligent robots which use their local logics for knowledge. This is presented in Fig. 17.10.

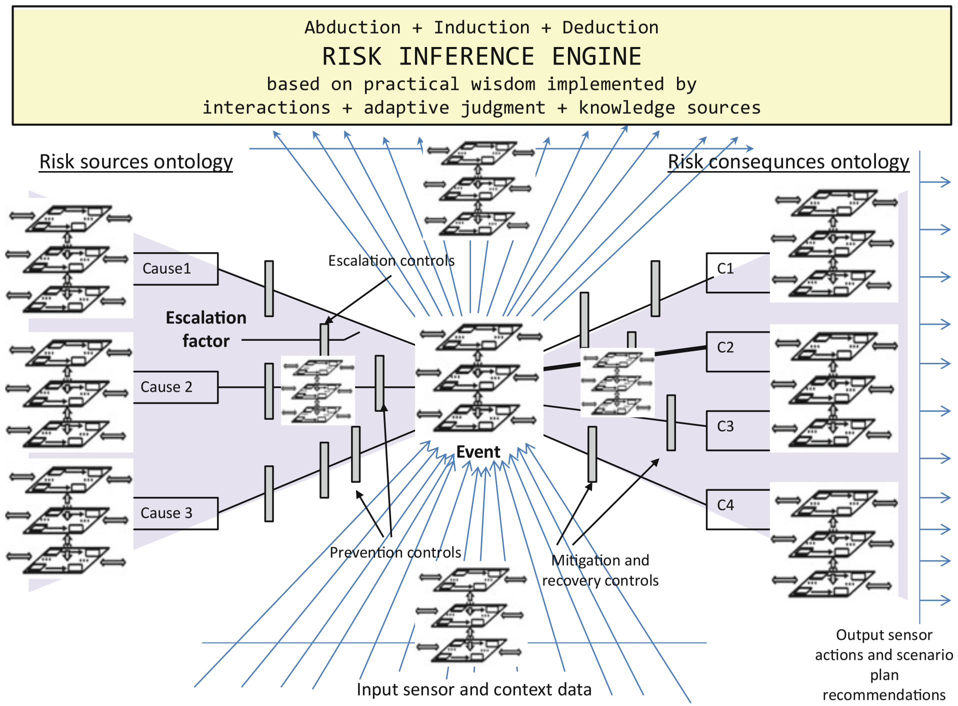


Fig. 17.9 Illustrative example of a network of interacting hierarchies of information systems obtained from extended bow tie diagram based on ISO 31K (additionally inference engine and optimal selected action path—marked by bold line—for risk treatment are shown) extended by hierarchical interactive structures illustrating complex structures implementing causes, consequences (C_1, C_3, C_4 are abbreviations for *Consequence* 1, *Consequence* 3 and *Consequence* 4, respectively), and event as well their interactions

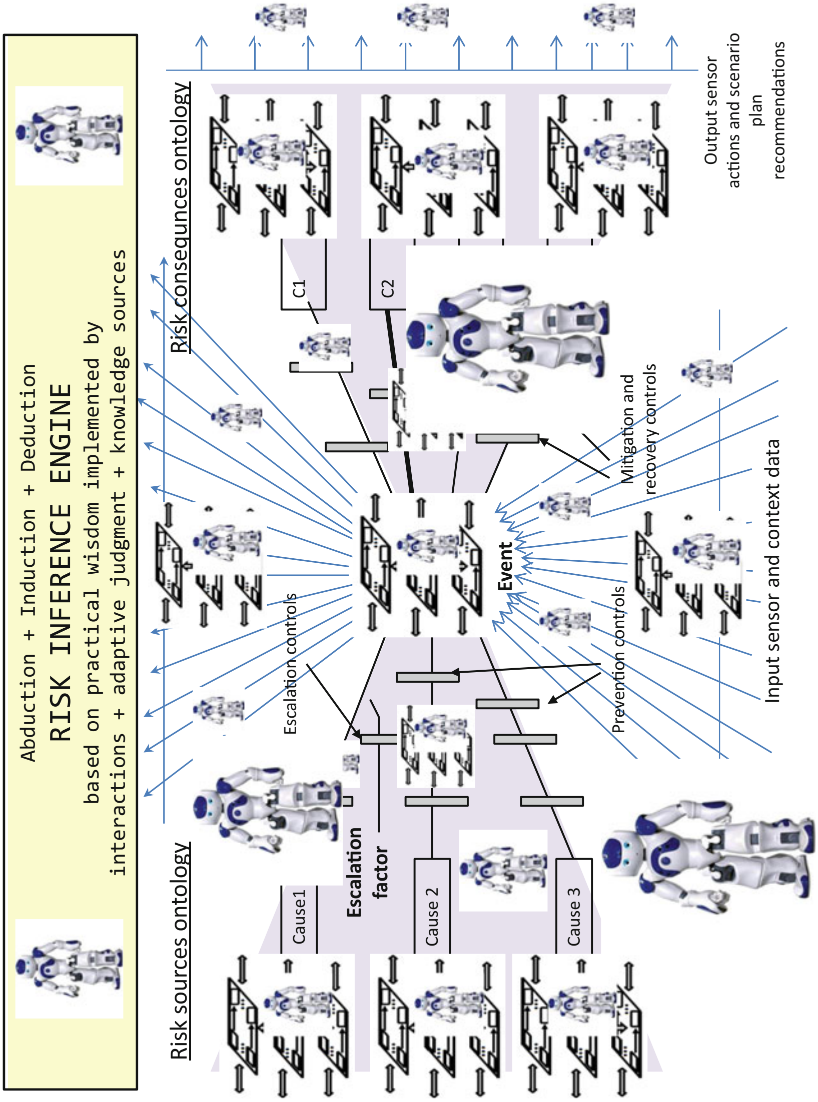


Fig. 17.10 Extended bow tie diagram with software-intelligent robots

The above schematic approach to the architecture of the Risk Manager component will be developed in subsequent chapters of the book devoted to WisTech (e.g. Chap. 33).

In addition to these architectural decisions, which set out the rules for the construction and development of a typical algorithmic trading system, it is necessary to take into account a number of CSE principles that should be present in all traditional IT systems (e.g., BPCD principle). These rules can also be designed based on the example of the POLTAX project.

17.3 AlgoTradix: Brief History

The AlgoTradix project was launched in the autumn of 2006 and was carried out with an impressive growth up to the first quarter of 2011, when because of the lack of further funding, it began to ending in the autumn of 2011. In a monthly basis, average 15–20 people took part in the project. On average, the team consisted of about 3 Doctors of Science with habilitation qualifications and a number of Ph.D. students. It is estimated that the total cost of this project during 2006–2011 was more than 1000 man-months. These costs were covered by private investor sources. At that time, more than 100 mathematical models based on different approaches to algorithmic trading were implemented and tested on.

Over 1,000 complex experiments, verifying the quality of functioning of the models that had been developed, were carried out and documented. The following four software environments were developed:

1. HIGH SCHOOL—an environment for rapid prototyping and testing of new mathematical models (not yet implemented in ACADEMY);
2. ACADEMY—an environment for planning experiments and generating Monte Carlo data for the purpose of these experiments (including obtaining the data from reliable sources), start-up and evaluation of experiments;
3. PLATFORM—an environment for trading on a real account (using real money) as well as on demo and virtual accounts;
4. DEVELOPMENT—an environment for the construction and development of software.

The software supporting the functioning of the four environments had the length of more than 300,000 lines of code in the final phase of the project.

At the beginning of the project, it seemed that the advanced technologies used in the evolutionary genetic programming, which represent logical formulas expressing different market situations using dozen thousands of features, should be a sufficient mechanism to quickly obtain stable and positive results. However, it soon became apparent that this approach was not valid in terms of playing on a demo account as well as in a number of experiments on data generated using the Monte Carlo methods. For the first months, the results of experiments carried out on a more or less complex logical formulas, expressing aspects of tens of thousands of changes in the market, consistently led to "stable" negative results. Undoubtedly, the first failure was a big surprise. For the whole team, it was a great lesson of humility.

This was a great learning experience for a whole team engaged in the project which could experience the difference between the theory and practice of algorithmic trading. The fundamental conclusion drawn from this first experience was that the key to success in algorithmic trading is risk management, based on varying assumptions about the adaptive ontology and adaptive logic of constructed mathematical models of risk management. The sets of these assumptions were called within the team, *algorithmic trading paradigms*. Along with the change of situation in the international financial markets, their algorithmic trading paradigms clearly needed changes.

The history of the project largely comes down to the evolution of these paradigms. Each such paradigm was the basis for the verification of a variety of mathematical models that were based on it.

In retrospect, in the simplest terms, it can be assumed that roughly every six months, these paradigms of algorithmic trading were significantly altered. Quality assessment of paradigms and the resulting conclusions as well as recommendations for further modifications and paradigm shifting were the most important aspects of the process. Of course, it was done in the Deming PDCA cycle. In Fig. 17.11, we show the progress achieved in the subsequent semesters of the AlgoTradix project development in the ACADEMY and environment (cf. Figs. 17.3 and 17.4).

Basic indicators of the approach are expressed in annual earnings which are represented by return on investment (ROI) and risk, measured by the volatility (standard deviation of ROI) of the results ('Volatility' column in Fig. 17.11). Obviously, the experiments in the Forex market can be performed using various levers. The values of these levers are shown in the 'Leverage' column. These experiments were carried out on historical data and confirmed by the Monte Carlo perturbed data (similar to historical data).

It should be emphasized that apart from the above-mentioned experiments on historical data, a significant part of these results were confirmed in the real market, with the real money involved (for trading between Fall 2009 and Spring 2011).

17.4 AlgoTradix: Main Results

From September 2009 to March 2011, the AlgoTradix algorithms were tested using real money on the Oanda platform in New York.[7] Furthermore, regardless of the environment, tests were also carried out on other platforms, such as Baxter FX, FXCM, etc. Overall, this was to verify the effect of cultural and jurisdictional conditions present in different countries on the effectiveness of trading. Hence, a particular attention was paid to the environment of electronic trading platforms, legally registered in New York, London, Tokyo and Dublin. Despite appearances, it turns out

[7] http://www.oanda.com/.

Year	Strategy	ROI	Volatility	Leverage	Evolution of technologies applied for the first Milestone of the Algo AdgaM Development Roadmap
2011H2	Lucy	30%	7,5%	50:1	Occam's Razor approach to summary of five years of Algo AdgaM R&D
2011H1	Nicole	28%	8,0%	50:1	A Wisdom Technology approach to behavioral finance and game theory in algo trading
2010H2	WiktoriaB	28%	8,5%	50:1	Wisdom Technology based on rough granular computing
2010H1	WiktoriaA	25%	9,0%	50:1	Random Forests Classifiers optimalized by multiobjecive optimization using swarm intelligence technology
2009H2	Mary	23%	9,5%	100:1	Martingale technology, Taleb Distribution and Monte Carlo stress testing technology esp. based on ERM (Empirical Risk Minimization) and SRM (Structural Risk Minimization)
2009H1	Betty	21%	10,0%	400:1	Ontology engineering - especially building features based on Fourier and wavelets automatic analysis
2008H2	Sandra	14%	10,5%	400:1	Statistical learning classifiers especially constructed by hierarchical regressions models and by Support Vector Machines
2008H1	John	5%	11,0%	400:1	Rough sets classifiers
2007H2	Leon	2%	11,5%	400:1	Metaheuristics applied for machine learning of control parameters for stochastic processes based on generalizations of Black–Scholes model
2007H1	Ewa	1%	35,0%	400:1	Evolutionary programming - mainly based on genetic algorithms

Fig. 17.11 Progress in the AlgoTradix project development based on historical data

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
2009									5,12		3,22	-0,74	7,70
cumulative									*5,12*		*8,50*	*7,70*	*7,70*
2010	0,82	2,80	1,52	0,80	1,16	0,10	1,04	1,71	1,27	3,58	0,58	-0,71	15,62
cumulative	*8,58*	*11,62*	*13,32*	*14,22*	*15,55*	*15,66*	*16,86*	*18,87*	*20,38*	*24,69*	*25,41*	*24,52*	*24,52*
2011	0,50	2,11	0,87										3,51
cumulative	*25,14*	*27,79*	*28,90*										*28,51*

Fig. 17.12 Estimated effects of the scaling

that the implementation of the trading rules is quite significantly different in each of these environments.

These differences arise not only from differences between various ECN platforms and market makers, but from the country-specific legal constraints, culture and place on the international business map. After many studies carried out in the early stages of the project, it was decided that a good environment for a reliable testing was the Oanda platform. However, even in this environment, there were notable significant differences in trading on small amounts, e.g., a few thousand dollars or mid-sized amounts, e.g., in the range of half a million dollars. This lesson shows the importance of taking into account specific constraints of trading in developing mathematical models, which are used as the basis for expressing algorithms. When carrying out the trading on the Oanda platform, we used different amounts of money in different months. The only change in the account during a given month resulted from autonomous trade, carried out independently by robots. Basically, this trade was fully automatic (i.e., without human intervention), except for just a few incidents when the trading was stopped due to the lack of mental strength displayed by the supervisors or due to some technical reasons. On a side note, it is worth noting that this lack of mental strength of the supervisors generated losses rather than helping to avoid them because usually, it turned out that maintaining "risky" positions under these exceptional circumstances could have led to profits.

This was an important lesson in the context of the future supervision of the risk associated with algorithmic trading by the societies of software robots. Assuming the behavioral nature of the markets, the dominant patterns of behavior are largely a reflection of the collective psyche and emotions of the market participants. If the dominant part of the market performs transactions automatically generated by robots, that means that the patterns of market behavior are a reflection of the collective "mental and emotional" states of the robots.

As mentioned above, during one and a half year of trading, we started each month with a different initial amount. Then, it was assumed that the game started from some amount, such as one million U.S. dollars, and in the following months, the balance was modified depending on the results obtained during the previous month. In this way, it was possible to illustrate the entire course of the game during one and a half year. The estimated effects of this scaling are visualized in the table presented in Fig. 17.12.

		Live volume
Total return of investment in the period of time September 2009 – March 2011	ROI.Tt - Total rate of return [%]	28,9
Annualized return of investment	ROI.At - An. total rate of return [%]	18,0
Volatility	Vol.Ad - Annualized Daily Volatility . [%]	10,2
Sharp ratio	Sh.Ad	1,8
Percentege of Winning positions	Position Win Ratio [%]	79,7
Maximum drawdown of closed positions	MDD [%]	4,5
Maximum drawdown of all positions (including open positions)	OMDD [%]	10,6
Total no. of positions	Total number of positions	24 468
Daily avg. no. of transactions	Daily avg. no. of transactions	63,2
Max No. of successive winning pos.	Max No. of successive winning pos.	748
Max No. of successive losing pos.	Max No. of successive losing pos.	256
Maximum time to recovery of capital	MxTCR [days]	41

Fig. 17.13 The auto trade results of year and a half game on the Oanda platform

During year and a half of algorithmic trading on the Oanda platform, about 24 thousand positions (i.e., approximately 48,000 transactions) were completed. About 80% of them ended up with winning. On average, about 63 positions were played per day. The maximum number of sequential losing positions in a row was 256, while the maximum number of sequential positions in a row that ended with winning was equal to 748. The maximum time to recover the lost capital was 41 days. The maximal drawdown, taking into account the closed positions, was approximately 5%, while the maximum drawdown on the open positions was about 11%. Volatility was approximately 5.5%, while Annualized ROI was approximately 18%. Trading was carried out on just one currency pair, EUR/USD, with the leverage of 1:50. These results are summarized in the table presented in Fig. 17.13.

The data collected in the table shown in Fig. 17.13, constitute a fine substitute for a huge number of statistics that were used in the AlgoTradix project to monitor the performance and risk of the trading process. Of particular interest, seem to be partial higher-order statistical moments. The analysis of the sets containing hundreds of statistics on the risk itself is a very complex process, especially that at different times, different statistics are particularly important. In addition, the statistics should reflect the preferences of the client, who invests his (or her) money.

There is another complication as the risk profile is very dependent on culture, knowledge and awareness of the investor. Certainly, the most appealing form of visualizing risk-related transfer of knowledge is to represent different aspects of risk with appropriate diagrams, graphics or even an animation.

During the trading, particularly noticeable are two values: the involvement of capital as measured by the OPV (open present value) and DD (drawdown) in the closed position, and the drawdown on open positions (ODD). In Fig. 17.14, we show the value of OPV expressing the value of the involved capital (under the risk in game). It is shown over the x-axis (positive values of y). At the same time, under the x-axis (negative values of y), the changing value of DD is shown. Often, Fig. 17.14 is combined with a simultaneous visualization of the results. An example of such visualization is presented in Fig. 17.15.

Both these figures do not reflect the resilience and vulnerability (sensitivity) of the undertaken transactions to disorders of data that actually appear and are used in algorithmic trading. In many theoretical models of algorithmic trading, the fact that the data that is supplied can be significantly delayed is often not taken into account or even worse, not provided (both due to technical and organizational reasons). What is more, signals to buy, sell, etc. can be implemented with a delay and/or cannot be implemented at all (both due to technical and organizational reasons). However, practitioners are certainly well aware of the fact that algorithmic trading cannot be performed without the use of a reliable source of data, and strategies as reliable as possible, for the transaction performed by the algorithms. Thus, in all models, it is necessary to verify the risk associated with the sensitivity of the technology that is being used to various disorders in the data streams that are supplied, especially to the delays in these streams, which can sometimes last a few seconds or even longer.

In order to assess the risk arising from the delay in the opening/closing of the transaction and its negative consequences on trading, many forms of graphical visualization of risk are used. For example, Fig. 17.16 presents the impact of possible delays in the opening of the transaction and in the closing of the transaction on the final result.

Let us imagine that at the base point (0, 0), treated as the reference point, we achieved ROI as high as 18% by playing on real positions (i.e., about 24,000). For example, if we select a random point (4, 6), it means that we examine the result of a time series of positions opened with delay, using random exponential distribution and the expected delay value equal to 4 seconds, and positions closed with delay, also using random exponential distribution with the expected delay value equal to 6 seconds. In general, numerous time series were generated, and thus we received the pre-determined value of ROI (relative to realized real-life trading) in the situation where the random events were causing 4 seconds delay in the opening of each of the approximately 24,000 positions and 6 seconds delay in the closing of these positions.

Of course, we have to remember that a delay in closing means the prolongation of the duration of a given position, and not only a shift in the closing time (to avoid a situation where the closing occurs before the opening). In the case of the point (4, 6), one can see that ROI is reduced by about 5% in comparison to the reference value, i.e., with respect to 18%, and amounts to approx. 17%. The results presented

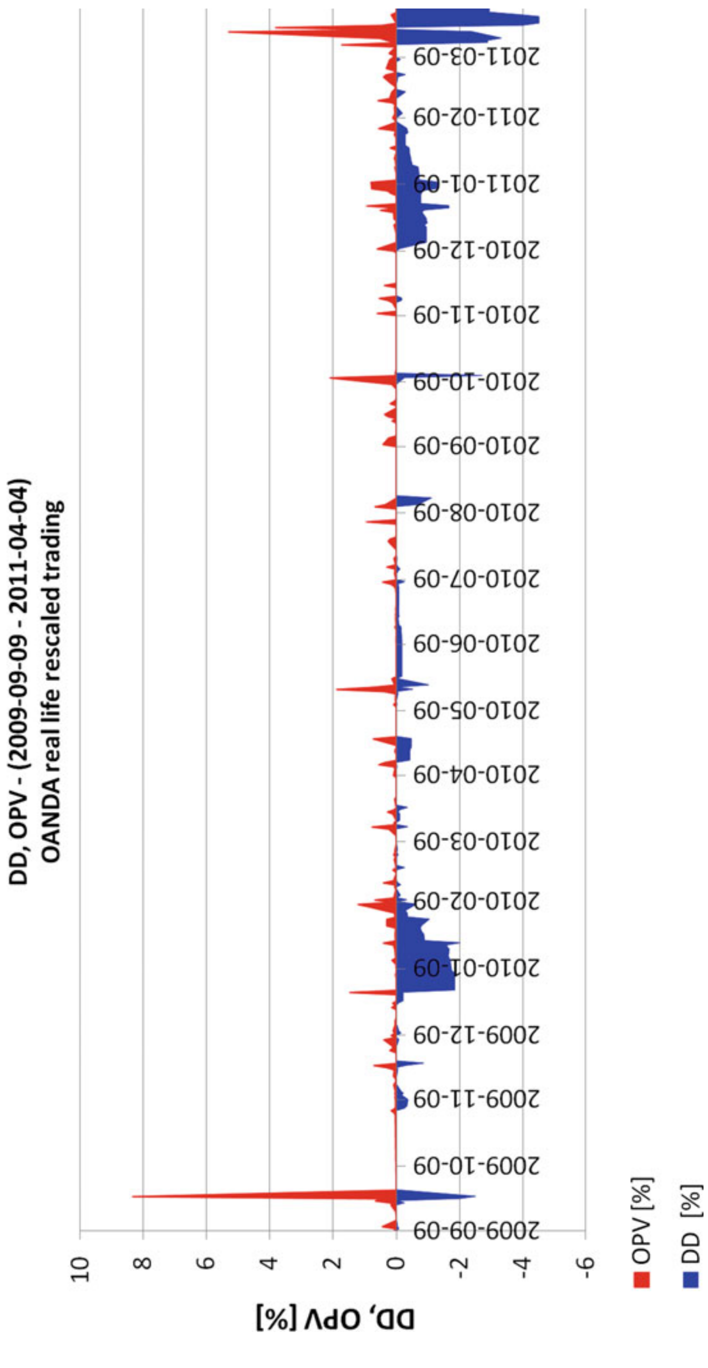


Fig. 17.14 Changes in the value of OPV

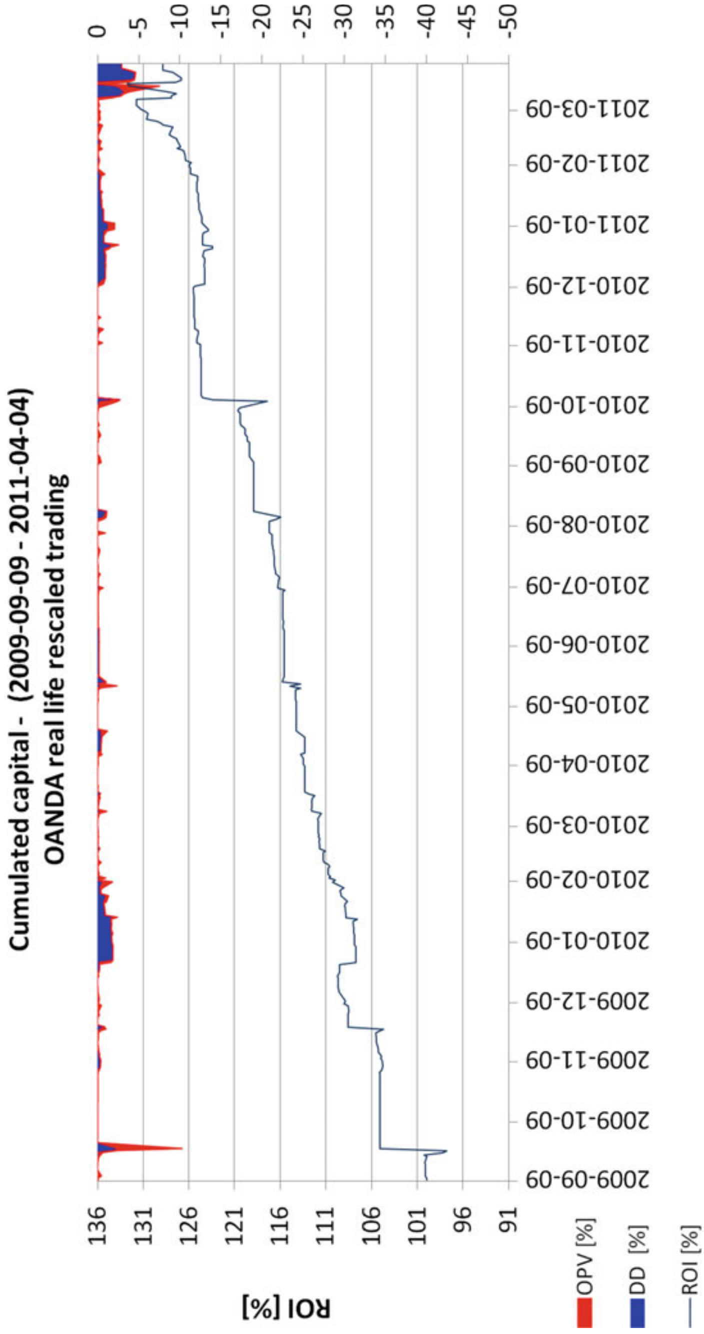


Fig. 17.15 Visualization of the cumulated capital

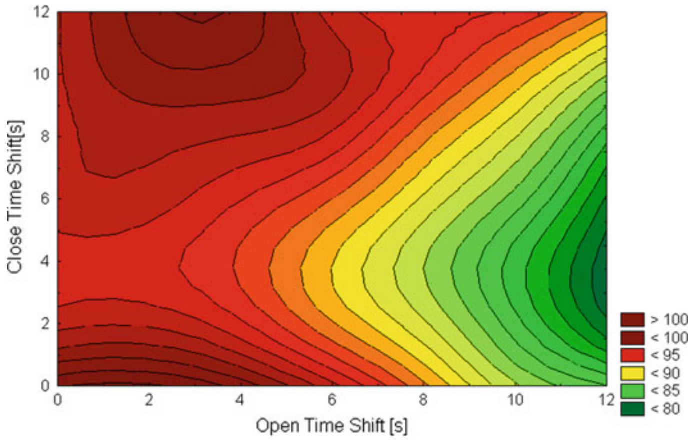


Fig. 17.16 Impact of random *open* and *close* time shift on ROI in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

in Fig. 17.16 were obtained by using the exponential distribution for disturbance generation within opening and closing positions. However, the risk analysis may be more thorough if one uses other distributions, e.g., deterministic shifts, marked as (x, y), which tell us that the opening and closing are delayed by x and y seconds, respectively. In the case of our point (4, 6), the shifts in opening and closing for all positions are equal to $4s$ and $6s$, respectively. The consequences of deterministic disturbances in opening and closing time for ROI are shown in Fig. 17.17. Notice that both the figures are different. In particular, it means that an appropriate selection of the delay distribution is an essential issue of this approach to the Monte Carlo risk assessment. Our recommendation for the beginners is to check both the above-mentioned distributions (i.e., exponential and deterministic) and then, on the basis of risk assessment using information from both the pictures, take into account the worst case scenario.

Analogously to ROI, we also examined the consequences of disturbances in the opening and closing time of positions on other game indicators, such as Volitality and Sharp ratio. The effects of these disorders (nondeterministic and deterministic variants) are shown in Figs. 17.18, 17.19, 17.20 and 17.21.

17.5 AlgoTradix: Conclusions

During the implementation of the AlgoTradix project, many very interesting results were obtained. Moreover, the team which developed the project gained a lot of experience in algorithmic trading in terms of design, construction and technological development. The good quality of results was confirmed during one and a half year

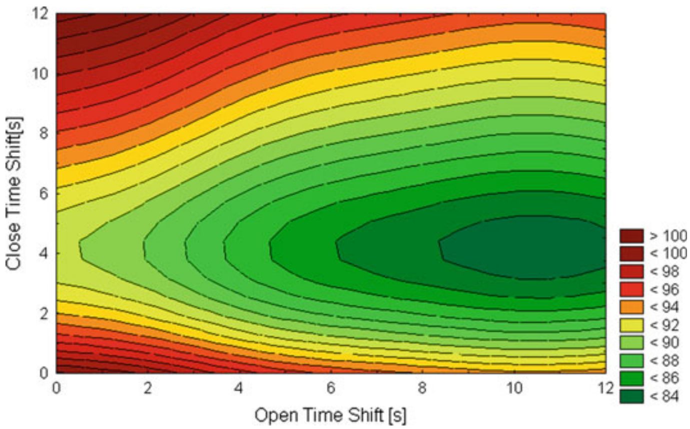


Fig. 17.17 Impact of deterministic open and close time shift on ROI in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

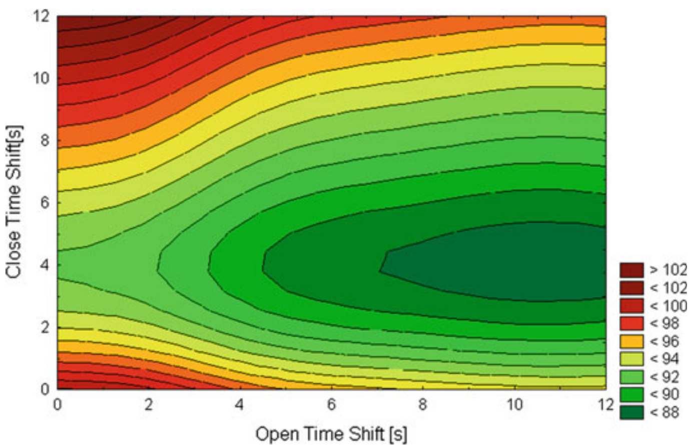


Fig. 17.18 Impact of deterministic open and close time shift on standard deviation (SD) in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

of real money trading on the Oanda platform. It is worth noting that 80% of the items out of approximately 24,000 positions were the winning positions.

A key aspect of the methodology of the AlgoTradix project was the effective management of possible shifts in algorithmic trading paradigms [290, 291] in the context of the PDCA cycle. The discovery of these shifts was possible due to the highly effective planning, implementation and evaluation of many experiments, which enabled us to gather information and made it possible to set up the priorities for paradigm shifts.

Obtaining such interesting results was possible owing to an extreme effort of a relatively small team, which was working extremely hard for 5 years. Devoting to

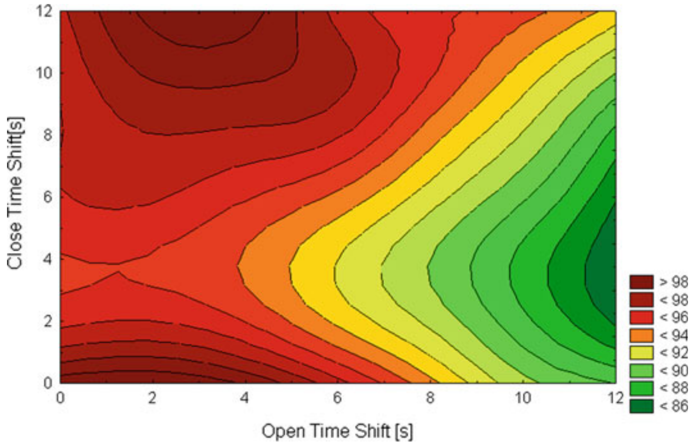


Fig. 17.19 Impact of random *open* and *close* time shift on standard deviation (SD) in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

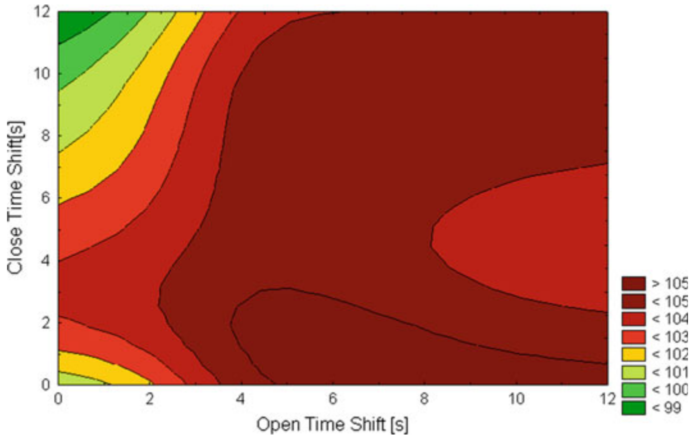


Fig. 17.20 Impact of deterministic *open* and *close* time shift on sharp ratio (SR) in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

the project over 1000 man-months is a unique experience when it comes to building algorithmic trading software and software, that can be applied in many other situations in which we deal with the multi-dimensional time series (e.g., medicine, economics). Certainly, in the real-life implementation of such projects, it is necessary to ensure a stable source of funding so that rapid changes can be introduced to the project. Under the standards enforced by the EU in this area, it is difficult to obtain the necessary flexibility in the financing of R&D from public funds.

From a technological point of view, the AlgoTradix project has broadened the experience of the team, engaged in its development, in terms of time series behavior

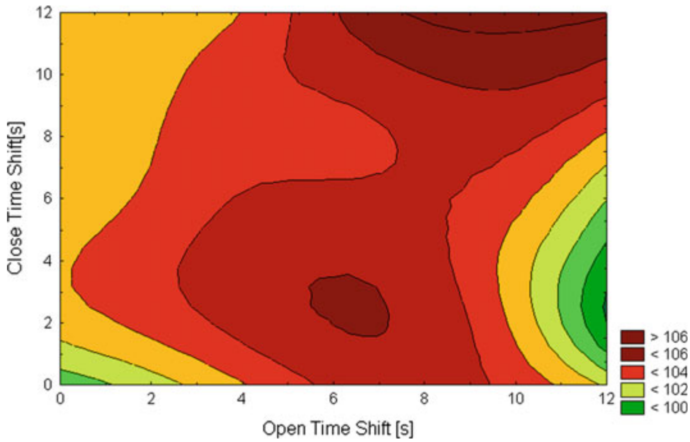


Fig. 17.21 Impact of random *open* and *close* time shift on sharp ratio (SR) in Oanda real-life rescaled trade from September 2009 till March 2011 (values in % relative to point (0, 0))

and the scope of risk management. This experience can be used in the future in the development of many interesting real-life applications in almost any area in which there is a need for analyzing and evaluating the effects of processes over time, which can be expressed in the form of multi-dimensional stochastic processes. In our opinion, a particularly interesting area of application is medicine.

It is worth enlisting the following important conclusions regarding trading in the financial market. Taking into account different types of commissions used by operators, some of the traders are doomed to defeat. If we were to assume that in the market, there exist dishonest players who possess valuable confidential information unavailable to other players, then, unfortunately, a group of players that can win is reduced to an even greater extent. In this situation, the ability to discover behavioral patterns which indicate that some players use valuable confidential information in order to take profitable positions in the future is an important feature of every trading system.

Chapter 18
Merix

18.1 Merix: Background, Genesis and Goals

At the end of the 90s of the 20th century, Zbigniew Michalewicz[1] and his son decided to start a business in North Carolina in the United States, devoted to high-tech development based on AI. Their aim was to develop such products, that could be used in a variety of industries and business areas. Their company was named NuTech Solutions.

In the mid-80s, Zbigniew Michalewicz was encouraged by the author of the book to start research on evolutionary programming [243], based on stochastic process control of computations for obtaining solutions to various problems. This concept refers to the idea of Natural Computing [420].

In January 2000, Zbigniew Michalewicz and Maciej Michalewicz invited the author of the book to collaborate with him in the establishment of an office for NuTech Solutions in Poland. The aim was to generate AI based technologies that could be further launched to the U.S. market as attractive products and solutions.

The most important projects (initiated in NuTech Solutions by the author of the book) were dedicated to two products development. The first one was the system of automatic detection of credit card frauds (the Merix project described in this section) and the second one was based on application for searching the Internet with the help of "intelligent dialogue" about available resources in the network (described later in Chap. 19 on Excavio).

After receiving pre-approvals of the projects, the author of the book, in cooperation with Andrzej Skowron and Maciej Michalewicz began to establish an office in Poland and discuss the validity of the project. Next, in 2000, a project team was created in Warsaw, consisting of approx. 50 researchers and developers. The team completed a number of projects for the U.S. and EU markets. In particular, the projects were

[1] https://en.wikipedia.org/wiki/Zbigniew_Michalewicz.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_18

implemented for such corporations in the U.S. as General Motors, Ford, Bank of America (BofA), Chevron, and in Poland BGK, PKN ORLEN, OLPP.

The business plan of NuTech Solutions assumed that the company will enter NASDAQ and then, allocate the earned capital to the further expansion of the company. Unfortunately, along with a crack of the Internet "bubble" companies in 2002, it turned out that the constraints of the NASDAQ IPO procedure for NuTech Solutions were not favorable. Therefore, after some time, NuTech Solutions was sold to Netezza, which was in turn acquired by IBM. Thus, currently, NuTech Solutions is, in some sense, a part of IBM.

The Merix project was based on the online analysis of transaction streams in a network of banks extended all around the world. When a transaction was carried out, the system had to answer the question of whether the transaction fits the pattern of behavior of the customer who is making it. NuTech Solutions proposed this technology to a number of banks. It was presented as a prototype for BofA. The prototypes of the Merix system were also presented in Phoenix, in the bank center for the analysis of complex risks involved in credit card transactions, and they were compared with the existing (in 2001–2002) technologies from HNC Financial Solutions based on neural networks.

While discussing the context and results of the Merix project, it is worth mentioning the following two issues:

1. From the point of view of analysts, there is a huge advantage of the approach used in the Merix system, in comparison with other systems that are mostly based on artificial neural networks. It stems from the fact that the Merix system is based on logical rules defining the behavioral patterns for a successful fraud identification, which are in a form that is easily comprehensible to human beings. Moreover, thanks to this feature, the rules also helped to modify security procedures for transactions with credit cards.
2. In this book, we outline some results of a very complex series of experiments. As the source materials, we have used the marketing materials of NuTech Solutions which are publicly available in various forms. The final results of experiments, obtained as part of the Merix project, may differ from the results included in those marketing materials because many experiments were conducted after the materials had been prepared. The results included in the marketing materials often have a more indicative and rough character. However, the author would like to stress that all the results reflect the magnitude of differences between Merix and the competing systems within all quality indicators that have been examined.

18.2 Merix: Examples of CSE Principles

In addition to the classic, universally applicable principles governing the design, construction and development of new technologies, such as the BPCD principle (cf. Sect. 14.3), for the development of the Merix system, it was particularly important to obtain and clean big datasets, containing both data directly related to transactions as well as contextual information. Thus, the environment for data quality management was particularly important. On this basis, experiments with the new technology were carried out.

A key assumption of the architecture implemented in the Merix project was that the project should combine various approaches, with particular emphasis on the approach that was most strongly represented by the NuTech Solutions team in Poland, i.e., a rough set approach to construction of rule-based hierarchical classifiers. These classifiers played a particularly important role in selecting important features for the construction of classifiers and identifying whether or not a given transaction is a fraud at different stages of the process. The specificity of the Merix project required its participants to pay special attention to the development of methods for inducing the classifiers, that will work on relatively small, available datasets, characterizing particular types of frauds. Issues related to performance and security were key for the success of the project.

18.3 Merix: A Brief History

At the very beginning, experts from BofA, engaged in the credit card fraud detection, were skeptical about the project as they were not convinced whether a relatively small group of professionals outside the U.S. can create an effective technology that would support automated online fraud detection. Intrigued by the idea, they agreed to carry out the tests. For this purpose, a small sample of data (with disturbed data relative to personal identification of clients) was transferred in 2001 to NuTech Solutions. The data were used by Merix for learning. Then, the system was verified on the basis of test streams (in which there was no information whether the transaction is fraudulent). The experiment ended with the great success for NuTech Solutions.

The results looked quite incredible for the experts from BofA. Therefore, it was decided to provide a larger collection of transactions to NuTech Solutions, including about a quarter million of records. The collection served as another learning material on the basis of which classifiers were induced. The classifiers induced by NuTech Solutions achieved better results on the test sample than the classifiers induced by HNC Financial Solutions (cf. Fig. 18.4).

In retrospect, it seems that the greatest technological achievement in the development of the Merix project was determining the methods for combining key features that would allow for, in a very fast way and with the high level of reliability, the evaluation of whether a given transaction is a fraud or not.

The results were presented to BofA. Unfortunately, this time, the prototypical solutions seemed to be unreliable for the experts from BofA. Therefore, the decision-making process concerning the implementation of the NuTech Solutions technology was delayed. At the same time, according to the information from BofA, HNC Financial Solutions intensified the works on their solution and managed to improve their own results. NuTech Solutions, however, could not afford developing the Merix project without the support from BofA. Efforts were made to introduce this technology to the smaller banks in the U.S. and Poland as well as in institutions using cards for different purposes. Some of these attempts were successful.

18.4 Merix: Main Results

In order to discuss the results of the Merix system, let us recall the basic concepts related with the performance measures for classifier systems, especially fraud detection systems. Most of these measures are derived from a confusion matrix (cf. Fig. 18.1).

Below, there are explanations of abbreviations used for various categories defined by the confusion matrix:

- tp—number of "Hits" or fraudulent transactions, correctly classified as frauds (True Positive);
- tn—number of "OKs" or legitimate transactions, correctly classified as clear transactions (True Negative);
- fp—number of "False Alarms" or legitimate transactions, incorrectly classified as frauds (False Positive);
- fn—number of "Misses" or fraudulent transactions, incorrectly classified as clear transactions (False Negative).

Furthermore, we define:

- $p = tp + fp$—number of all transactions classified as "Positive" (Fraud Alarms);
- $n = fn + tn$—number of all transactions classified as "Negative" (Clear—No Alarm);

		Actual	
		FRAUDULENT (af) (Positive)	LEGITIMATE (al) (Negative)
Classifier diagnosis	FRAUD ALARM (p) (Classified as Positive)	**True Positive (tp)** (Hit)	**False Positive (fp)** (False Alarm)
	CLEAR - NO ALARM (n) (Classified as Negative)	**False Negative (fn)** (Miss)	**True Negative (tn)** (OK)

Fig. 18.1 Confusion matrix

Ratio Name	**Type**	**Formula**	**Explanation**
Accuracy Rate	[%]	t/(t+f)	(all correct classifications)/(all transactions)
Error Rate	[%]	f/(t+f)	(all incorrect classifications)/(all transactions)
Alarm Precision Rate or Positive Predictive Value or Hit Rate	[%]	tp/p	(correct fraud alarms)/(all Fraud Alarms)
Alarm Efficiency Ratio (**BA's False/Positive Alarms**)	number	p/tp	(all Fraud Alarms)/(correct fraud alarms)
No Alarm Precision Rate or Negative Predictive Value	[%]	tn/n	(correct clear classifications)/(all transactions classified as Clear – No Alarm)
True Positive Rate or Sensitivity	[%]	tp/af	(correct fraud alarms)/(all fraudulent transactions)
True Negative Rate or Specificity	[%]	tn/al	(correct clear classifications)/(all legitimate transactions)
False Positive Rate	[%]	fp/al	(false fraud alarms)/(all legitimate transactions)
False Negative Rate	[%]	fn/af	(incorrect clear classifications)/(all fraudulent transactions)
Fraud Errors Rate	[%]	fp/af	(false fraud alarms)/(all fraudulent transactions)
Clear Errors Rate	[%]	fn/al	(false clear classifications)/(all legitimate transactions)
False to True Alarms Ratio	number	fp/tp	(false fraud alarms)/(correct fraud alarms)

Fig. 18.2 Merix ratios

Gini Coefficient	0≤G≤1		Derived from the scorecard of the Cumulative Gains Chart (see figure with *Cumulative Gains Chart*)
Speed	[transactions/sec]	(tn+fn+tp+fp)/time	
Adaptation time	days		

Fig. 18.3 Merix performance reports

- $t = tp + tn$—number of all correctly classified transactions (True);
- $f = fp + fn$—number of all incorrectly classified transactions (False);
- $af = tp + fn$—number of all fraudulent transactions (Actual Fraudulent);
- $al = fp + tn$—number of all legitimate transactions (Actual Legitimate).

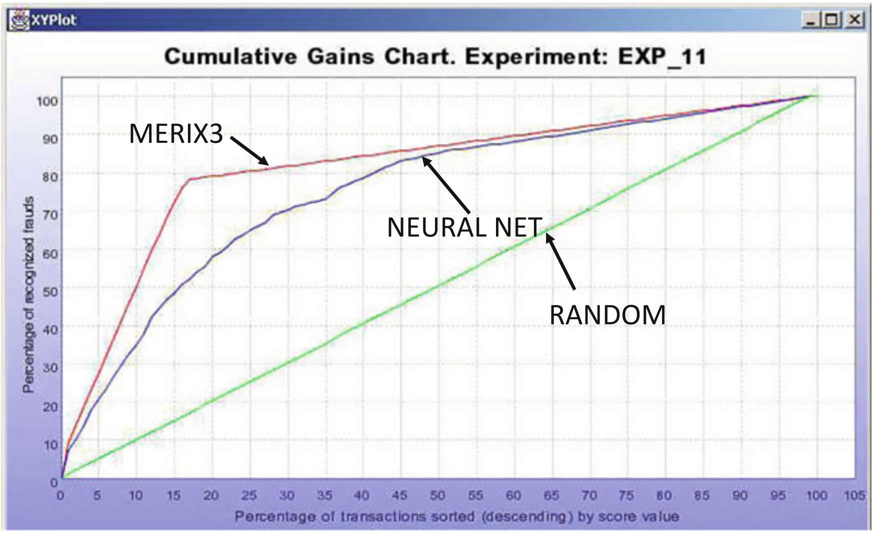


Fig. 18.4 Cumulative-gains-chart: experiment EXP_11

On the basis of the above values, one can calculate the ratios presented in Fig. 18.2.

The following values are also included in the Merix performance reports (cf. Fig. 18.3):

Quality charts (Cumulative Gains Chart, Lift Chart):

(a) Cumulative gains chart (cf. Fig. 18.4).

The y-axis in Fig. 18.4 shows the cumulative percentage of correctly classified fraudulent transactions. The x-axis shows the cumulative percentage of classified transactions, which is a fraction of all transactions in an experiment, sorted, in descending order, according to the classifier score. Point (x, y) on the lines on the chart represents $y\%$ of detected frauds after classifying $x\%$ of transactions as fraudulent. All classifiers start at the $(0, 0\%)$ point—if 0% of the transactions are classified as fraudulent, then we would recognize 0% of transactions that are actually fraudulent. Similarly, all classifiers end at the $(100, 100\%)$ point—if 100% of the transactions are classified as fraudulent, then we would recognize 100% of transactions that are actually fraudulent. Baseline: A straight line between $(0, 0)$ and $(1, 1)$ represents a classifier that correctly classifies $y\%$ of fraudulent transactions based on reviewing $y\%$ of all transactions, ranked by their classifier score. This would be a result of assigning random classifier scores to transactions. The cumulative gains chart for a given classifier reveals what percentage of fraudulent transactions we can expect to recognize from targeting a specific percentage of transactions with the highest score. For example, the cumulative gains chart in Fig. 18.4 shows that Merix correctly recognizes about 78% of fraudulent transactions by classifying about 17% of the highest scoring transactions as fraudulent. The larger the area between the base-

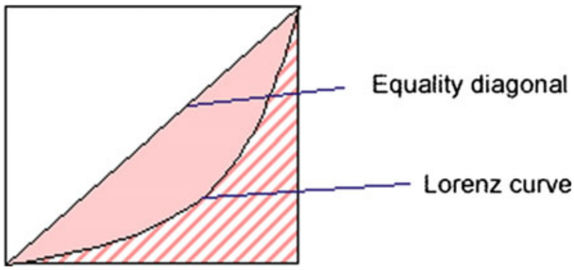


Fig. 18.5 Areas for calculation of the Gini Coefficient

line and the cumulative gains curve, the higher the quality of a classifier. Statisticians use the Gini coefficient as a measurement of this area.

(b) Gini coefficient (cf. Fig. 18.5).

The Gini coefficient is based on the Lorenz curve, a cumulative frequency curve that compares the distribution of a specific variable with the uniform distribution that represents "equality" (Fig. 18.5). This "equality" distribution is represented by a diagonal line, and the greater the deviation of the Lorenz curve from this line, the greater the "inequality." The curve can be below or above the diagonal, depending on the variable used.

When applying the Gini coefficient to the cumulative gains chart, the larger the area between the diagonal line and the Lorenz curve, the higher the quality of a classifier. The Gini Coefficient ranges from 0 to 1, 0 representing the baseline and 1 representing a perfect, error-free classifier. It corresponds to twice the area between the Lorenz curve and the diagonal (Fig. 18.5). There are different methods to calculate the Gini coefficient. We present a simple formula, proposed by Brown in 1994:

$$G = 1 - \sum_{i=0}^{k-1} (y_{i+1} + y_i) * (x_{i+1} - x_i), \tag{18.1}$$

where:

- k = number of data groups,
- y_i = cumulative percentage of recognized frauds, and
- x_i = cumulative percentage of transactions.

(c) Lift chart (cf. Fig. 18.6).

The lift on the y axis shows how much a given classifier is better than it is expected in the case of random classification.

The Merix's performance was compared between 2001–2002, with the results from the following systems:

- HNC Falcon expert system;
- HNC Falcon neural net at sensitivity set to 600 (high sensitivity);
- HNC Falcon neural net at sensitivity set to 900 (low sensitivity);

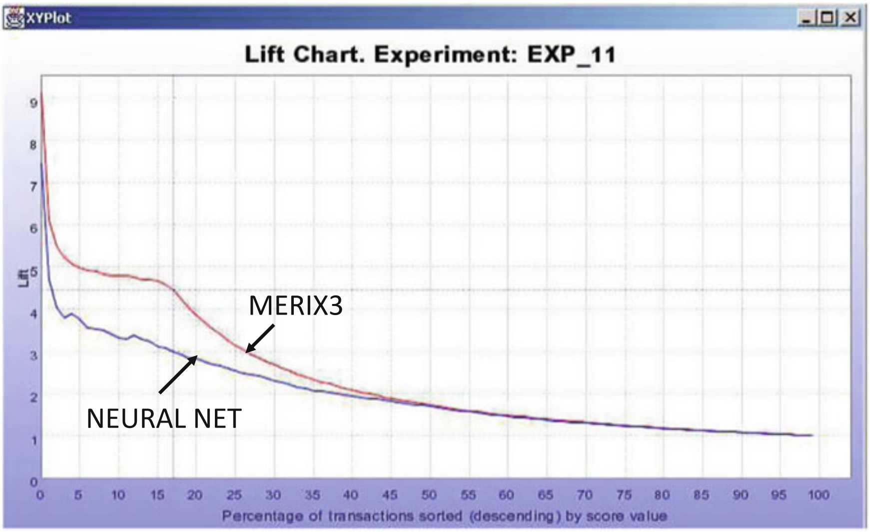


Fig. 18.6 Lift-chart: experiment EXP_11

Ratio	TSYS	FALCON 600	FALCON 900	Merix	
Accuracy Rate	76.77%	84.23%	82.97%	**84.83%**	*larger is better*
True Positive Rate	38.02%	43.49%	20.85%	23.02%	*larger is better*
True Negative Rate	86.00%	93.93%	97.76%	99.55%	*larger is better*
Alarm Precision Rate (Hit Rate)	39.26%	63.03%	68.94%	**92.36%**	*larger is better*
No Alarm Precision	85.35%	87.47%	83.84%	84.45%	*larger is better*
False Positive Rate	14.00%	6.07%	2.24%	**0.45%**	*smaller is better*
False Negative Rate	61.98%	56.51%	79.15%	76.98%	*smaller is better*

Fig. 18.7 Comparison of merix and other systems

- Total Systems Services, Inc.'s TSys;
- and combinations of the above.

These experiments proved the superiority of Merix over other systems in terms of performance. In particular, Merix effectively addresses Falcon's major weaknesses; technologies applied in Merix led to a decrease in a number of suspicious transactions and a dramatic reduction of the False Positive Rates in comparison with Falcon and other products based on neural networks. The Fig. 18.7 summarizes results of one of the experiments that are typical for Merix:

PARAMETER	TSYS	Falcon Expert	Falcon Neural Net @ 600	Falcon Neural Net @ 900	Combined Expert & Falcon @ 600	Combined Expert & Falcon @ 900	*Merix*		
							(all)	ATO	FA
Accuracy	60.0%	55.2%	58.8%	54.3%	58.9%	56.7%	85.8%	97.6%	90.1%
True Positive Rate	27.9%	14.3%	22.8%	11.0%	23.1%	17.4%	24.4%	32.4%	21.6%
False Positive Rate	6.7%	2.2%	3.6%	0.7%	3.9%	2.5%	0.4%	0.1%	.07%

Fig. 18.8 Comparison of merix and other systems (cont.)

Merix can be tuned to target specific types of frauds which increases the accuracy of the system even more, as it is demonstrated in Fig. 18.8.

By measuring the Hit Rate (Alarm Precision Rate), Merix easily outperforms competing systems, as explained below. Owing to the especially low False Positive Rate (0.4% over all types) of the system, its advantage over competition, increasing the proportion of fraudulent items in the screened population, is reduced from the artificially high levels, sometimes used for testing, to the lower levels, typically encountered in production.

The outstanding False Positive Rate of the Merix system, which is the lowest among competitors, is further improved when screening attention is restricted to ATO and/or FA, where false positive rates are 0.1 and 0.07% (seven one hundredths of one percent), respectively.

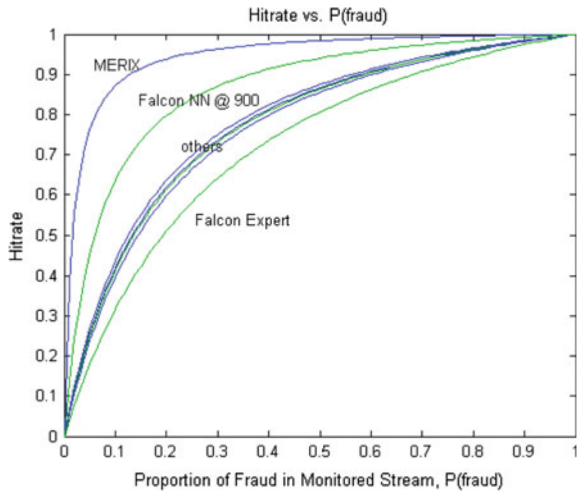
The "Hit Rate" (also known as "Alarm Precision Rate") of a detection system depends on that system's True Positive Rate, False Positive Rate, and the proportion of fraudulent items, "P(fraud)", in the stream that is being monitored.

Figure 18.9 plots Hit-Rate versus P(fraud) for the same set of systems for which values are subsequently tabulated.

Figure 18.10 gives the True and False Positive Rates for the systems under comparison. For all systems except Merix, the values given were included in the material furnished to NuTech by BofA. The values provided to Merix were obtained from the BofA transaction data. Merix classifiers were first developed on one set of such data. The values tabulated as Merix's True and False Positive Rates were then obtained, using the classifiers to screen a second, completely disjoint set of BofA transaction data.

Ratio of Hit rates for the Merix/TSYS is presented in Fig. 18.11.

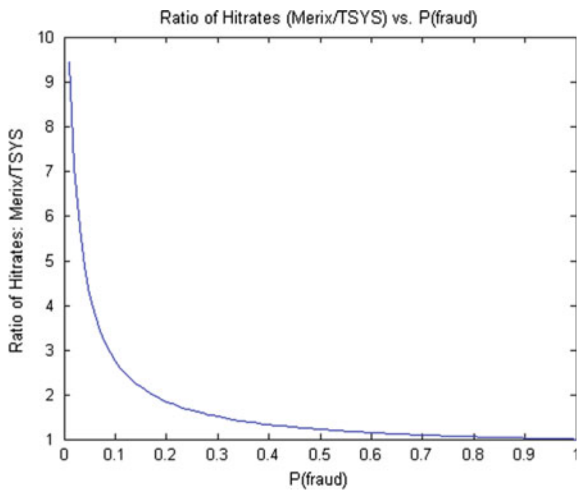
Fig. 18.9 Hit rate versus proportion of fraud in monitoring stream



Rates:	TSYS	Falcon Expert System	Falcon Neural Net @ 600	Falcon Neural Net @ 900	Combined Expert & Falcon @ 600	Combined Expert & Falcon @ 900	Merix
True Positive	27.9%	14.3%	22.8%	11.0%	23.1%	17.4%	24.4%
False Positive	6.7%	2.2%	3.6%	0.7%	3.9%	2.5%	**0.4%**

Fig. 18.10 True positive and false positive rates (BA data)

Fig. 18.11 Ratio of hitrates (Merix/TSYS) versus P (fraud)



18.5 Merix: Conclusions

It is worth mentioning that the above presented performance measures constitute just a small sample of many other performance measures, used for fraud detection systems in banks all around the world. The most important measures include time-to-decision (if it is fraud or not) or expected time to discover new types of frauds. It is important to remember that there is an ongoing war between bank officials and "black characters." Unfortunately, it is impossible to predict next moves of the "black characters." Generally, it is possible to recognize their behavior in data streams. Then, the system should be able to adapt to new patterns of behavior. Hence, the average adaptation time to new frauds is a very important performance measure. Once again, it shows the importance of appropriate integration of the three components of the Wisdom Equation (Eq. 8.1).

Chapter 19
Excavio

19.1 Excavio: Background, Genesis and Goals

The Excavio project was carried out by NuTech Solutions, whose background and history of formation was presented at the beginning of the discussion on the Merix project.

The main goal of the Excavio project was very ambitious. In short, the project aimed at developing software that could extract and structure knowledge from the documents available on the Internet. On the basis of this knowledge, the software was supposed to be able to intelligently communicate with the user to support problem solving. For the "intelligent dialog" between end-user and computer it was used simple language constructed based on relationships among knowledge domains and ontological structures representing structures of specific domains of knowledge (relevant to the current stage of the dialog). In some sense, it was a language of navigation among domains of knowledge and next navigation over ontological structures of specific domains.

For example, let us suppose that someone experiences some health problems. Then, on the basis of a dialogue with the system, he (or she) should know a lot (of what is possible) about the preliminary diagnosis of his (or her) health problems, together with an outline of possible scenarios for the medical treatment.

At the same time, it was assumed that the system would employ a very simple artificial language, relevant to the domain ontologies which would be acquired from the external sources or become automatically generated and developed on the basis of the documents available on the Internet. Of course, the problem that has been just formulated is extremely complex. As a result of the works conducted in this area between 2000–2002, NuTech Solutions came up with many implementation variants for this concept. In particular, a part of the work was published in a form of patents:

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_19

1. System and method for analysis and clustering of documents for search engine
   Inventors: Zbigniew Michalewicz, the author of the book
   U.S. Patent Classification: 715/259; 707/E17.091, 715/234
   International Patent Classification: G06F017/00 (with around 194 citations).[1]
2. Internet search engine with interactive search criteria construction
   Inventors: Zbigniew Michalewicz, the author of the book
   U.S. Patent Classification: 1/1; 707/999.003, 707/E17.092, 715/255
   International Patent Classification: G06F007/00 (with around 65 citations).[2]

This system was among others used in the Charlotte Chamber of Commerce[3] and UNCC.[4]

More specifically, the exact goal of the first phase of the Excavio project, launched at the beginning of January 2000, was as follows:

The development of a software system enabling an intelligent dialog assistance for users in searching for documents and/or text, which could in the best possible way provide relevant knowledge for the problem interactively defined by the user.

At that time, in 2000, the author of this book proposed a solution based on the following components:

1. The problem needs to be clarifies using an automatic dialogue between the system and the user:
   The dialog should be easy to use and possibly analogous to the exploration of a virtual three-dimensional space. It was assumed that the dialogue should be guided by:

   a. domain ontologies based on the domain knowledge, introduced to the system by some experts, and automatically generated ontologies acquired from the Internet and local network resources, which should be further integrated during an interactive collaboration with domain experts,
   b. specific ontologies for particular groups of users, taking into account their level of knowledge, conceptual apparatus that is used and personal profile; these ontologies are supposed to be interactively refined for specific groups of users,
   c. ontologies automatically generated by Excavio, integrating both ontologies from (b)–(c) and taking into account new concepts, emerging on the Internet and local network; by assumption, these ontologies should be verified by domain experts.

[1] http://www.google.com/patents/US20020065857.

[2] http://www.google.com/patents/US20020042789.

[3] http://charlottechamber.com/.

[4] http://www.uncc.edu/.

2. The best possible quality documents needs to be acquired from the Internet and other sources, using societies of intelligent software robots:
   These documents should be selected for each potential class of problems, encountered by potential users.
3. Automatic analysis and assignment of documents and/or important parts of texts to ontological structures for the construction of data structures, supporting the dialogue with the user is important.
4. Automatic summaries of relevant documents and their contents from different ontological perspectives and classes of users' concerns, is relevant as well.
5. Extraction of predefined semantic issues inherent in the documents acquired from the Internet (e.g., by matching offers for sale and purchase, mining prices of products and services for a given class of global products, analyzing emerging risk patterns for different companies as well as different aspects of the economic growth) is essential.

19.2 Excavio: Examples of CSE Principles

For Exavio project can also be applied appropriately adjusted universal CSE principles based on BPCD principle (as we have discussed in previous chapters and in the Sect. 14.3). However, when it comes to developing the Fundamental Adaptive Principles of CSE for the Excavio architecture, one must become aware of the difficulty and complexity of the project that had been undertaken. Roughly speaking, in 2000, the technical goal of the project could be intuitively interpreted as follows:

To design and build a software system, simulating the behavior of an intelligent assistant for various fields of applications.

In the Excavio project, great attention was paid to some specializations of intelligent assistants, such as providing medical or legal consultations, searching for entities counterfeiting branded products as well as conducting automated analysis of companies or emerging trends.

In 1950, Turing [510] defined a metaphorical test for a software system to be treated as an intelligent system. According to this criterion, a given system is intelligent if relevant experts cannot recognize whether they are talking with a system or with a human being. Actually, since then, designing and building systems that imitate the behaviour of an intelligent assistant can be considered as one of the most important technology related to the development of AI. There are some attempts to modify the Turing test (e.g., in the form of physically embodied Turing test [375]).

It should be noted that in the past, many other attempts were undertaken to build intelligent decision support systems for different areas of application. For example, in the 70s of the 20th century, Stanford University developed an expert system called MYCIN. In designing the MYCIN system, AI techniques were used to identify bacteria causing severe infections such as bacteremia or meningitis and to recommend antibiotics with dosage adjustments for the patients, depending on their body weight.

The name MYCIN was derived from the antibiotics themselves, as many antibiotics have the suffix "mycin".[5] The MYCIN system was also used to diagnose blood clotting diseases. MYCIN has never been actually used in practice, but research indicated that it proposed an acceptable mode of treatment in about 69% of all cases, which was better than the performance of the infectious disease experts, who were judged using the same criteria. In the simplest words, the system operated on the basis of knowledge representation, including diagnostic knowledge, therapeutic knowledge and knowledge on drugs in the form of about 600 rules. The acquisition of knowledge in the form of these rules was very costly. It took about 6 years of work of the distinguished experts from the field of medicine as well as other experts, who in the next decades were called as knowledge engineer. The basic obstacle for the MYCIN system was the *problem of escaping brittleness*. The name of this problem was introduced 10 years later by John Holland [209]. The essence of the problem lies in the fact that with the passage of time, changes occur in the way bacteria respond to antibiotics; thus new antibiotics are being discovered and furthermore, the general state of medical knowledge, explaining mechanisms of diagnosis and treatment of particular diseases, keeps evolving. In other words, it is only possible to define certain rules reflecting the actual state of knowledge (in a particular period of time). However, it is very difficult to update this knowledge to the above-mentioned changes. A solution was proposed by John Holland in the form of a classifier system. This approach meet some of the expectations. However, over the following decades, a number of alternative approaches to "escaping brittleness of the MYCIN system" were proposed. When in 2000, the author of this book started working on the Excavio system architecture, he was fully aware of the complexity of the problem. Besides, they felt that the approach to architectural decisions that would enable effective searching for a solution to this problem was the key to the success of the project.

Therefore, the most important architectural decisions in the Excavio system were oriented towards the design and execution of a series of experiments that could cast some light on the principles that should be applied during the development of the project to cope with the problem of the "brittleness of the MYCIN system." The approach that has been proposed for solving the problem was based on the hierarchical structures of the Pawlak information systems and the creation of ontological structures based on rough mereology [402]. This approach was based on the previous experience of Andrzej Skowron that he gained in a number of similar projects, especially in the project related to the concept of risk in traffic, assessed from the point of view of an unmanned vehicle (UAV), discussed in the following sections of this book. The main idea, being under development at that time and developed further in a number of papers, e.g., [27, 363], was based on the application of hierarchical structures of the Pawlak information systems for the approximation of ontology relevant to the concept of traffic risk. In order to clarify the nature of this approach in solving the problem of "escaping brittleness of the MYCIN system," let us imagine that instead of introducing new rules to a system by a team of experts, we decide to build a system in which we introduce new examples/cases (characterized by

[5] http://en.wikipedia.org/wiki/Mycin.

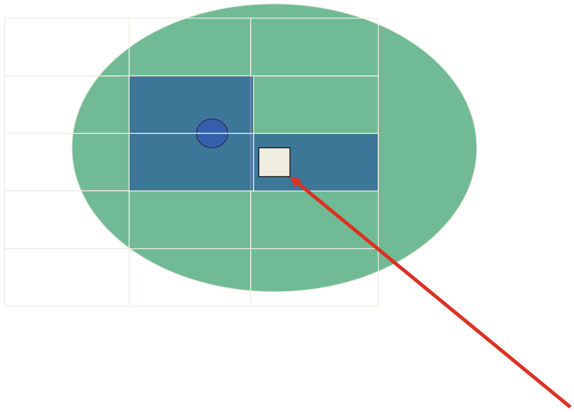


Fig. 19.1 Lower approximation (*blue color*) of a particular bacterial infection relative to an information system (boxes represent the indiscernibility classes of the system); *red arrow* is pointing to a particular case (*white color*)

features or attributes related to diseases of patients, bacterial behaviors, properties of new drugs, diagnostic capabilities of new instruments, therapeutic procedures) and possibly some new general principles for the inference of medical knowledge.

Note that a given system, after receiving new examples, should automatically update the rules used for an interactive dialogue with experts in order to ensure and verify the quality of their knowledge. Then, these rules could be used to control the dialogue with the patient or the doctor, aiming at obtaining the support related to the latest medical knowledge.

In other words, in the computer system, knowledge is represented in the form of the Pawlak information systems in which different classes of objects represent different actual cases (such as diseases of patients, bacterial behaviors, properties of new drugs, diagnostic capabilities of new instruments, therapeutic procedures) which are described by attributes conveying various aspects of the represented cases. It is well known that rough set-based methods can automatically generate rules [360, 362, 389, 442, 449, 450] related to these cases for each information system separately. However, the proposed approach is based on the representation of dependencies between the hierarchies of these information systems that allow generating knowledge from specific cases for addressing more general situations. However, the approach should take care of the relationships between different aspects of iterative processes of diagnosis (in the form of dialogue), supporting the treatment and monitoring the progress taking into account its effectiveness.

In order to illustrate the main concept behind the architectural decisions aimed at solving the problem of "escaping brittleness of the MYCIN system," let us assume that we have an information system describing the symptoms of a variety of bacterial infections, using a set of attributes. This system defines a partition of objects (patients) as visualized in Fig. 19.1, which is then used to define the lower approximation and the upper approximation of a disease related with a bacterial infection.

In Fig. 19.1, the blue color marks the lower approximation of a specific bacterial infection. We are analyzing a particular case that is represented as a white box

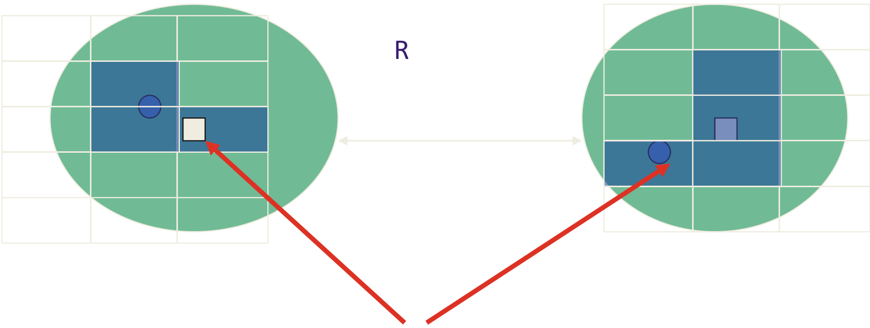


Fig. 19.2 Illustration of changes in the lower approximation (*blue color*) of a particular bacterial infection (relative to a given information system) before and after using a given R treatment and changes in the location of the cases in approximations

(indicated by the arrow). At the same time, let us assume that we have a second information system, representing the effects of a specific treatment (marked by R). In this way, a relationship between the cases before and after the administration of the treatment R is shown in Fig. 19.2. In particular, one may observe a change in the lower approximation of the concept, related to the infection, as well as changes in the location of objects, relative to the approximations of the analyzed cases.

At the same time, we can also consider another Pawlak's information system, in which objects are pairs (or more generally paths) representing the successive states of the treatment R. In this new information system, we can also add a number of new attributes, determining the effectiveness of the treatment R or different risks associated with it. This creates a simple hierarchical system, shown in Fig. 19.3.

Of course, each of these systems can automatically generate different rules of inference which lead to the formation of simple dialog patterns, associated with the potential effects of administering the treatment R in cases that are sufficiently close in nature to the cases that has been already identified and examined in the information system in Fig. 19.1. It should be also noted that in medicine, as well as in other areas of complex IT applications, often we encounter possibilities to use a different potential relationships of concepts (e.g., action scenario, procedures and medical treatments) instead of just one. In other words, we can imagine that we have a lot of such schemes as shown in Fig. 19.4 and, these schemes are compared and evaluated from the point of view of the higher levels of the hierarchy of information systems. These levels of hierarchy arise naturally on the basis of the ontology of the concepts related to the integration of more in-depth knowledge. Ontological structures are specific forms of representation of important fragments of knowledge in many specific areas of application. It should be noted that these structures are generally subject to modification over time. Therefore, in the development of dialog systems and inference systems based on this type of hierarchies, reflecting the ontology of the domain, the architecture should be flexible enough to ensure the gradual evolution of such structures. Returning to our example, one can intuitively imagine the resulting

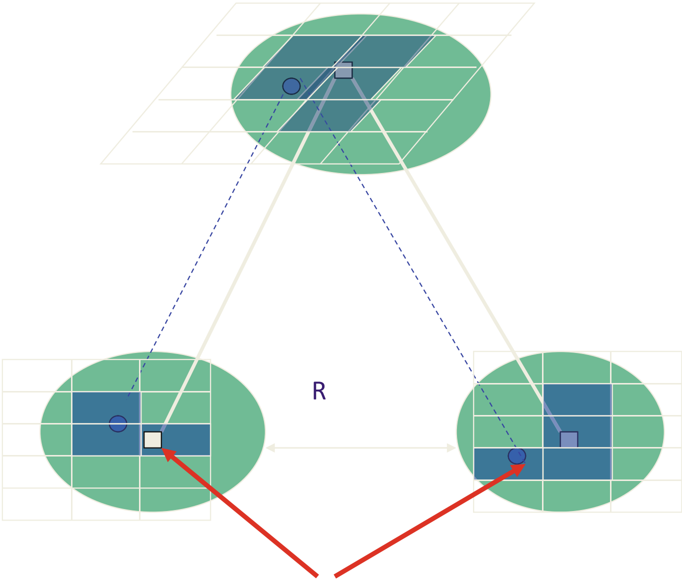


Fig. 19.3 A simple hierarchical system

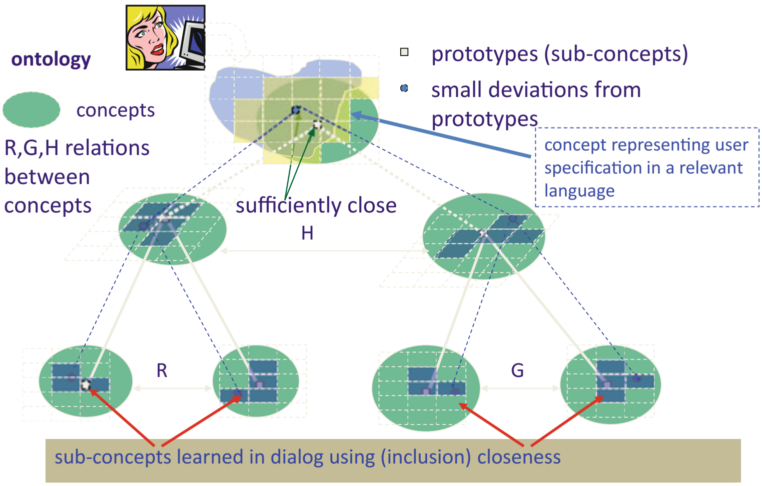


Fig. 19.4 An example of the aggregation scheme for different aspects of diseases

structure, presented in Fig. 19.4, where we have two treatments: R and G. Moreover, the top information system describes the attributes, on the basis of which one can compare the efficiency of both treatments.

The scheme in Fig. 19.4 is an example of the integration of many aspects of knowledge about a given disease. In this case, the scheme enables us to match a

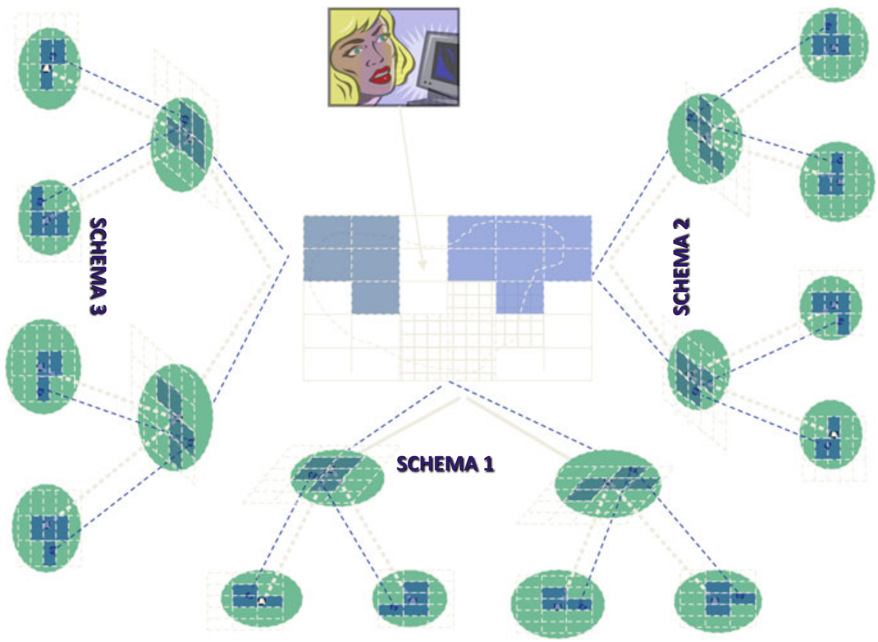


Fig. 19.5 An illustration of the compound aggregation scheme

new case as close as possible to the case that has already appeared in one of the information systems, and then find the optimal treatment for this specific case. In rough set theory, this type of integration mechanism between specific knowledge, existing knowledge and much more general reasoning is often represented by the so-called schemes (cf. e.g., schemes of approximate reasoning in [27, 455]). In fact, the ontological structures used for generating schemes can be much more compound, and can form packages of schemes that are illustrated in Fig. 19.5.

In the Excavio project, particularly important information systems are based on cases which are parts of texts, documents or groups of documents. The effectiveness of this type of analysis is mostly dependent on the choice of relevant attributes. In this case, the attributes are generated by many different ways. The approaches used in the Excavio project include, among others, the following ones:

1. *Vector Space Model*—based on the ideas of Gerard Salton from the SMART system (System for the Mechanical Analysis and Retrieval of Text) developed in the 60s [426].
2. *Frames*—extracting semantic "scenes" from the text, which constitutes a modification of the approach proposed by Marvin Minsky, based on the analysis of syntactic structure of sentences and text and transforming it into "scenes."

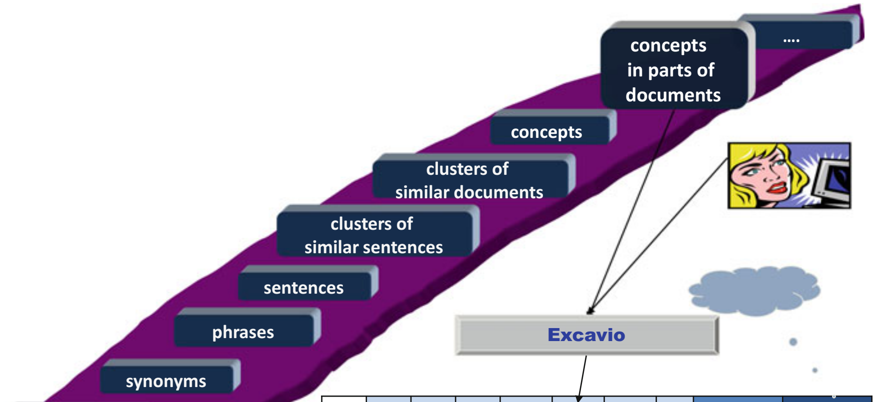
3. *Rough Mereology*—the approach used for clarifying specifications of user requirements, defining the need for knowledge contained in a fragment of a text and the method of constructing solutions from parts, which are groups of documents (linked by domain ontology acquired from experts or automatically generated for the needs specified by users). This method is proved to be largely successful in identifying significant fragments of texts related to the user's query. This method is based on the ideas contained in [402] which were further developed in a number of papers (e.g., [401, 403] and references in these works).
4. *Stone Semantic Spaces*—a term used in the Excavio project as a paraphrase of the Stone representation theorem of Boolean algebras. Different applications of this theorem in the context of various aspects of the relationship between logic and topological spaces were of particular interest for the Polish School of Logic [123, 410]. Intuitively speaking, any theory can be represented by a topological space which is a Stone representation of the Lindenbaum—Tarski algebra. For example, the Cantor set is a topological space for classical propositional logic. For the analysis of texts in the Excavio project, involving the grouping of ”close” documents, attempts to design the hierarchical structures of ontological concepts were undertaken. The essence of this approach was understanding the concept of “closeness” as a form of semantic similarity between documents. In general, for this purpose two approaches were used:

   a. metrics based on the ideas of the Gerard Salton group [426];
   b. similarity relations based on different forms of tolerance relations [267, 454].

The approach based on the Stone semantic spaces appears to be particularly promising when it comes to the future prospects for the development of architectures dedicated to dialog systems. Stone's theorem tells us about the duality between the world of processing symbols (syntax, deduction) and the world of modifying models (semantics, induction). Of course, it is very difficult to capture all aspects underlying the interaction between deductive thinking and inductive thinking (intuitively, we are talking about the coupling between the left and right brain hemispheres from the point of view of efficient processing of simplified semantic aspects, derived from large collections of documents). This is evidenced by the works conducted as part of [358], which is a continuation of the Excavio project. Its aim is to boost the efficiency of the thematic grouping of documents and increase search relevance.

To illustrate the process of document structuring, one can imagine that attributes are assigned according to some ontology of the attribute generation. Such an ontology can reflect the degree of details and complexity of computations, leading to the determination of attribute values on successive levels of their aggregational complexity. This process is illustrated in Fig. 19.6.

Architectural concepts that has been outlined above provided the basis for the creation of algorithms and data structures in the Excavio project, aimed at controlling a simple dialogue between the system and the user. They were planned to be used in the Excavio system as illustrated in Fig. 19.7.



	w_1	w_2	...	ph_1	ph_2	ph_3	...	approx. concept	target concept
D_1	7	1	...	8			...	+	+
D_2	1	3	...	4			...	+	-
D_3	4	2	...	6			...	+	+
...	...	...							-

Fig. 19.6 An illustration for attribute generation in the Excavio project

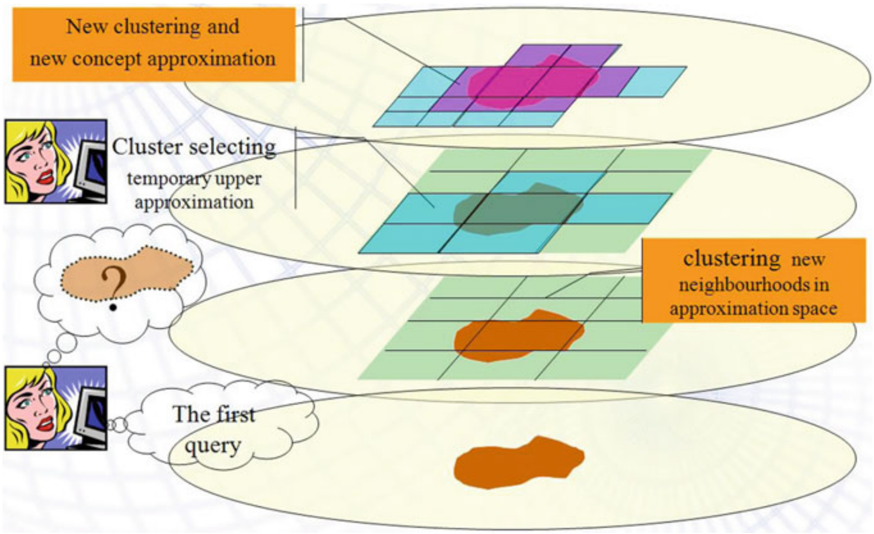


Fig. 19.7 The basis for dialog with users in the Excavio project

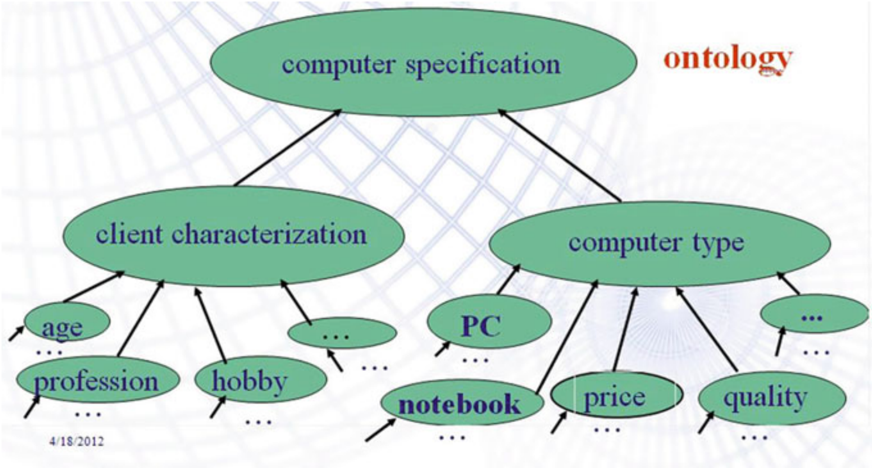


Fig. 19.8 The ontology framework for dialog with user

The lower part of Fig. 19.7 shows an user who asks the first query. The query corresponds to a subset of documents. The information system determines the “first level approximation” for the set of documents that are to be searched for. Then, in an iterative manner, through a simple dialogue with the user and available attributes, the system is trying to improve upper and lower approximations, clarifying where the documents searched for by the user are. This process proved to be very helpful for the grouping of documents along the domain ontology, to which the query is related. There were also carried out various experiments for the online generation of ontologies. However, the technology available at the beginning of the 21st century was not efficient enough for the online grouping of millions of documents which was necessary for practical applications. Hence, the development was limited to the documents grouped in the offline mode.

In order to explain the principles of dialogue control, let us imagine that you are asking the following question: *I would like to buy a cheap computer for my son*. Then, the system should capture the phrases *would like to buy* and *computer*. Next, we assume that the system equipped with the domain knowledge, encoded in the form of ontological structures and rules of inference, should know the details concerning the specifications of the computer that you want to buy and show all offers, sorted by the criteria that best match your needs. Figure 19.8 presents the ontology framework for the dialog control.

Undoubtedly, the most difficult aspect of the Excavio project was to develop architectural variants of potential algorithms and data structures for a “semantic” dialog with the user. Of course, in addition to this important aspect, numerous other architectural decisions were important for the project. The Excavio general software architecture was usually presented as it is shown in Fig. 19.9.

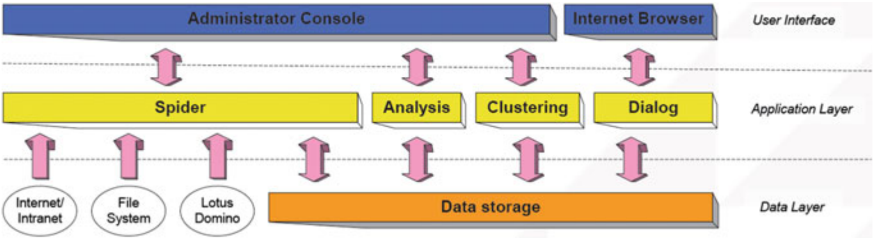


Fig. 19.9 Excavio software architecture

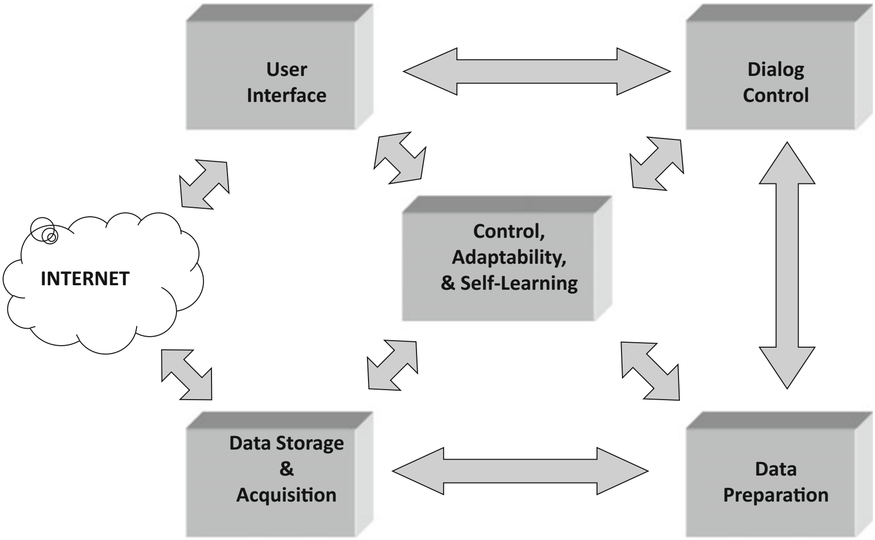


Fig. 19.10 Excavio software architecture (cont.)

This architecture is based on a multi-level software architecture, strongly advocated by large corporations such as Microsoft, Oracle, IBM and others. A slightly more subtle look at the architecture of the Excavio system is presented in Fig. 19.10.

In the central part of the diagram in Fig. 19.10, a special role of the module responsible for learning, adaptability and control is emphasized. Basically, this module integrates all other functionalities of the system. Since the diagram shown in Fig. 19.10 is very general, Fig. 19.11 presents a further refinement of the scheme.

This scheme was used in the administrator control panel, controlling all processed data in different modules. A schematic representation of the panel is shown in Fig. 19.12.

Of course, the user interface looked different. Sample visualization of the user interface is shown in Fig. 19.13.

It should be noted that the Excavio system was built in parallel with the Merix system described above. Both systems were built by the same team and the general

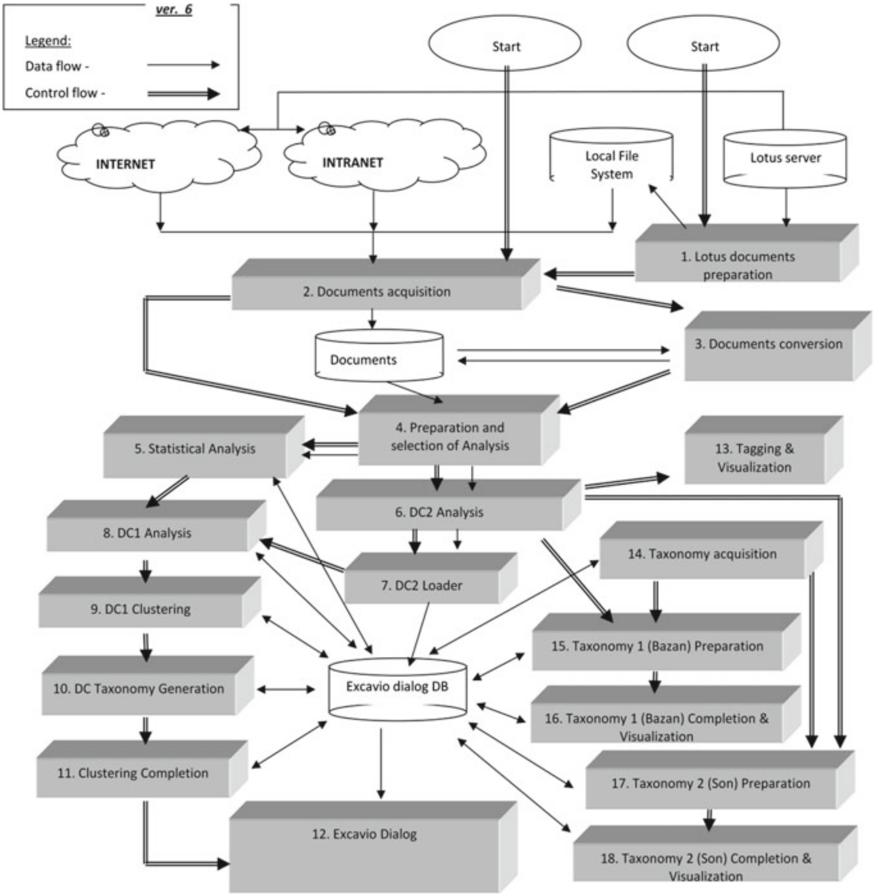


Fig. 19.11 Excavio software architecture (cont.)

principles of architecture were basically very similar. Consistent planning, design and development of a system, based on the research cycle in accordance with the Deming PDCA cycle, was of utmost importance. In other words, testing different approaches towards the design of a system, based on the principle of change management paradigm, is a key to any project related to the development of AI technology, especially dialogue systems. However, in the case of each of the projects, specific properties of these systems were also taken into consideration. In addition, some other elements of the Excavio architecture were published online and are available in Internet.[6]

---

[6] http://www.google.com/patents/US20020065857, http://www.google.com/patents/US20020042789.

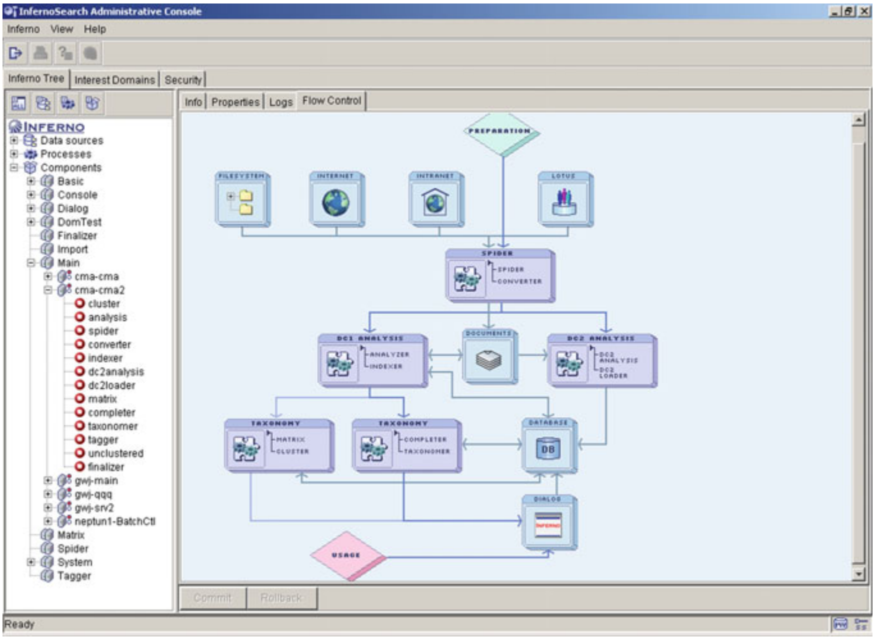


Fig. 19.12 Excavio administrator control panel

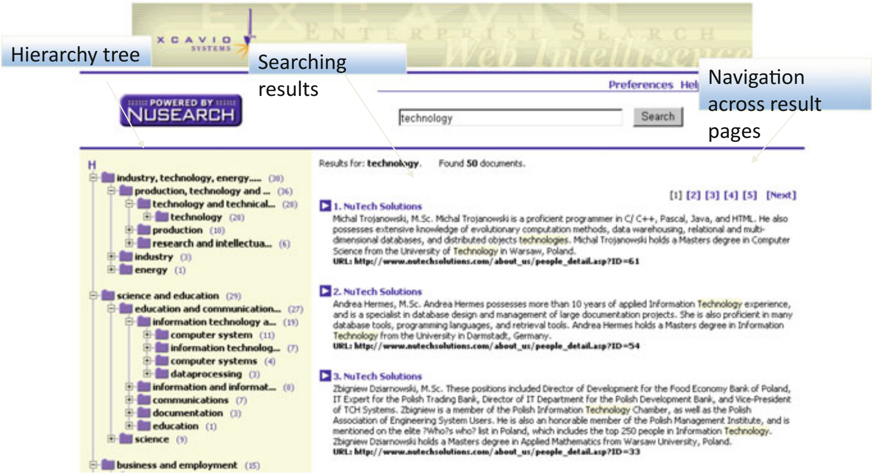


Fig. 19.13 Example of Excavio user interface

19.3 Excavio: A Brief History

Initially, the system was designed as the Excavio core for generating domain systems, whose functionality was limited to intelligent assistance in areas such as medicine, law, economics and education. It soon turned out that the complexity and costs of building a generic system are too great for a young company, pursuing at the same time several other parallel projects. Therefore, it was decided to focus on the needs of specific customers. The system was implemented in two languages—English and Polish. In Poland, a module that can automatically generate summaries of documents was especially popular. This concerned summarizing legal and economic documents for which there was a high demand from the MPs and bankers involved in the process of integration with the European Union. In this way, the prototype solutions were delivered to the Polish Parliament and the Polish National Bank. In the U.S., the Excavio system has been implemented as a search engine for documents in the Charlotte Chamber of Commerce and in the UNCC. At the same time, numerous interviews were conducted with customers in the banking and energy sectors. However, in this case, they were limited to the presentation of prototype solutions.

19.4 Excavio: Main Results

During the development of the Excavio project, the following interesting features were implemented:

1. Automatic semantic grouping on the basis of documents provided by ontology experts.
2. Automatic generation of ontological structures based on the given set of documents.
3. Searching the Internet for purchase and sale offers and matching them together.
4. Searching the Internet for offers with prices significantly deviating from the selected intervals.
5. Automatic summarization of documents in Polish and English.
6. Supporting dialogue with the user on the basis of simple ontological structures acquired from domain experts.

This all was achieved by a relatively small team between 2000–2002, with very modest resources allocated to the project. At the same time, during the development of the project, numerous interesting ideas, that in retrospect seem to be still attractive, were tested and verified. The ideas are cited in the patents developed by the research and development centers of the largest corporations in the world involved in the development of AI (such as: IBM, Google, Microsoft, Yahoo!, Hewlett-Packard, Oracle, Saor Kabushiki Kaisha, Matsushita Electric Industrial, MusicMatch, NEC, Endeca Technologies, Fuji, Xerox, Canon, Boeing, Hansen Medical, Sony, Accenture, and Red Hut).

19.5 Excavio: Conclusions

It should be noted that during the implementation of the Excavio system, a number of different approaches were tested and, in retrospect, it seems that there are still many ideas, put forward by the author of this book in the course of the project, that are worth of further research and scientific work. This includes unfinished works on the application of rough mereology in ontology engineering and data structure control for handling the dialogue with the user, using the tolerance relation approach [454]. This approach to extracting semantic structures from texts was initiated in Japan [267]. In addition, over the course of many years which have passed since the implementation of the Excavio project, a number of new technologies were developed, providing much more subtle and more effective approaches to many problems than the ones that had been adopted during the Excavio project.

Chapter 20
Unmanned Aerial Vehicle (UAV)

20.1 UAV: Background, Genesis and Goals

Between 2000 and 2005, a group of scientists from Warsaw, including professors Andrzej Skowron, Witold Łukaszewicz and Andrzej Szałas, worked closely with a group from the Linkoping University, led by Professor Patrick Doherty, on the project called WITAS[1] [117, 538]. WITAS is an abbreviation for "Wallenberg Laboratory for Information Technology and Autonomous Systems." It was a research project carried out from 1997 to 2005, dedicated to the design of intelligent autonomous vehicles and other autonomous systems. In particular, the project focused on the development of an airborne computer system, capable of making rational decisions about the continued operation of the aircraft on the basis of various sources of knowledge, including pre-stored geographical knowledge, knowledge obtained from visual sensors and knowledge obtained through data link. Figure 20.1 presents the WITAS RMAX helicopter in an urban environment.

A group of scientists from Warsaw, together with the group of Professor Doherty from the Linköping University, was interested, among others, in research on the development and application of AI methods for monitoring the level of safety on public roads, identifying risk factors for road incidents and/or traffic control reporting of these events. Achieving these goals (related to the security of vehicles driving on the road) was one of the assumptions of the WITAS sub-project. The case study described in this book under the name of UAV consists of some conclusions drawn from the aforesaid sub-project. The results of the UAV project have been published in [117] as well as in other papers cited in this book and in [33]. Figure 20.2 presents a picture of the test area that was also used for data collection during the experimentation with the WITAS/UAV helicopter.

[1] www.ida.liu.se/ext/Witas/index.html.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_20

Fig. 20.1 WITAS RMAX helicopter in an urban environment [117]

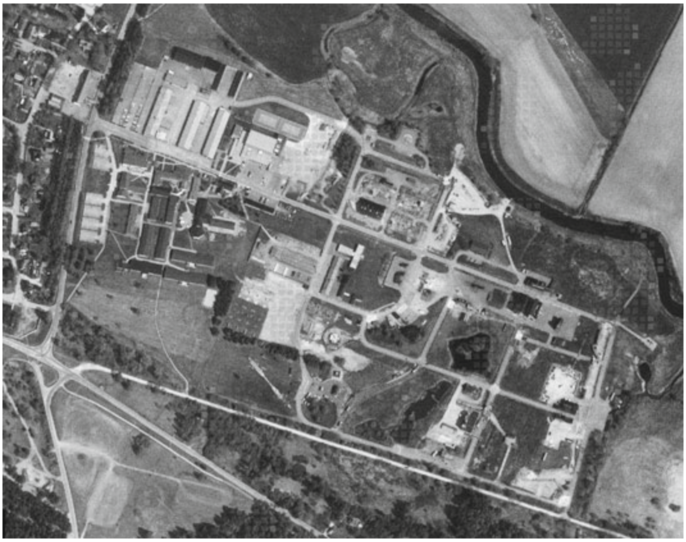


Fig. 20.2 Revinge emergency training area in southern Sweden [117]

20.2 UAV: Examples of CSE Principles

The design and construction of unmanned aerial vehicles are very demanding and compound tasks, greatly exceeding the scope of the UAV project described above. In our case, we limit ourselves only to the architectural aspects of the analysis and processing of data, collected by the helicopter sensors to identify the level of risk on

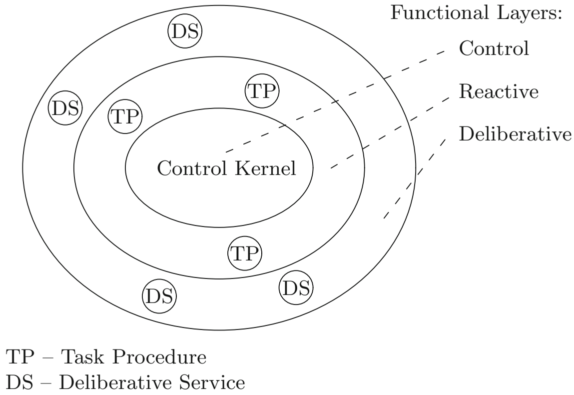


Fig. 20.3 Functional structure of the architecture [117]

the road. In the simplest words, the functional structure of the system architecture can be understood as shown in Fig. 20.3.

Deliberative Services consume a significant part of the processor time. They include a path planner, a task procedure execution module and other components. Other important subsystems are of course a Geographic Information System and an Approximate Deductive Database System.

Given the architecture of technological solutions dedicated to the approximate deductive reasoning, there are two architecture decisions which are especially important:

1. The use of hierarchical approximate inference to identify the level of risk on the road.
2. The combination of non-monotonic reasoning with the rough set technology.

In the book, we would like to present some basic ideas for hierarchical approximate reasoning that were developed during the implementation of the UAV project. As the WITAS project ended, the ideas have been further significantly extended and successfully implemented in several real-life projects, e.g., [27, 32, 363, 378, 451].

The essence of the use of hierarchical approximate reasoning boils down to the fact that the analysis of the situation, in the way we see it, starts from sensory data. Next, the data are accumulated in groups that define the characteristics of the higher levels, such as the distance between groups of vehicles (on the road that is being monitored) or a proper aggregation of weather aspects (e.g., temperature, humidity, visibility), to determine whether or not the situation on the road is safe. In other words, the concept of risk is decomposed into a hierarchy of concepts, simpler components that make up a sort of a skeleton for the ontology, representing inferences about the level of risk. An example of such an ontology is presented in Fig. 20.4.

Of course, in order to use the sample ontology, which is shown in Fig. 20.4, you must first have a set of inference rules at every level. Examples of such rules are depicted in Fig. 20.5.

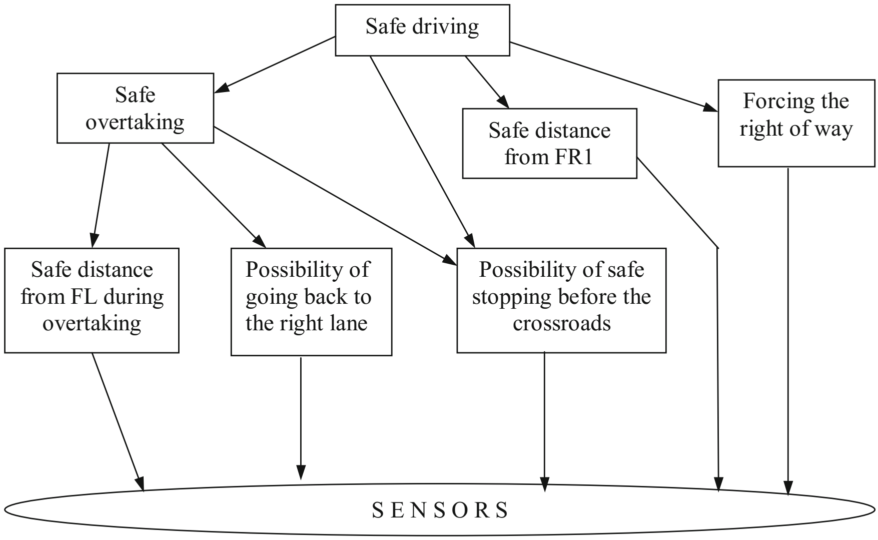


Fig. 20.4 The relationship diagram for presented concepts [33]

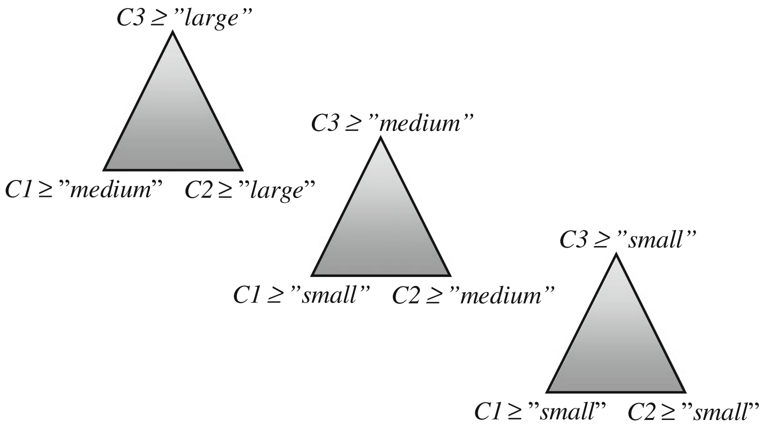


Fig. 20.5 An example of production as a collection of three production rules [33]

The rules presented in Fig. 20.5 themselves (i.e., each treated separately) describe only the selected aspects that are aggregated at the global level to assess the level of risk in the changing situation on the road. The practical use of this type of policy is possible only after a skilful synthesis is performed. Examples of such a synthesis of approximate reasoning schemes (AR-scheme) are presented in Fig. 20.6.

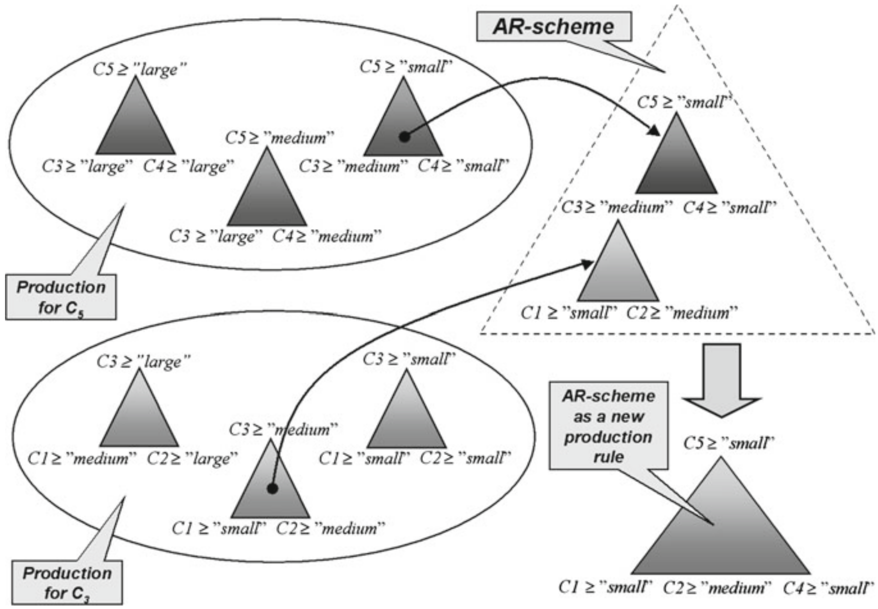


Fig. 20.6 Synthesis of approximate reasoning schemes [33]

The idea of using a domain ontology was further developed by the group of Andrzej Skowron, which devised methods for ontology approximation that were successfully applied in different areas, e.g., [27, 32, 363, 378, 451].

When it comes to the practical applications of non-monotonic reasoning, there is a very interesting combination of these paradigms with the rough set approach, which has been developed as part of the UAV project. In the simplest words, it comes down to the fact that the world of objects in the information system is treated as a base for default reasoning. For example, to paraphrase Reiter and his motivation to introduce non-monotonic logic [411, 412], if we have an information system in which the objects are only birds that can fly (or in a more general case, if these are the objects from the upper approximation [117] of the concept "can fly"), then on that basis, we can use the default rule which says that "all birds fly." However, when counterexamples to this rule began to appear in this information system, for example an object representing a penguin who cannot fly, then, the rule is no longer valid. In other words, as long as we do not receive contradictory examples in a given information system, we can apply the principle of "closed world assumption", using inductive reasoning based on the already known examples. However for new and so far unseen examples, one may find that the previously applicable rules may lose their power.

20.3 UAV: A Brief History

The UAV design history is shown on the project website[2] as well as in numerous studies (e.g., [117] and references in this book).

20.4 UAV: Main Results

According to Andrzej Skowron, the most important result of the project was the experimental confirmation of the decision related to architecture adopting the approaches for construction of algorithms, supporting complex concept approximations. The validity of the approaches, including hierarchical approach to approximate inference scheme using synthesis (Fig. 20.6) and the combination of non-monotonic reasoning with the rough set approach, was positively confirmed in many experiments for both the UAV project and within a broader area of application.

20.5 UAV: Conclusions

The ideas presented above, concerning complex hierarchical approximate reasoning about vague concepts, seem to be ones of the most important conclusions drawn from the UAV project. These results were further developed and used in the construction of technology for AI-related projects discussed in this book. It should be noted that the scope of research projects related to the recognition of dangerous situations on the road by unmanned vehicles is much wider than just the issues related to the techniques for the application of rough sets in hierarchical approximate reasoning for assessing the level of risk on the road. However, in a much broader sense, one can say that the approximation of complex vague concepts (such as the concept of risk) is one of the key challenges for many projects related to the control of autonomous unmanned vehicles as well as for most of the projects related to the application of AI, including, in particular, the projects described in other case studies (i.e., Excavio, Merix, and AlgoTradix).

[2] http://www.ida.liu.se/ext/witas/index.html.

Chapter 21
Conclusions: Toward Science-Friendly Languages for Interaction Rules & CSE Principles

21.1 Introduction to Some Conclusions from Case Studies

In this chapter, we present selected conclusions, drawn from the projects described in Part IV of the book (Case Studies), as well as many other projects.

The starting point of the conclusions presented in this chapter will be to try to deepen the understanding of common benefits (synergies) which stem from the analogy between the implementation of CSE projects and computations in IGrC. These synergies appear as a result of an observation that **planning, designing, and implementing a complex project may be seen as discovering, learning, processing (including communicating), and developing concepts that are necessary for learning of interaction rules, in particular related to the project's implementation**.

At the same time, the conclusions presented in this chapter are a continuation of the remarks made in Sect. 14.4. A variety of some further interesting relations between CSE and IGrC are presented in the following chapters of the book, especially in Chap. 31.

As part of the conclusions included in this chapter, we put a particular emphasis on the consequences of conclusions drawn from practical projects for the process of learning interaction rules (along with techniques for constructing interaction plans) in different complex systems. These rules constitute a basis for controlling computations in such systems and their aim is to satisfy the adaptively changing needs of agents and/or their societies that take part in these computations.

It turns out that the existing experience and knowledge, verified while managing big projects, may be a source of knowledge used by (the societies of) agents who control interactive computations in various types of complex systems. For example, the knowledge about mechanisms for the verification and validation of quality may be used to build meta-ontologies, supporting the process of learning interaction rules for controlling computations by agents.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_21

One should be aware of the fact that the ontology associated with the mechanisms for the verification and validation of quality is only an example and part of the ontology that is necessary for the implementation of a complex project. Other necessary ontologies include efficiency management, especially risk management. The ontology for adaptive strategies that support adaptive learning of interaction rules may constitute another challenge. These strategies are naturally related to the mechanisms for the verification and validation of quality (e.g., by the PDCA cycle).

One should also remember about serious problems which may arise when trying to establish ontologies and make them available to artificial agents. These problems are connected, e.g., with a need to understand, by artificial agents, numerous complex vague concepts, which appear in the ontology, along with relations between these concepts. For this purpose, mechanisms of learning the approximations of ontologies (cf. [27, 31, 33, 363]) may be used. Another problem is that in such a case, it is necessary **to provide artificial agents with techniques of approximate reasonings on concepts from such an ontology**. The reader will notice that these problems are directly related to the challenges formulated as **a motto of this book, which has been taken from Lotfi Zadeh and Judea Pearl**.

Any attempt to the problems discussed above should include proposals for modeling of objects on which interactive computations are performed as well as processes used for perceiving of interactions of physical objects in computing models for complex systems. New techniques for deeper understanding perception processes used for comprehension of perceived situations are needed. Hence, again domain knowledge can support construction of relevant ontology for perception.

The knowledge associated with the communication of various types of agents, including their cooperation or competition, is an important consequence of the conclusions discussed in this section. At this point, knowledge about interaction techniques, which help engage project participants in an effective dialogue aimed at reaching a consensus, e.g., when making decisions and/or taking some actions in a project, or knowledge about techniques of settling conflicts, based on arguments provided by various project participants, is of great value.

Currently, we are far away from developing fully effective techniques that could enable us to transfer (as well as translate) and filter human experiences related to, e.g., managing projects into knowledge that would be useful for particular complex systems, consisting of both artificial agents and human beings. Such agents should be able to assume an increasingly greater scope of roles performed by human beings in a complex system and/or project. On the other hand, it seems inevitable to use such domain knowledge due to the lack of perspectives for developing both techniques of learning and discovering complex human knowledge in a fully automatic way and techniques of using this knowledge in an effective way, as human beings do. The aforesaid lack of perspectives is associated with the complexity of the search space (for discovering necessary concepts or principles), which hinders an effective and practical implementation with the current information processing technology.

The above outline of only some of the aspects important for modeling computations of complex systems shows the complexity of framework postulates of WIsTech (FPW postulates), which include these aspects and which will be discussed in more details in Chap. 32. These postulates should constitute necessary bounds for monitoring and/or controlling interactive computations in complex systems by agents.

21.2 How to Understand the Essence of Wisdom of Project Managers and Other CSE Project Participants?

A skillful selection, discovery, and compliance with a proper family of survival march principles, when implementing complex projects, requires not only advanced knowledge, but an adequate dose of wisdom in real-time interactions. This concerns all project participants (especially managers, engineers, domain experts, and key end users). In this context, a question, which has its roots in antiquity, arises: **what is the wisdom of a manager and/or engineer?**

According to the metaphorical "Wisdom Equation" (cf. Chap. 29), in the book, the wisdom is understood as the ability to aggregate, decompose, and/or process such components as:

a. Agent's interaction with the environment and internal interactions.
b. Adaptive judgment.
c. Knowledge.

These integrated abilities are often referred to in this book as granulation. A proper granulation makes it possible to discover computational building blocks (in Valiant's sense[1]) for perception, which helps understand a current situation or choose a proper action or plan.

Thanks to granulation, agents and human beings may communicate at various levels of abstraction. The creation of proper granulation mechanisms will help bridge the gap between artificial systems and human beings when it comes to decision-making processes. This requires developing not only techniques for the approximation of vague concepts, which people use, but also techniques for the "approximation" of reasonings (made by people) on these concepts.

The "Wisdom Equation" (cf. Chap. 29) particularly means that the wisdom is more than knowledge (learnt or acquired through experiences), which is represented in some language of an agent. By intuition, we may compare this to the ability to drive a car (which stands for the ability to manage CSE projects): *perfect knowledge of road traffic rules and driving rules is not enough for being a wise driver*.

Wisdom includes the ability to engage in as optimal interactions with the environment as possible at the right time. What is particularly important, wisdom requires the ability to make adaptive judgment of the current situation, in order to be able to make plans, and the ability to undertake proper interactions at the right time and in the right place. At the same time, interactions with the environment constitute the basis for perception, which leads to the understanding of a perceived

[1] https://www.seas.harvard.edu/directory/valiant.

situation. Adaptive judgment plays a key role in the assessment of what is currently important and what remains tangential. As a result, it also constitutes the basis for the validation and improvement of interaction plans that are being implemented.

In some sense, *judgment* [181, 183, 263, 276, 349, 524] may be treated as the extension of the concept of rational reasoning (especially about the properties of computations in IGrC) due to the necessity of taking into account not only mechanisms for logical reasoning, but also other **mechanisms which affect the decision-making process. These other mechanisms may be related, for example, to perception, emotions, instincts, habits, intuition, fast thinking** [263], **and experiences.**

In the context used by us, adaptive judgment is mainly limited to adaptive strategies used for the granulation (including aggregation, decomposition, and processing) of concepts, which represent the properties related to:

a. Displaying important aspects of the physical world perceived through interactions.
b. Forecasting scenarios for a further development of the current situation with regard to those properties, which are important for an agent or a society of agents.
c. Processing knowledge and currently accepted reasoning schemes.
d. Adapting the current hierarchy of needs of an agent or a society of agents.

More information about the concept of adaptive judgment in the context of implementing complex projects may be found in Chaps. 24 and 32 of the book.

21.3 Representation of Interaction, Perception, and Learning Rules by C-granules

One should always remember that computations in IGrC are performed by the agent's control on objects, which may sometimes have a very complex structure. Both the structure itself as its properties must frequently be discovered from data gathered during interaction with the environment. These complex objects constitute one of the most important concepts of the approach to computational models, proposed in this book. They are referred to as *complex agent's granules* or *agent's c-granules*, in short (cf. FPW-03 in Chap. 32). Based on what has already been said, they must be, in some sense, "anchored" in the physical world but also, they must ensure the supply of information (to adequate agents participating in the implementation of a project or system), obtained as a result of interactions between physical objects.

Let us notice that c-granules may be used both by agents and the societies of agents. If they are used by the societies of agents, they are called meta-social c-granules (cf. FPW-03 in Chap. 32). In a sense, both the whole complex projects and their components may be perceived as the examples of meta-social c-granules.

The principles for computations in IGrC (including those performed by c-granules) may be treated as the extension of CSE principles (e.g., those included in the ABC book, BPCD). Based on the experiences and conclusions of the author presented above, in a book, we proposed a generalization of CSE principles. These principles are expressed in a form of postulates for computational models in IGrC. These models constitute the basis for the discussion included in the book (cf. Chap. 32). Moreover, we proposed some generalizations concerning risk and co-risk management (with controlling granular computations) in a form of a generalized model for efficiency management (cf. Chap. 33 and simplified models of interactive computations on c-granules presented in Chap. 36, points igc-1–igc-18). It should also be added that the principles (rules) of risk and co-risk management, similarly to the rules (principles) for all c-granules, are created as a result of the processes of discovering concepts (including ways of using them) and learning. These processes are conducted through interactions of proper c-granules with the environment. This approach differs from the classical approach to understanding risk. The classical approach relies mainly on the stationary foundations of the ontology of concepts that describe risk management. On the other hand, the approach presented in this book assumes that concepts necessary for risk management (including the concept of "risk" itself) have meanings which change with time. The dynamics of these changes depends both on the techniques of interaction between agents and the environment and some unknown external factors which affect the agent's perception of the concept of "risk."

In the IGrC computational model proposed in this book, we assume that agents not only have incomplete knowledge about their environment of activity, but also they do not entirely understand their needs and aims of actions. During interactions with the physical world, they consistently learn concepts (expressed by means of adequate c-granules), their needs (expressed by means of adequate c-granules), techniques for the fulfillment of these needs (expressed by means of adequate c-granules), and interactions plans (expressed by means of adequate c-granules) which should help satisfy their priority needs. The needs and techniques for their fulfillment are determined by models (expressed by means of adequate c-granules), constructed in a language of concepts which are constantly discovered and keep changing their meanings. What is more, the needs of agents are defined by means of actions that they undertake (expressed by means of adequate c-granules) based on the verification of the degree to which relevant concepts (expressed by means of adequate c-granules) are satisfied.

The comprehension of a perceived situation (i.e., the perception of a situation) (including the discovery and adaptive judgment of aspects which are the most important for an agent) is a basis for determining next actions that are to be performed. The results of these actions may satisfy (to some degree or to no degree at all) the properties expected by an agent.

The perception and verification of the expected consequences of actions are performed by means of adequate c-granules. These c-granules may represent interaction rules used by an agent (e.g., *if a vague condition X is fulfilled—to an appropriate degree—perform action A and expect the fulfillment—to an appropriate degree—of a vague property Y as a consequence of this action*). Thanks to such rules, agents may use their c-granules to control interactive computations performed by them, to communicate, and to construct common c-granules for the society of agents, which enable them to cooperate when performing important tasks.

The discussion above shows the **importance of mechanisms for learning the rules of interaction** between agents (who cooperate and/or compete with themselves). These rules are often discovered from available data. Due to this and the incompleteness of information an agent has about the physical world in which interactions occur, these rules are uncertain. As a result, this means that these rules must be constantly adapted to take into account the results of gaining new knowledge or experiences. In reference to complex systems engineering, this fact is sometimes emphasized by using the name of *adaptive principles* instead of a shorter version, which is *principles*.

The basic aim of an agent (and/or a group of agents) is to **fulfill his/her (their) priority needs** (cf. FPW-14, Chap. 32). The fulfillment of these needs requires a continuous use, construction (learning), and adaptation of *interaction plans* (represented by c-granules in our approach). It is important for an interaction plan to be adaptable to the constantly changing environment of the agent's activity.

In the simplest situation, an **interaction plan may be a family of rules (principles) for controlling interactions**. Analogically as in driving a car, a driver does not need a detailed itinerary (with an accuracy of 1 mm). However, she/he should know and be able to use a proper family of rules (principles) for driving a car. Such rules enable a driver to reach a destination, even if she/he is unexpectedly forced to change the route. These rules are expressed by means of concepts that are known to agents (which, in our case, also constitute c-granules).

Such a construction, implementation, and modification of interaction plans by an agent requires, in particular, a **computational model which represents the processes of perception, along with a possibly deep understanding of a constantly changing agent's situation**.

For this purpose, let us assume that an agent—on the basis of interactions with the environment—models in her/his memory images of her/his environment of activity that she/he previously recorded and forecasted. In our approach, these images are also represented by means of c-granules. Forecasted images should be connected with the **criteria and scenarios for the verification** (the compliance of a forecasted image with the actual state of the environment) and **evaluation of risk/co-risk**, which appears in a dynamically changing environment (especially from the perspective of agent's priority needs).

Through an interaction with the environment, an agent verifies a forecasted image and with the flow of information, modifies and/or specifies the recorded image, the forecasted image, and the hierarchy of needs. Moreover, she/he discovers new potentially useful concepts.

It is worth mentioning that an agent may generate in her/his memory other types of images than those listed above. For example, images showing the desired role of an agent in the environment, as well as templates with proposals for the construction of interaction plans that could help achieve the properties represented by these images, are both associated with the agent's hierarchy of needs.

The simplified model of interactive granular computations (including models of perception and understanding related to a given situation) outlined above and all its key components can be represented by c-granules. A more detailed explanation of the intuitions behind interactive granular computations presented above can be found in this book, particularly in Chaps. 27–32. In Chap. 28, we presented the most important properties of the concept of a c-granule and signalized the key ways of using the concept in models of interactive granular computations. The intuitions behind these properties are illustrated in Figs. 28.1–28.15.

21.4 Need for Constructing and Improving "a Science-Friendly Language"

The above remarks show that there are some strong **relationships between the principles (rules, principles) for the implementation of CSE projects and interaction rules in IGrC computational models**. These relationships are discussed in the book, particularly in Chap. 22 entitled "CSE as Interactive Thinking by Society of Agents" and in Chap. 31 entitled "Complexity of Designing an Ontology for Practically Useful IGrC Models."

Concepts used by a project team (represented in IGrC as c-granules) ensure mutual interactions and cooperation of the society of project participants and stakeholders. In fact, it should be emphasized that the implementation of a project depends on interactions between project participants and their interactions with the surroundings. These interactions are naturally mediated by c-granules that represent concepts used in a project. These concepts are used, e.g., to define and communicate objectives, to specify requirements, and to define tasks, plans, and techniques used in a project. Moreover, these concepts are used to make reasonings and arrangements concerning the implementation of a project, including further actions plans. Therefore, to simplify, the implementation of a project may be perceived as the discovery, construction, processing, communication, improvement, and archiving of c-granules (in this case, these are mainly social meta c-granules). At the same time, we mean here a classical approach to computational models in IGrC.

In modern systems, it is necessary to ensure the communication between various types of agents (human beings, machines, systems, or modules). Consequently, it is important to provide agents with good-quality approximations of concepts (c-granules), which do not come from their language, but which other agents use. These are usually very complex concepts, expressed in a natural language by means of vague concepts. In order to improve the communication between agents, is it is necessary to use techniques that induce the approximations of such concepts (of a suitable quality) that are only used by some agents in order to make them available (in a form of adequate c-granules) to other agents, who are the recipients of these concepts (agents-recipients).

For example, let us consider principles expressed in a natural language by means of complex vague concepts from a natural language. If we want these concepts to be effectively used by other agents-recipients (e.g., computer systems), they should be approximated with a sufficient quality by means of concepts available to these agents-recipients. In a sense, we are driving here at the necessity of constructing "a science-friendly language," which was mentioned in the motto of this book, taken from Judea Pearl.

In this way, basic project concepts (represented by c-granules), e.g., principles, are approximated by c-granules of agents-recipients to make them comprehensible to these agents (human beings, machines, or systems). Next, we strive towards a proper organization of (the societies of) agents-recipients to ensure such agent's abilities as the identification of principles and satisfiability degrees of criteria by means of arguments "for" and "against" this satisfiability.

To be more precise, survival principles act as *logical bounds expressed in a natural language*, which set a proper direction for controlling the implementation of a project. The key to success is the discovery and use of "survival march" principles, that are possibly optimal for the specificity and context of a project, and a quick identification and minimization of negative consequences of the "death march" principles. In the approach proposed in this book, concepts and expressions from the agent's language (including specifications of principles) that are constructed on the basis of these concepts are represented by c-granules. Therefore, a key role is played here by the techniques for the adaptive and interactive identification (through interactions with the environment), search, construction, and use of c-granules. In this context, we may speak about the need of an agent to learn the rules of a complex game in an adaptive way. In these rules, a significant role is played by the previously learnt approximations of complex concepts which, if satisfied to a particular degree, signalize a need to undertake actions aimed at preventing a growing risk or enforcing the positive effects, which potentially stem from a growing co-risk.

21.5 Relationships with Rough Sets and Ecorithms

The main ideas presented in the book have their roots in the studies on rough sets initiated by prof. Zdzisaw Pawlak [381, 383, 385, 463], particularly his approach to such concepts as: information systems, decision tables (as they are understood in the theory of rough sets), approximations of sets, and Boolean reasoning about vague concepts [360, 384, 386, 391, 463].

The approach to modeling computations in IGrC suggested in this book may be illustrated (cf. Sect. 34.1, conditions DT1-DT8) using decision tables (understood as sets of conditional and decisive attributes) connected with physical objects (cf. Chap. 28).

For this purpose, c-granules may be implemented (cf. Sect. 34.1) by means of decision tables, in which **decisions are treated as proposals to launch proper interactions with the surroundings by an agent and the values of conditional**

attributes are obtained by proper interactions with physical objects. Depending on the values of conditional attributes, proper interactions are activated, based on rules induced from decision tables. In practice, it may occur that when analyzing new situations (cases), the previously induced rules lead to contradictory decisions. In such a case, agents should have mechanisms for resolving conflicts (cf. [388]) in a family of activated rules. When resolving such conflicts, a major role is played by the current hierarchy of the agent's needs (which is constantly updated by adaptive judgment based on the perception of phenomena observed by an agent).

Computations in IGrC performed by c-granules only partially represent computations performed by physical objects thanks to the interaction of these objects. Therefore, the states of these computations in a "physical reality" are usually only partially known at all stages of computations (especially during the initial and final phases) performed by means of computational models in IGrC, which represent them.

Consequently, projecting the next states of computations in IGrC in relation to the current ones is burdened with some degree of uncertainty as states of computations depend on unpredictable interactions between physical objects in their surroundings. This means that there is no foreknown "constructive" definition of a function, which could perform computations in IGrC. However, through interactions (including "prototyping," which is well-known in software engineering and means complex iterations of interactions performed in a PDCA cycle) with the surroundings, it is possible to learn the approximation of this function. Hence, computations in IGrC cannot be implemented by means of classical algorithms performed by modern computers, which draw on the concept of computations based on the Turing machine.

As a result, *ecorithms*,[2] as defined by Valiant [521], have become a more suitable tool. Apart from analogies to Valiant's ecorithms, algorithms based on IGrC computational models, which were proposed in this book, have a wide range of other features that stem from the motivations of scientific research in numerous domains (e.g., learning systems, adaptive complex systems, soft-computing, multi-agent systems, natural computations). IGrC computational models are also related to the foundations of AI, particularly the understanding of machine learning.

The studies conducted on computations in IGrC are also significant for natural computing [420]. At this point, we will focus only on one aspect related to the proposal by Leslie Valiant [521], the so-called probably approximately correct (PAC) algorithms. To simplify, in the PAC approach [521], learning systems, after a sufficient period of learning (and using a proper strategy), **have a great chance of discovering a satisfactory (from a practical point of view) approximation of a decision function, which constitutes the solution to a problem**.

However, one should remember that we do not have infinite resources and often, in the "open" world, there may occur interactions which will change the conditions, motivations, and objectives associated with performing computations. This unpredictability of learning in a real environment was emphasized, for example, by John Holland, who in [210] used the metaphor of learning how to survive in a jungle. In this metaphor, he stresses out the role of the limits imposed on interactions

[2] https://www.quantamagazine.org/20160128-ecorithm-computers-and-life/.

between animals and the surroundings by the so-called "niches." The balance of interactions—which ensures the survival of co-existing organisms—in such a niche may be disturbed by some unpredictable factors in the surrounding environment (e.g., someone may poke a stick into an anthill and disturb the local balance of a colony of ants).

The aforesaid niches constitute concepts, which require further intensive basic research. This may be a step towards explaining how societies of organisms evolved in nature, leading to the formation of respective niches, possibly long before the condition of learning on sufficiently long samples, according to PAC, could be satisfied. It seems that computational models based on IGrC may be a starting point for such research. Therefore, a further development of IGrC computational models may stimulate the research on the character of computations in nature and, by the same token, the creation of foundations for natural computing [420].

Part V
CSE as a Metaphor of Mind "Computations"

Chapter 22
CSE as Interactive Thinking by Society of Agents

[...] psychologists tried to imitate physicists - by searching for compact sets of laws to explain what happens inside our brains. However, no such simple set of laws exists, because every brain has hundreds of parts, each of which evolved to do certain particular kinds of jobs; some of them recognize situations, others tell muscles to execute actions, others formulate goals and plans, and yet others accumulate and use enormous bodies of knowledge.
And though we don't yet know enough about how each of those brain-centers works, we do know their construction is based on information that is contained in tens of thousands of inherited genes, so that each brain-part works in a way that depends on a somewhat different set of laws

Marvin Lee Minsky [341, p. 2].

On the basis of a substantial and practical experience in CSE, it can be said, in short, that it involves solving many complex problems, discovering some aspects of a constantly changing project and implementing mutually entangled activities among the societies of agents (including humans, computer systems and/or automatic control systems), which attempt to carry out assigned tasks through mutual interaction.

This seemingly trivial observation means that **in order to construct such a system, we need networks of cooperating agents with high level of reasoning and mutual communication, as well as the ability to quickly learn complex vague concepts that may change over time**.

Societies of agents participating in this kind of activity must learn the core concepts that define each project, especially its ever-changing requirements, including the final products and/or services that form the target complex system. What is even worse,

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_22

these ever-changing project requirements, in the case of complex systems, are often only partly understood by the participating agents.

Thus, it would be futile to expect the societies of agents to successfully implement any system under such circumstances. Therefore, nowadays, virtually no one expects a complete success of a CSE project (right after its development and deployment, especially when it is innovative), built in a dynamic and unknown environment, where changing requirements were only partly understood by large societies of component agents.

Hence, big corporations, before they decide to launch their innovative, highly complex technologies (e.g., airplanes, cars, or software) into the mass market, usually had to test their new products on a selected group of users, who are already accustomed, in both psychological and business terms, to unexpected glitches.

Moreover, they are prepared to face the loss of such testing customers as well. Thus, this shows the advantage of subjecting new complex products to an elaborate testing cycle before they can be marketed to key customers.

Otherwise, a given corporation risks losing its strategic customers and consequently, weakening their market position.

Nowadays, modern CSE provides many potential techniques for minimizing the risk of negative consequences, resulting from the unknown and ever-changing requirements of the users. Possible techniques of this kind focus largely on protective and preventive measures that aim at satisfying, to the best extent possible, potential requirements that are not fully understood.

A classic example of such a technique is fast prototyping and the agile software philosophy (cf. Chap. 10, especially Fig. 10.11). Analysts often try to over-please customers with a whole spectrum of excessive functionalities, in case anyone complains that an important batch of functionalities have been omitted. In consequence, a great number of functionalities that have been implemented will never be used after the project is launched.

In all areas of CSE, ample experience in specific domain knowledge is gathered through decades and then, through iterations transformed into the sets of professional codes of conducts, recommendations and standards. Thus, in each application domain, increasingly advanced standards and CSE techniques constantly emerge.

The main disadvantage of these techniques is that they are usually expressed by means of hermetic terminologies, often based on the practice of obfuscation, specific to a given CSE application domain. Therefore, it is difficult to expect a quick knowledge transfer between different CSE domains. On the other hand, practitioners can notice many common mechanisms and practices that can be transferred between different CSE domains.

Consequently, **there emerges a need for a better understanding of concepts, rules and mechanisms that are common to all possible fields of CSE application**. These problems are particularly visible already at an early stage of the project's implementation, when efforts are being made to improve either communication (both within the developing team and with the surrounding environment), aimed at a more efficient specification of problems to be solved within a given project, or communication that is to boost the efficiency of collective problem-solving.

Thus, the whole problem comes down to an obstacle in the form of the natural language, used by the project's participants, and the quality of concepts used and processed therein. This includes the language in which the participants (agents) "think" (reason). Without a doubt, the quality of knowledge processing (carried out both individually by an agent and collectively within a team), with the aim to solve the consecutive issues that arise within a given project, has an impact on the quality of the project's final results.

A beginner in CSE might think that the natural language, with its vague concepts, is the main culprit and thus, propose all the participants of a project to learn a mandatory language of mathematics and physics. Then, all "inexactitudes," which otherwise could lead to a misunderstanding, would be eliminated. It would be equally easy to apply the theoretical apparatus of mathematics and physics and build supporting computer programs. At this moment, however, paradoxes involving mixing natural languages with the mathematical language come to mind. A list of such paradoxes can be found in Internet.[1]

Many of these paradoxes strongly indicate that the application of mathematics (for instance, the techniques of probability) for solving practical problems is meaningful only to an approximate extent, and will be truly useful only after appropriate interpretations of mathematical concepts, within a real domain. The process of establishing such interpretations is called, by specialists, "the construction of a model" (for example, establishing the probabilistic space for objects, investigated by statistical methods).

It should be noted that mathematics (e.g., statistics) cannot solve the modeling problem, which means that it cannot decide how and which model should be chosen to solve a given problem (in other words, which probabilistic space to select). Finding such a model is sometimes even impossible (e.g., establishing a model to predict the probability of an atomic war breaking out tomorrow). The difficulties with the construction of models are well illustrated by numerous paradoxes, pertaining to the application of probability (e.g., the paradox of random numbers, of a game's duration and others [493]).

From a didactic point of view, the difficulty in establishing a probabilistic model is notably visible in the case of the Bertrand paradox.[2] The conclusion here is that the application of a strict scientific language depends to a large extent on the appropriate interpretation (or modeling) of that language within our physical reality. As shown by many practical cases (e.g., in financial risk management), contemporary concept-manipulating tools, as well as the techniques for modeling complex concepts and reasoning about them, are still far beyond our expectations.

It is obvious that all non-trivial activities of a team involved in the development of a complex project, require a solid set-up and coordination among numerous different

---

[1] http://en.wikipedia.org/wiki/List_of_paradoxes.

[2] http://en.wikipedia.org/wiki/Bertrand_paradox_(probability).

thoughts, communications, and, particularly, thought-refining processes that aim to meet the continuously changing and vague requirements of a project.

In this sense, CSE can be considered as a highly complex computing model of the aforesaid processes. The better we understand these models, the greater chance we have to effectively support a successful management and implementation of CSE projects.

The main premise of the approach, proposed above, is a concept of a c-granule as a central entity in IGrC models, using which a better understanding of the processes such as human thinking, communicating, and thought-refining can be aimed to. The c-granules, in particular support the learning of complex vague concepts by agents involved in a project. By complex vague concepts, we understand the vague concepts, any relations between them, and properties of approximate reasoning processes about them.

Let us recall that the approximate reasoning schemes cannot usually be used at any level of approximate reasoning. It is unlikely to the classic reasoning rules in typical formal/natural deduction/induction systems, where they can be applied at any level. It derives from the fact that as the complexity and depth of approximate reasoning increase, so the vagueness of their result. This means that multiple repetition of deductive reasoning in the sense of classical logics (i.e., conserving truth) is not applicable to reasoning about vague concepts. This observation is valid to inductive reasoning (i.e., conserving falsehood), widely used in science.

Considering the high complexity of practical projects involving complex vague concepts, and bearing in mind that the meaning of such concepts may frequently change over time, systems that attempt to learn these concepts must conduct a systematic, interactive verification on how well a concept has been learned at a given point of the project's implementation. Such a systematic verification, checking the quality of our understanding of basic concepts used in a project, as well as the verification of our understanding of the concepts pertaining to the deployment of this very project in its environment, are extremely important. The final success depends heavily on the ability of agents to understand each other (especially while carrying out complex tasks).

This is particularly true for tasks requiring a collective endeavour for solving of complex issues, that is tasks requiring intensive and valid reasoning about dynamically changing complex vague concepts. The better we understand the process of thinking of a developed society, more we are able to steer the process and complete the task satisfactorily. Besides, the more our reasoning model is compatible with the requirements of a project, the better chances we have for a success. It is worth to note that within this approach, the conceptual and technical apparatus for the rationally substantive understanding and construction of computing models, which support interactive learning of complex vague concepts, are paramount to the success of a project. The construction of such computing models should also be adapted to the context knowledge, which may have a potential impact on our understanding of: the project's requirements, the organizational and communication culture of the participants, their motivation, habits and the external constraints of a project as well.

To sum up—the **success of CSE projects depends heavily on the ability to understand, learn and refine the human thinking, communicating, and thought-refining processes of the key agents involved in those projects**. A quality measure for this type of projects is the extent to which they can lead to the construction of optimal solutions that result in a successful implementation of projects.

Of particular importance is the ability to turn those competencies into systematic development of a project. **Agents in turn must refine their level of knowledge and the ability to apply techniques of interactive learning, discovering and/or concealing complex vague concepts. It is particularly important to conduct a systematic learning of concepts related to the key knowledge domains, especially those connected with specific project requirements, perform a decomposition of problems and adapt appropriate problem-solving techniques**.

Chapter 23
The Model of Thinking Problem (MT): How Do We Understand and Describe the Processes of Human Thinking, Communicating and Thought-Refinement?

How Can the Human Mind Occur in the Physical Universe?
J. R. Anderson [6]

23.1 Definition of the Model of Thinking Problem (MT)

Taking into account that CSE can be treated as the process of interactive thinking by the society of agents, one can ask the following question:

How to understand and describe the computing models of such processes as human thinking, communicating, and thought-refining?

Hereafter, this question, along with its scientific and technological aspects, will be referred to in brief as the *MT problem*. Since such a formulation of the problem might be too general, we should elaborate on how we come to understand it. To this end, we will herein propose the following definition of MT problem:

1. How to understand and describe (if possible—in the form of computing models implemented within an environment of intelligent agents) the following methods:
    a. Emulation and systematic refinement of the human thinking processes (in particular human reasoning).
    b. Effective communication among intelligent agents.
    c. Learning of concepts, their relations, behavioral patterns, plans, adaptation strategies, hierarchies of needs and values.
    d. Acquisition of required knowledge, new thinking schemes and techniques for refining thinking schemes.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_23

2. Enabling the agents to undertake, whenever possible, "wise" actions (or interaction plans) in an environment. By "wise" actions (or action plans), we understand actions (or action plans) that yield the most optimal possible approximate satisfaction of the following two conditions:
   a. To complete the goals and fulfil the conditions desired by the agents, as required from a properly executed action or action plan.
   b. To meet the needs of the agents as required by their hierarchies of needs and values (should agents have such), which are on the agenda within the context of actions (interaction plans) being carried out by the agents.

23.2 Some Acceptance Criteria for Potential MT Solutions

From the viewpoint of applications in CSE, potential MT solutions should meet a number of acceptance criteria requirements. Those criteria derive first of all from the analysis and practice of complex projects that have been already completed. This type of project often involves many potential pitfalls, which greatly hinder "wise" actions (action plans) carried out by agents. Therefore, each acceptable solution to the MT problem should minimize the risk associated with those pitfalls. The most important ones are presented below:

1. The goals are specified by means of vague, complex concepts and relations, which means that they are only approximately expressed.
2. While completing an action, we often navigate with incomplete and uncertain information on approximated concepts. In consequence, during the process, an agent is usually uncertain of such information as:
   a. Whether and to what extent it is approaching the goal of the action.
   b. What risk is associated with a given scenario, aimed at reaching the action's goals.
3. The goals evolve over time, due to reasons unknown to us. For instance, while managing the risk associated with the bankruptcy of a bank, despite of the fact that all formal safeguarding criteria are met, we may not fathom that the external organizations had already taken some actions leading to the bankruptcy of a bank.
4. Concepts and attributes that we employ to steer the path to the project's objectives are imprecise and they keep evolving with time. For instance, the risk indices for the failure of a project may change, which means that we may wrongly calculate this risk using outdated indices or the data of dubious quality.
5. The goals are not set according to our real intentions. For instance, in the case of large and complex IT projects, upon their completion the already documented user requirements very often turn out to be different from the actual ones. Hence, the need for the constant, systematic and comprehensive interactions with the

actual domain experts, as well as for the systematic trials in situ, in order to verify the actual consequences of any thought experiment, is observed.
6. Each non-trivial action calls for the use of specific domain knowledge, associated with the execution of the action. Very often, the access to such knowledge is impeded due to the inability to identify the required knowledge or to the limited resources needed for its transfer. In effect, the deficiency of domain knowledge usually hinder the successful execution of a given action.
7. Each action is carried out in a real, physical environment, often governed by the rules unknown to us. This environment is open and requires a constant vigilance for metaphorical "disasters" when least is expected (Murphy's law).

This list of acceptance criteria for potential solution to the MT problem is naturally far from being exhaustive. It works similarly both from the viewpoint of the CSE applications as well as potential applications of the MT problem in other fields. To this list, we should add one highly important, universal acceptance criteria for potential MT solutions—the criteria of a successful deployment. The verification of this criteria, however, requires finding a solution that would satisfy the above-mentioned criteria in the first step.

In the line of Poggio and Smale [400], who say that the problem of understanding intelligence is said to have been the greatest problem in science for a century, we also concur that there is no known solution to the MT problem that satisfies all acceptance criteria that have been mentioned above. Over the last decades, there were many proposals concerning the general frameworks on which such solutions can be based. The ideas by Minsky, described in his books [340, 341] and associated with contemporary research trends, seem particularly interesting. In [342], we may

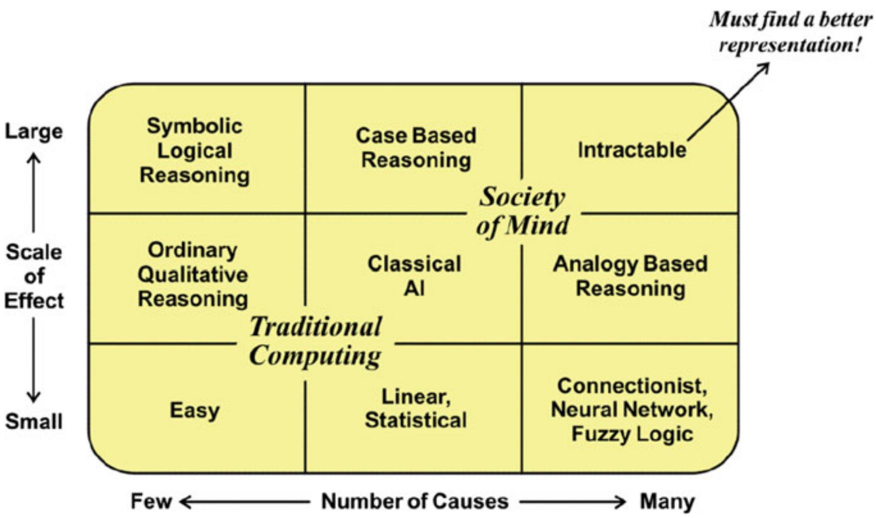


Fig. 23.1 Minsky's challenge [477]

find especially interesting proposals on the issue of designing architectures for the human-level intelligence, aimed at solving nontrivial problems and learning by language understanding. Moreover, in this book, Minsky also proposed some research directions on such architectures for the human-level intelligence.

A visual adaptation of this approach, found in [477], is presented in Fig. 23.1.

23.3 MT Solution Framework

Hereinafter presented proposals steer toward a metaphoric understanding of the following three proposals:

1. Mervin Minsky's ideas presented in his books [340, 341];
2. Zdzisaw Pawlak's ideas on rough sets [384, 386, 391];
3. LoftiZadeh's ideas on CWW [550, 554].

On the basis of these three proposals and a substantial real life experience acquired during the implementation of projects that have been already discussed in this book, we have established a proposal for a new paradigm on building computing models for AI, called the WisTech [244–247], in cooperation with Professor Andrzej Skowron.

The lessons learned from numerous attempts made in the last decades at modelling thinking processes have shown that the task still remains as a big challenge. Equipped with the ontology of modern mathematics and the Church-Turing thesis, one can attempt to construct such models by way of adapting computation models based on the Turing machines. Many practitioners, however, are convinced that the effectiveness of this approach is quite limited and hence, alternative methods are needed. The judgment is in the line of the conclusions, drawn from the experience in the creation and management of complex projects that have been described in this book. The models of interactive computing, that are being discussed, aim at describing the key processes in CSE in terms of models for interactive learning of such complex concepts as project requirements, risk, environment, cooperation and competition among agents.

Probably, the most concise idea of the approach towards an MT solution framework based on the WisTech, presented in the following parts of this book, is as follows:

Interactive granular computing is carried out by adaptive c-granule networks (a metaphor of the Minsky's proposals, laid out in his books entitled "The Society of Mind" and "The Emotion Machine" [340, 341]). Information systems and decision systems are crucial to the creation of such networks. These networks represent classifiers that change over time and, in a more general sense, dynamic theories on ontological structures and reasoning schemes of concepts and their relations. Moreover, these theories are associated with a hierarchy of needs and values, which are elicited and updated as a result of interaction processes between the experts, environment and computing conducted by the networks. An important aspect of this *approach is to assume a physical completion of interactive computation in an open environment, of which agents have only partial and uncertain knowledge*.

The approach differs from many attempts that have been already made to solve the MT problem. It is due to the fact that the proposed computation models cannot be conveyed by processes, expressible in the calculus of predicates (which means they are not the examples of computation models based on the Turing machines).

For a better presentation of the aforesaid differences, as well as the context that gave rise to our approach to the models of GrC based on the WisTech, we shall make a number of remarks on the ontology of modern mathematics, computation models based on the Turing machines, and constructive mathematics according to the Brouwer's intuitionism.

Chapter 24
Judgment, Constructive Mathematics, and Intuitionism

Experience without theory is blind, but theory without experience is mere intellectual play.

Immanuel Kant

In Sect. 12.3, we have introduced potential problems associated with the misuse of non-constructive techniques of modern mathematics in CSE. In our considerations, we often referred to the consequences stemming from the *Axiom of Choice* (AC). Numerous experts on the foundations of contemporary mathematics agree with the following opinion regarding AC, taken from the book [145]:

> probably the most interesting and, in spite of its late appearance, the most discussed axiom of mathematics, second only to Euclid's axiom of parallels which was introduced more than two thousand years ago.

AC is also strongly correlated with intuitionistic logic.[1] In a sense [113], AC implies the non-constructivity of mathematical foundations (to be more precise, AC endorses classical logic instead of intuitionistic logic).

In the following chapter, we will analyze the relations between AC and some types of reasoning used in a constructive mathematics based on intuitionistic logic. These relations provide an important context for better understanding of the concepts related to thinking and judgment processes.

Many mathematicians, when confronted with the idea of AC (formulated in 1904 by Ernest Zermelo), quickly realized that it is inconsistent with the intuition of admissible mathematical constructions. For example, Émile Borel, was particularly articulate his opinion and stated, in 1904 in a note sent to Mathematische Annalen [63], what follows:

[1] http://en.wikipedia.org/wiki/Intuitionistic_logic.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_24

> It seems to me that the objection against it is also valid for every reasoning where one assumes an arbitrary choice made an uncountable number of times, for such reasoning does not belong in mathematics.

This firm opposition of Borel resulted in a vivid debate among other mathematicians.

On one side, Jacques Hadamard supported Ernest Zermelo, whereas on the other side, René-Louis Baire, Henri Lebesgue and Luitzen Egbertus Jan Brouwer supported Émile Borel. Indeed, the infinite sets constructed by the "reasoning where one assumes an arbitrary choice made an uncountable number of times" are difficult to imagine.

On the other hand, if there was a method to select some elements of such sets in a determined fashion, then one can envision a selection process and thus the set that emerges as a result. Hence, perhaps it would be better to limit ourselves to mathematics which allows only for such a construction of sets, which can be comprehended by the human mind. The intuitionism associated with Carl Friedrich Gauss, Leopold Korcneker, Henri Poincaré, Émile Borel and Henri Lebesgue is an example of such a trend. However, it is generally accepted that the intuitionism was founded by L.E.J. Brouwer in his Ph.D. thesis from 1907 [70, 71, 525].

Let us very briefly remind that Brouwer's original approach to intuitionism assumed that the truth of any mathematical statement was a subjective process in our minds: a mathematical statement corresponds to a mental construction, and a mathematician can assert the truth of a statement only by verifying the validity of that construction by her/his intuition.

In consequence, for Brouwer, pure mathematics consists primarily of the act of thinking by making special mental constructions. The key point of the construction processes is the intuition of the time flow. This intuition is based on the perception of the form "one thing and again a thing, and a continuum in between." This form, which unites the discrete and the continuous, is called by Brouwer as "the empty two-ity." It is one of the basic intuitions of intuitionistic mathematics: the discrete cannot be reduced to the continuous, nor the continuous to the discrete.

The essence of this limitation to sets constructible in our mind was elegantly summarized and specified by Arend Heyting, a student of Brouwer, in his book [202]:

> In the study of mental mathematical constructions "to exist" must be synonymous with "to be constructed."

Let us remind that the intuitionistic mental constructions do not accept proofs for the existence of objects for which no actual construction is available. In other words, it is not possible to prove existential propositions by contradiction. In consequence, the law of excluded middle is rejected. Therefore, the question whether a given object is within the scope of a complex, vague concept may sometimes not be resolved using the intuitionistic base. For instance, at times, we are unable to tell whether a person suffers from a given disease or not.

The rejection of the law of excluded middle naturally turns many crisp concepts into vague concepts, especially those defined by classical mathematics, which appear to be vague within the intuitionistic logic. For example, let us consider the definition of the "π" number, which is the ratio of a circle's circumference to its diameter. It is an irrational number, which means that we do not know all of its digits.

However, from a practical point of view, an adequate approximation of π is sufficient. To paraphrase Michael Dummett,

> [...] the essence of infinity of the sequence of digits in decimal representation of the number, lies in the conception of a structure which is always in growth, precisely because the process of construction is never completed.

In other words, we may have a recipe for computing subsequent digits in the decimal representation of such irrational numbers as π, but we may never know all of these digits. However, this does not apply to rational numbers. A constructive proof of rationality immediately yields a representation in the form of a fraction between integers, which means we will obtain all the digits in a decimal representation in a finite number of steps.

An analogous situation may be encountered when learning the descriptions of concepts, expressed in a language with a finite number of attributes. No matter how many instances of objects are available to investigate, it would be still impossible to describe them using this language. As with finding the subsequent digits in the representation of π, here we can also only learn the approximations of a concept. In contrast, learning a crisp concept in this language is like finding the digits in the representation of a rational number, which means that it can be completed by examining the instances of objects which have a finite number.

An interesting insight on how the "intuitionists" perceived the polarization of views between the proponents and opponents of the ontology of Cantor's set theory can be found in [120]:

> From an intuitionistic standpoint, the platonistic conception is the result of blatantly transferring, from the finite case to the infinite one, a picture appropriate only to the former. In making this transference, the platonist destroys the whole essence of infinity, which lies in the conception of a structure which is always in growth, precisely because the process of construction is never completed.

Here, Michael Dummett noted that while constructing many objects, we only have access to the fragments that have been already built or to parts that are currently being built. Those parts constitute only the approximations of the target concepts.

Intuitionistic constructivism in mathematics has yielded many interpretations. It was partly due to the differences in the understanding of the admissible construction concept. Brouwer, the main inventor of intuitionism [524], viewed mathematics as an effect of the intellectual constructions of mathematicians.

He opposed the view by David Hilbert, in which mathematical entities are the results of the admissible formal proofs within a formalized symbolic deduction system. However, one of the best known approaches to intuitionistic logic in the 20th century is the Hilbert style formalization.

Namely, Heyting provided a formal basis for Brouwer's program of intuitionism in a framework that was similar to Hilbert's program. Of course, numerous tautologies of classical logic can no longer be proven within intuitionistic logic. Examples include not only the law of excluded middle, $p \vee \neg p$, but also Peirce's law, $((p \to q) \to p) \to p$, as well as double negation elimination.

Radu Diaconescu [113] in 1975, showed that **the Axiom of Choice implies the Principle of Excluded Middle. More specifically, he showed that every topos that satisfies the Axiom of Choice is Boolean**. Along with the results which were really intriguing, simple arguments used by Diaconescu, based on classical logic, were also very impressive.

Andrej Bauer presented a very simple proof of Diaconescu's theorem in "Five Stages of Accepting Constructive Mathematics".[2] The argument is as follows: Let $2 = \{0, 1\}$ and let p be a proposition (which doesn't depend on x, y). Consider the sets:

$$A = \{x \in 2 : x = 0 \vee p\}, \tag{24.1}$$

$$B = \{y \in 2 : y = 1 \vee p\}. \tag{24.2}$$

These are the inhabited sets so there is a choice function $f : \{A, B\} \to 2$. Since 2 is decidable set, we have $f(A) = f(B) \vee f(A) \neq f(B)$. In the first case, p must hold for if $f(A) = f(B) = 0$ then $0 \in B$ and hence p and if $f(A) = f(B) = 1$ then $1 \in A$ and hence p. If $f(A) \neq f(B)$ then we must have $A \neq B$ and thus $\neg p$. Therefore, we conclude that $p \vee \neg p$.

The rejection of the Law of Excluded Middle by intuitionists is quite natural. However, many logicians consider weaker forms of this law permissible, while investigating various features of constructivism. Therefore, many intermediate logics between intuitionistic logic and classical logic have been the subject of research. Together with the works on intuitionistic constructivism, there were many attempts to reconstruct modern mathematics towards a more "constructive" approach using sources considered as intuition according to Brouwer.

These attempts followed the advances in approaching the concept of "constructive mathematics" from the viewpoint of intuitionism. As the IT technologies developed, they also become a test bed for computer applications that can perform intuitionistic constructions. In the 20th century, there were many attractive approaches to constructing intuitionistic mathematical entities and, as a consequence, to a reconstruction of the whole mathematics in the spirit of intuitionism. Some examples can be found in [16, 39, 150, 174, 277, 306, 318, 326, 329, 507].

There are many differences across various authors in the understanding and implementation of the concept of "constructive" mathematics. From this diversity of

[2] http://video.ias.edu/members/1213/0318-AndrejBauer.

views, one can conclude that the difference between the constructible and the non-constructible may be quite subjective judgment. This, in turn, constitutes a judgment, which is one of the fundamental concepts of intuitionism [37, 76, 116, 131, 183, 188, 276, 316, 349, 397, 427, 504].

This list of books devoted to the understanding of the concept of judgment shows that the concept has been central to the interpretations and disputes of philosophical and logical schools. Nevertheless, many logicians consider Gottlob Frege [37] as the initiator of the modern approach to judgment. It has naturally become the subject of analysis and investigation by intuitionists. What seems the most important is that the approach by Frege to judgment inspired Per Martin-Löf in his works on constructive dependent type theory (CDTT) [39, 329]. These works can be considered one of the main scientific achievements in the recent decades in the field of understanding and establishing the foundations of constructive mathematics. The basic intuition of the concept of "dependency" in the dependent type theory was elegantly put forth in [39] as follows:

> In a simple type theory or a local set theory, each type is independent of other types and is thus, so to speak, absolute or static; this holds in particular of the type of propositions or truth values. Now formulas or propositional functions in general manifest variation, since their values vary over, or depend on, the domain(s) of their free variables. Because of this they cannot be accurately represented as static types. This limitation makes it impossible for a simple type theory to realize faithfully the propositions as types doctrine. In order to achieve this it is necessary to develop a theory of "variable" or dependent types, wherein types can depend on, or vary over other types. In a dependent type theory, type symbols may take the form $B(x)$, with x a variable of a given type, $A : B(x)$ is then a type dependent on or varying over the type A.

A key concept of this book is the metaphoric Wisdom Equation. In this equation, the concept of judgment plays a very important role and it is understood more broadly than in the works by Martin-Löf. Nevertheless, we shall investigate the concept as understood by Martin-Löf in order to have a better perspective of its wider use in this book. A starting point for understanding the concept of "judgment" by Martin-Löf is his reference to a proposal by Frege, who called for a distinction between "judgment" and "proposition." Martin-Löf illustrated this distinction on the third page of his book [329]:

> When we hold a proposition to be true, we make a judgment (cf. Fig. 24.1). In particular, the premises and conclusion of a logical inference are judgments. The distinction between propositions and judgments was clear from Frege to Principia. These notions have later been replaced by the formalistic notions of formula and theorem (in a formal system), respectively.

A little further in his book, Martin-Löf also clarified that the distinction between the two concepts, judgment and proposition, was intended to enhance the independence of his approach to constructive mathematics from the mainstream, 20th century approach that was in accordance with the "metamathematical tradition originated by Hilbert." This separation from Hilbert's approach brought the approach by Martin-Löf closer to the core of intuitionism, posited by Brouwer (i.e., to view mathematical entities as the results of intellectual constructions by mathematicians and, at the

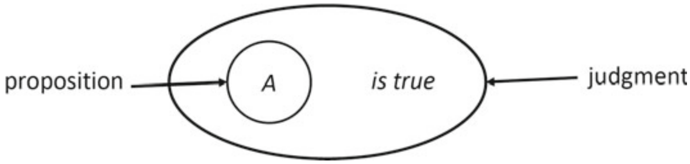


Fig. 24.1 "When we hold a proposition to be true, then we make a judgment" [329]

same time, to separate mathematics from formal theorem proving within formalized deduction systems). The idea is expressed by Martin-Löf as follows:

> What we do here is meant to be closer to ordinary mathematical practice. We will avoid keeping form and meaning (content) apart. Instead we will at the same time display certain forms of judgment and inference that are used in mathematical proofs and explain them semantically. Thus we make explicit what is usually implicitly taken for granted. When one treats logic as any other branch of mathematics, as in the metamathematical tradition originated by Hilbert, such judgments and inferences are only partially and formally represented in the so-called object language, while they are implicitly used, as in any other branch of mathematics, in the so-called metalanguage. Our main aim is to build up a system of formal rules representing in the best possible way informal (mathematical) reasoning. In the usual natural deduction style, the rules given are not quite formal.

The above presented intuition and motivation of the distinction between "judgment" and "proposition" is quite clear. Let us not forget, however, that there were many works (including those by Martin-Löf) which pointed to a deeper subtlety of this distinction. A concise summary of this fine distinction can be found in [39]:

> A key element in Martin-Löf's formulation of type theory is the distinction, which goes back to Frege, between propositions and judgments. Propositions (which, as we have seen, in Martin-Löf's systems are identified with types) are syntactical objects on which mathematical operations can be performed and which bear certain formal relationships to other syntactical objects called proofs. Propositions and proofs are, so to speak, objective constituents of the system. Judgments, on the other hand, typically involve the idealist notion of "understanding" or "grasping the meaning of." Thus, for example, while $2 + 2 = 4$ is a proposition, "$2+2=4$ is a proposition" and "$2+2=4$ is a true proposition" are judgments. Martin-Löf also follows Frege in taking the rules of inference of logic to concern judgments rather than propositions. Thus, for example, the correct form of the rule of $\rightarrow$ -elimination is not
>
> $$\frac{A,\ A \rightarrow B}{B} \tag{24.3}$$
>
> *but*
>
> $$\frac{A\ true,\ A \rightarrow B\ true}{B\ true} \tag{24.4}$$
>
> That is, the rule does not say that the proposition B follows from the propositions A and $A \rightarrow B$, but that the truth of the proposition B follows from the truth of the proposition A conjoined with that of $A \rightarrow B$. In general, judgments may be characterized as expressions which appear at the conclusions of rules of inference. Another important respect in which Martin-Löf follows Frege is in his insistence that judgments and formal rules be accompanied by full explanations of their meaning. (This is to be contrasted with the usual model-theoretic semantics which is really nothing more than a translation of one object-language into another.) In particular, the judgment A is a proposition may be made only

> when one knows what a (canonical) proof of A is, and the judgment A is a true proposition only when one knows how to find such a proof. Judgments, and the notion of truth, are thus seen to be mind-dependent.

In the original approach by Brouwer, an important aspect of judgment was the intuition resulting from the current knowledge and experience. He strongly emphasized that the states of knowledge are changing over time and a seasoned mathematician, using his intuition and experience, knows when an "infinite" construction is no longer worth continuing. It is very difficult to model this important bond between intuition, knowledge, and professional experience to a satisfactory degree in line of Matin-Löf's approach to intuitionism. Nevertheless, let us recall that "Rome was not built in a day" and certainly, the results obtained by Martin-Löf were an important step towards Brouwer's intuitionism.

On the other hand, the efforts to lay the foundations of constructive mathematics are very important for a proper understanding of the models of human thinking processes and, as a consequence, they should bring us closer to finding the right approach to the most optimal and possible solutions to the TPGP (cf. Chap. 11 and Sect. 1.2.2). While looking for these solutions, we should keep in mind that in the case of CSE, virtually all important decisions concerning the design constitute judgments.

However, instead of making such judgments about expressions as "A is true," we make judgments like "A is C," where C represents such concepts as "user requirements," "project risk," "clear to the working team." Thus, the figure from Martin-Löf's book, illustrating the distinction between judgment and proposition, is an important pillar of the approach proposed in this book, which is largely in accordance with many suggestions made by the creators of intuitionism and is substantiated by the experience in implementing complex projects. In particular, this approach refers to such foundations of intuitionism as:

1. Key role of judgment (including judgment pertaining to intuition, resulting from pre-acquired knowledge and experience).
2. Avoidance of unnecessary formalisms (backing out from thinking about a project through the lenses of overly formalized methods).
3. Emphasis on thinking (in the case of complex projects, this means collective problem solving) instead of formal "proposition proving" during the implementation of a project.

In order to better explain the above conclusions and introduce a somewhat broader perspective of the concept of judgment within the Wisdom Equation, let us use an example in which driving a car serves as a metaphor for the project management. While planning a route and driving a car to our destination, we are not interested in a path measured to the nearest millimeter. In this process, the ability to make judgment regarding all important details relevant to driving a car is more essential. This is particularly true in the case of the ability to quickly assess and apply the "Stop" criteria (e.g., to evaluate whether the obtained solution satisfies the design requirements). This metaphor emphasizes the importance of a swift and apt application of judgments while driving a car. We will continue to use it in more details further in the book.

Chapter 25
Thinking as a "Languageless Activity of the Mind" Having Its Origin in the Perception of a Move of Time

Logic will get you from A to B.
Imagination will take you everywhere.
Albert Einstein (http://www.brainyquote.com/quotes/quotes/a/alberteins121643.htm)

It is quite revealing that many outstanding minds reject to treat the understanding of creativity of the process of rational thinking, as the processing of grammatical statements expressed in a natural language. Moreover, numerous scientists reject the role of symbolic language expressions as a basic mechanism for acquiring knowledge and learning. On the other hand, a great number of scientists consider the activity of imagination, controlled by the intuitions and hierarchy of agent's needs, as significant in the process of creative thinking. To illustrate this line of reasoning, let us recall some popular statements by Albert Einstein:

1. *I never came upon any of my discoveries through the process of rational thinking.*
2. *A really valuable thing is intuition.*
3. *The words or the language, as they are written or spoken, do not seem to play any role in my mechanism of thought.*

In the discussions concerning the advent of intuitionism, presented in the previous chapter, mutual relationships between thinking and language processing activities played a central role.

Due to the fact that this kind of discussions took place primarily among mathematicians and logicians, the concept of thinking, either for pursuing mathematics or rational reasoning, constitutes an important reference point. In order to have better understanding about the relationships between the process of thinking of mathematicians and the processing of symbols that prepare the precise language for modern mathematics, let us analyze the opinions of some prominent scientists. We will start

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_25

with the view expressed by Brouwer. His opinions in this respect were quite unambiguous and so important for the creators of intuitionism, that they are sometimes collectively referred to as the “First Act of Intuitionism” [524]:

FIRST ACT OF INTUITIONISM

> Completely separating mathematics from mathematical language and hence from the phenomena of language described by theoretical logic, recognising that intuitionistic mathematics is an essentially languageless activity of the mind having its origin in the perception of a move of time. This perception of a move of time may be described as the falling apart of a life moment into two distinct things, one of which gives way to the other, but is retained by memory. If the two-ity thus born is divested of all quality, it passes into the empty form of the common substratum of all two-ities. And it is this common substratum, this empty form, which is the basic intuition of mathematics.
>
> Inner experience reveals how, by unlimited unfolding of the basic intuition, much of ‘separable’ mathematics can be rebuilt in a suitably modified form. In the edifice of mathematical thought thus erected, language plays no part other than that of an efficient, but never infallible or exact, technique for memorising mathematical constructions, and for communicating them to others, so that mathematical language by itself can never create new mathematical systems.

In his another work [70] (p. 451), Brouwer evokes a highly suggestive metaphor which defines the relationship between mathematics and formal languages expressing it:

> Formal language accompanies mathematics as the weather-map accompanies the atmospheric processes.

In the context of the aforesaid “FIRST ACT OF INTUITIONISM” by Brouwer, a comment by Martin-Löf from 2007 was quite surprising. More than two decades after his attempts to oppose the intuitions contained in the “FIRST ACT OF INTUITIONISM” with his constructive type theory, he wrote an article entitled “The Hilbert-Brouwer controversy resolved?” [523] in which he says:

> if you take Brouwers view that the mathematical objects are mental constructions, or thought constructions, then there arises immediately the problem: how do we get to know them? and not much self-reflection is needed to see how we do get to know them. After all, there is no mathematical zoo to which we can be taken in order to have them displayed to us without any linguistic mediation: we go to listen to some lectures or to read some books or articles, which means that we are dependent on language and symbols from the very start. <u>So this view of Brouwers is hardly credible any longer because of the developments in the philosophy of language during the last century</u>, as Dummett has emphasized for the first time in his paper “The philosophical basis of intuitionistic logic” (Dummett 1975). But there is a kind of mediating position, namely that the mathematical objects are not just purely formal sign configurations: they are meaningful sign configurations, and that is what gives them properties which do not come from their combinatorial nature, but come from the meaning with which they are endowed, as beautifully stated by Gödel in the beginning of his “Dialectica” paper (Gödel 1958). It is this view of the mathematical objects, mediating as it is between Hilbert and Brouwer, which has turned out to be the most credible one to my mind: at least it is the one which underlies all of my own work from 1974 until the present day.

Most probably, many specialists of contemporary linguistics might feel somewhat puzzled by the above quote, especially by the phrase:

> So this view of Brouwers is hardly credible any longer because of the developments in the philosophy of language during the last century.

Interestingly, many prominent mathematicians expressed, views similar in style as those by Albert Einstein. Some of them went even further, arguing that the creation of mathematical theorems in general was not a deductive process. For example, we may quote Paul Halmos, who gave a lecture on Mathematics as a creative art in Edinburgh on May 7, 1973. The lecture was published in [185][1]:

> Mathematics - this may surprise or even shock you - is never deductive in its creation. The mathematician at work makes vague guesses, visualises broad generalisations, and jumps to unwarranted conclusions. He arranges and rearranges his ideas, and he becomes convinced of their truth long before he can write down a logical proof. The conviction is not likely to come early-it usually comes after many attempts, many failures, many discouragements and many false starts. It often happens that months of work result in the proof that the method of attack they were based on cannot possibly work, and the process of guessing, visualising, and conclusion-jumping begins again. A reformulation is needed-and this too may surprise you-more experimental work is needed. To be sure, by 'experimental work' I do not mean test tubes and cyclotrons. I mean thought-experiments. When a mathematician wants to prove a theorem about an infinite-dimensional Hilbert space, he examines its finite-dimensional analogue, he looks in detail at the 2- and 3-dimensional cases, he often tries out a particular numerical case, and he hopes that he will gain thereby an insight that pure definition-juggling has not yielded. The deductive stage, writing the result down, and writing down its rigorous proof are relatively trivial once the real insight arrives; it is more like the draughtsman's work, not the architect's.

Certainly, when quoting different opinions on the relationship between thinking and language, one should not ignore the views of L.Wittgenstein on this subject, especially that his views have a large impact on the next part of this book. In a sense, one may say that the philosophy of Wittgenstein was heavily based on the evolution of his views regarding the relationship between language, thought and reality.

Naturally, in order to present a concise summary of his views, we have to make some simplification. A very good source of such a concise overview can be found in [166]. In the first place, we should look at the concept of language as a "rule-guided activity," which is sometimes referred to as language games [166] (p. 193):

LANGUAGE IS A RULE-GUIDED ACTIVITY

a. *Like a game, language has constitutive rules, namely those of GRAMMAR. Unlike strategic rules, these do not determine what move/utterance will bring success, but rather what is correct or makes sense and thereby, define the game/language,*
b. *The meaning of a word is not an object it stands for, but is determined by the rules governing its operation (LWL 43-5, 59; AWL 3, 30, 44-8, 120, 151; PG 59; see MEANING-BODY). We learn the meaning of words by learning how to use them, just as we learn how to play chess, not by associating the pieces with objects, but by learning how they can be moved (cf. USE),*

[1] http://www-history.mcs.st-and.ac.uk/Extras/Creative_art.html.

c. *A proposition is a move or operation in the game of language; it would be meaningless without the system of which it is a part. Its sense is its role in the unfolding linguistic activity (PI 23, 199, 421; PG 130, 172; BB 42). As in the case of games, what moves are primitive language-games. This approach dominates the Brown Book, which discusses a string of fictional language-games, without providing any philosophical stage-setting, or weaving them into any line of argument. Mercifully, this monolithic language-game method had receded by the time of Philosophical Investigations. Another strategy is to use language-games as part of a reductio ad absurdum argument. This constructs language-games which correspond to the understanding of certain concepts, underlying a certain philosophical theory, and points out to the contrast with our actual language-games and concepts. For example, a language-game is set up in which 'knowing' and 'understanding' are used to refer to states of consciousness with genuine duration, in order to point out that we do not use them in this way.*

When it comes to the concept of language, it is described more or less as follows [166] (p. 216): *He treats 'thinking' as a 'widely ramified concept', and discusses four major employments (Z 110-12, 122; RPP LI 194, 216)*:

a. *thinking about or meaning something;*
b. *reflecting on a problem;*
c. *believing or opining that p;*
d. *having occurrent thoughts which cross one's mind at a particular moment.*

None of them consist in physical or mental processes, either words or images crossing one's mind, since such goings-on are neither necessary nor sufficient [166] (p. 359).

The importance of the concept of a language and thought process put forward by Wittgenstein enabled him to establish key mutual links between these concepts [166] (p. 361):

> There are essential links between thought and language, but they do not require any actual inner vocalization. For one thing, we identify thoughts/ beliefs by identifying their linguistic expressions (see BB 4-5, 161; PI501- 2; MS108 237). The answer to the question 'What are you thinking?' is not a description of an inner process, but an expression of my thoughts in words (e.g., 'I think that it will rain'). If I am challenged by a Platonist or mentalist to express the thought behind that utterance, I do not re-examine some inner process to see whether I can describe it better. Instead, I paraphrase my utterance into other symbols. Consequently, language is not just the only, if distorting, expression of thought, as Frege had it (Posthumous 225, 269-70), it is its ultimate expression. Equally, it is the expression of thoughts which allows one to speak of their having constituents, as Fregeans do.
>
> [...] The second essential link between thought and language is that the capacity for having thoughts or beliefs (c) requires the capacity to manipulate symbols, not because unexpressed thoughts must be in a language, but because the expression of thoughts must be. The reason is that ascribing thoughts makes sense only in cases where we have criteria for identifying thoughts. Something must count as thinking that p rather than that q. This means that thoughts, although they need not actually be expressed, must be capable of being expressed. And only a restricted range of thoughts can be expressed in non-linguistic behaviour. A dog can think that its master is at the door, but not that its master will return in a week's time. For it could not display such a thought in its behaviour (PI 344, 376-82,

650, II 174; Z 518-20). Equally, we can ascribe thinking to, for example, chimpanzees only because of their problem-solving activities.

The above quotes which show how Wittgenstein understands the relationship between the concepts of language and thinking is an important starting point to analyze the model of interactive calculations, proposed in this book, for implementing the processes connected with simulated thinking and simulated use of language as a game.

The approach by Wittgenstein laid the foundations for the creation of a central concept of the book, namely the concept of c-granules. Therefore, in the next Chapters of the book, we will discuss these issues more thoroughly for a better presentation of the computational models proposed by us and, in particular, a better understanding of the concept of c-granules. Another highly important concept in our research is the further development of the concept of judgment.

Chapter 26
Physical World and Uncertainty as Parts of the "Essence of Human Language"

As far as the laws of mathematics refer to reality, they are not certain;
and as far as they are certain, they do not refer to reality.
Albert Einstein [125]

One might think that great rational minds can easily handle the problem of appropriate interpretation of a language, expressing the achievements of modern science concerning the world around us. All in all, we constantly witness a rapid and spectacular progress in science and technology. Our horizons of scientific knowledge about the world are systematically expanding. It turns out that, in general, the great minds of science tend to argue that scientific knowledge is not a certain knowledge. In some way, those minds referred to the foundation of the modern civilization, which is a paraphrase of a statement by Socrates, stating that true knowledge starts from the acknowledgement of ignorance ("I know that I know nothing"). For illustration, let' us cite the examples of statements made by such 20th-century scientist as Richard Feynman:

> If you thought that science was certain - well, that is just an error on your part.
>
> Richard Feynman [139]

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_26

> If you think you understand quantum theory, you don't understand quantum theory!
>
> Richard Feynman[1]

The above quotes indicate that we cannot be sure of the validity of propositions expressed in a scientific language in the real world. The statement by Ludwig Wittgenstein on a larger range of uncertainty within propositions expressed in a natural language, in which he regarded the meaning of an expression as its use within a language, is even more emphatic. The idea is contained in an aphorism, which constituted the principal thought in his "Philosophical Investigations" [539]:

> Don't ask for the meaning, ask for the use!

Wittgenstein began his "Philosophical Investigations" with an analysis of a text, which, in his opinion, constituted a "picture of the essence of human language." This text came from St. Augustine of Hippo, the author of "Confessiones" (written in Latin between AD 397 and AD 398).

> 1.8.13 ... This I remember; and have since observed how I learned to speak. It was not that my elders taught me words (as, soon after, other learning) in any set method; but I, longing by cries and broken accents and various motions of my limbs to express my thoughts, that so I might have my will, and yet unable to express all I willed, or to whom I willed, did myself, by the understanding which Thou, my God, gavest me, practise the sounds in my memory. When they named any thing, and as they spoke turned towards it, I saw and remembered that they called what they would point out by the name they uttered. And that they meant this thing and no other was plain from the motion of their body, the natural language, as it were, of all nations, expressed by the countenance, glances of the eye, gestures of the limbs, and tones of the voice, indicating the affections of the mind, as it pursues, possesses, rejects, or shuns. And thus by constantly hearing words, as they occurred in various sentences, I collected gradually for what they stood; and having broken in my mouth to these signs, I thereby gave utterance to my will. Thus I exchanged with those about me these current signs of our wills, and so launched deeper into the stormy intercourse of human life, yet depending on parental authority and the beck of elders.
>
> St. Augustine [13]

This intuitively means that the meaning of words is defined by mutual interactions between human beings and other physical objects, associated with the context in which these words are used. More specifically, the use of language, in the interpretation of Wittgenstein, is a part of the so-called language games according to which the same word, when used in different contexts, can have many different meanings. For instance, in complex situations, the tone of our statements and our body language can completely alter the meaning of the spoken words. To understand complex language games, it is necessary to start considering them in simple situations. A classic example of such a game is a simple "builder's language," introduced in the first paragraphs of the aforementioned "Philosophical Investigations" [539]:

---

[1] https://en.wikiquote.org/wiki/Talk:Richard_Feynman#.22If_you_think_you_understand_quantum_mechanics.2C_you_don.27t_understand_quantum_mechanics..22.

> § 2 Let us imagine a language for which the description given by Augustine is right. The language is meant to serve for communication between a builder A and an assistant B. A is building with buildingstones: there are blocks, pillars, slabs and beams. B has to pass the stones, and that in the order in which A needs them. For this purpose they use a language consisting of the words "block", "pillar", "slab", "beam." A calls them out; - B brings the stone which he has learnt to bring at such and- such a call. Conceive this as a complete primitive language.

> § 3 Augustine, we might say, does describe a system of communication; only not everything that we call language is this system. And one has to say this in many cases where the question arises "Is this an appropriate description or not?" The answer is: "Yes, it is appropriate, but only for this narrowly circumscribed region, not for the whole of what you were claiming to describe." It is as if someone were to say: "A game consists in moving objects about on a surface according to certain rules ..." - and we replied: You seem to be thinking of board games, but there are others. You can make your definition correct by expressly restricting it to those games.

> § 4 Imagine a script in which the letters were used to stand for sounds, and also as signs of emphasis and punctuation. (A script can be conceived as a language for describing sound-patterns.) Now imagine someone interpreting that script as if there were simply a correspondence of letters to sounds and as if the letters had not also completely different functions. Augustine's conception of language is like such an over-simple conception of the script.

Naturally, from the viewpoint of our applications, it is equally important for novice builders, or even children, to understand the learning process of such simple language. Wittgenstein strongly emphasizes that this process does not rely on the "rules of the game" as much as on a proper training ("the teaching of language is not explanation, but training"). In the process of teaching a language, learning the links between words and specific physical situations proves especially crucial:

> § 6 We could imagine that the language of § 2 was the whole language of A and B; even the whole language of a tribe. The children are brought up to perform these actions, to use these words as they do so, and to react in this way to the words of others. An important part of the training will consist in the teacher's pointing to the objects, directing the child's attention to them, and at the same time uttering a word; for instance, the word "slab" as he points to that shape. (I do not want to call this "ostensive definition", because the child cannot as yet ask what the name is. I will call it "ostensive teaching of words." - I say that it will form an important part of the training, because it is so with human beings; not because it could not be imagined otherwise.) This ostensive teaching of words can be said to establish an association between the word and the thing. But what does this mean? Well, it may mean various things; but one very likely thinks first of all that a picture of the object comes before the child's mind when it hears the word. But now, if this does happen is it the purpose of the word? Yes, it may be the purpose. I can imagine such a use of words (of series of sounds). (Uttering a word is like striking a note on the keyboard of the imagination.) But in the language of § 2 it is not the purpose of the words to evoke images. (It may, of course, be discovered that that helps to attain the actual purpose.)

The above-mentioned considerations by Wittgenstein on the "picture of the essence of human language," based on the works by St. Augustine, may be interpreted in such a way that thinking and therewith related rules of using a language by human beings can be represented by interactions. These interactions may then generate or use links and associations between mental phenomena and physical phenomena, occurring in the world that encompasses human beings. Some of these mental phenomena can stimulate a human process by which certain sentences of a natural language are expressed. The meaning of these expressions can be acquired by learning the rules for the use of these expressions or by creating new rules and motivating others to use them. Ever since the Antiquity (especially since the time of Plato), we have had philosophical discussions about the relationship between the physical world and that in which human beings individually experience mental phenomena. Here, we have a whole range of views, from such extreme approaches as solipsism, which denies the existence of anything other than the human individual mental phenomena, through those which posited no mental phenomena apart from the physical world. Nevertheless, it is not our intention to settle these discussions.

We assume that from the point of view of the conceptual construction of human thinking models, it is convenient to admit that there exist two worlds (that is, the physical world and the world of mental phenomena). However, it is worth emphasizing that in our further considerations, an assumption that there exists at least one world, i.e., the physical world (i.e., the world in which any hard_suit) is embedded, proves to be central. On the other hand, the world of mental phenomena and the world of relationships among these mental phenomena can be treated as encoded (interpreted) in the physical world. This can be intuitively understood in such a way that for "smart robots" hardware will suffice and the expressions of software, which control such a robot, can be encoded in its hardware. Such an interpretation requires an establishment of links between the words and actions of robots. This establishment was also a key mechanism in the reasoning by Wittgenstein cited above ("establish an association between the word and the thing").

Further expanding the interpretation of Wittgenstein's metaphor, proposed in the work by St. Augustine, we can distinguish the following three layers (cf. Sect. 28.2 and Chaps. 29–32) in the human thinking process (or, more generally, in thinking processes simulated by agents):

1. **Soft_suit**—a layer for the representation of entities and phenomena that make up the mental phenomena related to human thinking. They include encoded phenomena such as: language of agent's thinking; rules for the use of languages (including communication); agent's "emotions"; methods for influencing and processing agent's mental states; agent's "imagination"; imagination-manipulating methods such as: selection, abstraction, refinement, finding analogies and similarities, remembering, prediction, associating and processing scenes perceived in the environment and those occurring in an agent's mind, as well as recognition, memorization and association of important facts and changes concerning these scenes. A particularly important feature of the mind is the ability to create new scenes in the agent's imagination for the purpose of "what-if analysis," to anticipate potentially possible consequences of interaction plans. An agent can use imagination to verify possible consequences of her/his actions against the actually observed events in the environment (for example, in order to explain the perceived phenomena and/or to plan and/or to improve the agent's decision-making process).

2. **Link_suit**—layer for the representation of relationships (links) between the important phenomena in the physical world and mental phenomena. For example, an agent talking about a specific tree, observed in the physical world, produces a certain link between this particular tree and the ideas and/or thoughts/emotions of the agent concerning this tree. An agent navigating in an environment generally uses a model of that environment in the form of constructions, available to this agent in her/his mind to represent parts of the modelled reality. This type of model generally consists of multiple link packages (between the agent's ideas/thoughts and the objects/phenomena in the physical world) and links that represent possible agent's actions upon the environment. In the case of human beings, these models often define a relation that we might intuitively call maps (geographic, astronomical, geographical, car construction, aircraft construction, …). An example of such maps is a relation of "being a part" (i.e., "what is part of what" in terms of mereology). An agent equipped with mereology, taxonomy, or, more generally, an ontology of the world, crafts the package links to objects/physical phenomena that are currently important, with respect to the current state of the agent's changing hierarchy of needs. The basis for the functioning of the agent within an environment includes her/his actions, aiming at controlling a process (or individual actions) in that environment and. in particular, the processes for measuring and recording the value of attribute packages available to the agent. Such a control requires an effective use of appropriate links between the objects/phenomena in the physical world and agent's thoughts. In other words, if an agent is a "dreamer," it can create interaction plans which may prove impossible to be connected (linked) with the actual phenomena in the physical environment. In such a case, the chance of their implementation is negligible. Link_suit also includes the links between actual configurations of physical objects/phenomena and basic abstract entities, comprising of Soft_suit and link_suit.
3. **Hard_suit**—physical layer covering a related part of the environment which is accessible to the agent by the links (from the link_suit).The agent conducts particular processes, which has a direct influence on the thinking of this agent. Intuitively, it can be the physical body of an agent (in particular the brain) and the environment potentially capable of exerting a direct influence on it. In this layer, there are physical instances of objects and phenomena performing the functions that make up the two previous layers (Soft_suit and Link_suit). What is particularly important is that both the calculations run and/or controlled by an agent; any interactive processes that occur in the physical part of an agent and in the physical environment are implemented in this suit.

One of the basic mechanisms to generate, develop, and modify the language games by Wittgenstein are interactive processes that occur in the physical world between the players who generally obey the rules of the game, which are not fully known to them. In particular, this means that it is difficult for the agents to deal collectively with private internal sensations and ideas of one of the agents. An experience in the physical world can only be recorded by observing physical phenomena that encode the psyche of such an agent (e.g., monitoring by lie detectors). Nonetheless, we cannot be sure that an agent herself/himself has not altered in any way her/his own codes that represent her/his internal states. The difficulty of establishing the rules for using words related to the phenomena, with no grounding in the physical world, makes it hard to learn the current rules for using a given language. What is worse, as indicated by Wittgenstein, human beings often change such rules in their common parlance, which causes additional difficulties in communication between them. This is particularly visible in the convoluted discussions between different groups of philosophers, especially those dealing with metaphysics. Wittgenstein even believes that metaphysics accumulates the intricacies of a language to such an extent that, as a consequence, it poses meaningless questions.

In his assessment, these are not real problems, but misunderstandings that arise from the abuse of language, whose rules of usage must relate primarily to a possibly specific, observed through the links, physical world. The language of human beings is formed primarily through the establishment of common rules of usage, which facilitates the descriptions of states of commonly observed physical phenomena. In a language, we can express rules of conduct, which may enable the execution of interactive processes in order to confirm, challenge or establish some of the physical properties of the physical world. In this context, the common understanding of the meaning of a language, within some societies of human beings (agents), can be treated as a common application of a protocol (conventions) for the interpretation of rules of using a language. Communication understood in such a way is based on the observation of specific physical phenomena caused by the "sender" agents to the "receiver" agents. These phenomena are interpreted in accordance with the preestablished rules for the interpretation of a given language. Typically, these apriori established rules rely on the establishment of appropriate links between specific objects or physical phenomena and the rules of usage of other physical phenomena (which represent the use of a language). It is worth noting that establishing links between the links describing the mutual relationships among unambiguous expressions (e.g., between an empty set, a set containing an empty set, etc.), and links among physical objects (e.g., in terms of some complex vague concept) is no longer clear.

Therefore, we may experience huge problems with relationships between the world of ideal mathematical models and the real world. This was indicated by both Wittgenstein in "Philosophical Investigations" [539] as well as by Einstein in his quotation about the separation of the mathematical laws and laws of the physical world [125]. Language usage rules, especially the mechanisms for building links between concepts and objects, in these two worlds are different. For example, in the world of mathematical entities, links between concepts can be established in an unambiguous way, whereas in the physical world, this is not always possible and required at all.

In order to better explain these differences, let us imagine a computing model, created on the basis of abstract mathematical entities, such as the Turing machines, and computing models formed by people on computers. Of course, people and physical computers are not able to carry out computations on a theoretical Turing machine, if one consider such a trivial reason as the fact that there is no unambiguous "link" between a Turing machine and the physical world. As a consequence, a Turing machine cannot directly manipulate physical objects. However, there exists no inverse relation, i.e., a Turing machine in theory (but not in the physical world) can complete a computation without subjecting it to real (rather than artificially enforced) laws and unpredictable events, which, in contrary, restrict the freedom of an action in the physical world. It is worth noting that these simple observations have fairly deep implications on the scope of application of both models.

This observation can interpreted from the point of view of CSE practice as follows. Models of computation based on a Turing machine and computing models carried out in the physical world are quite different (the negation of the Church–Turing postulate). This is due to the fact that computing models carried out in the physical world are subject to interaction, which can be only partially controlled. The world around us is highly complex and open, so we are not able to anticipate everything. Some unforeseen interaction which is beyond our perception can always happen.

Nevertheless, it is worth noting that in accordance with our understanding of (or to paraphrase Feynman, "misunderstanding") quantum mechanics, we live in a world of an indeterministic nature. Turing machines carry out calculations in a world that is not so much a physical world, but one that is available to our imagination. This world is closed (meaning that theoretically nothing will disturb the calculations carried out in this world or that we can artificially generate streams of interferences that are de facto controlled) and largely deterministic and may exist for an infinitely long period of time. This is the world of the timeless Cantorian ontology, on which modern mathematics is founded and where the actual physical world has many mutually contradicting features. These features consequently lead to different models of computation.

To recapitulate, in order to gain a better perceptive on the generation, use, and development of mechanisms determining the rules for processing concepts (concept computation models) in a natural language, we must find tools beyond the ontology of modern mathematics, which is based on set theory idealized by Cantor. These tools must indeed embed the new concept computation models within the surrounding physical reality and be based on the assumption of an "open" world. At this point, it is also worth mentioning that many researchers on computation models for AI have reached similar conclusions. Let us conclude with the quotation by Vladimir Vapnik [527] (Epilogue, p. 721) about the statistical learning (as an important part of AI), who says:

> Constructing the physical part of the theory and unifying it with the mathematical part should be considered as one of the main goals of statistical learning theory.

Part VI
WisTech Approach to Models of Mind: Preliminaries

Chapter 27
Preliminaries of IGrC in the Context of Agent Architecture

I cannot state strongly enough my conviction that the preoccupation with Consistency, so valuable for Mathematical Logic, has been incredibly destructive to those working on models of mind. At the popular level it has produced a weird conception of the potential capabilities of machines in general. At the "'logical"' level it has blocked efforts to represent ordinary knowledge, by presenting an unreachable image of a corpus of context-free "truths" that can stand separately by themselves. This obsession has kept us from seeing that thinking begins with defective networks that are slowly (if ever) refined and updated.

Marvin Minsky [339]

27.1 Agents, Interactions, Perceiving Glasses, and Complex Windows

In the literature of AI, there are many approaches to the architecture and functionality of the concept of an agent. A particularly common approach can be found in the classic, introductory book on AI [422]:

> [...] a model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

This approach can be illustrated in Fig. 27.1.

In the book quoted above, a great emphasis was given on an important feature of agents, i.e., their ability to learn. On page 55 in [422], one can find a general structure, illustrating the process of learning by an agent (cf. Fig. 27.2).

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_27

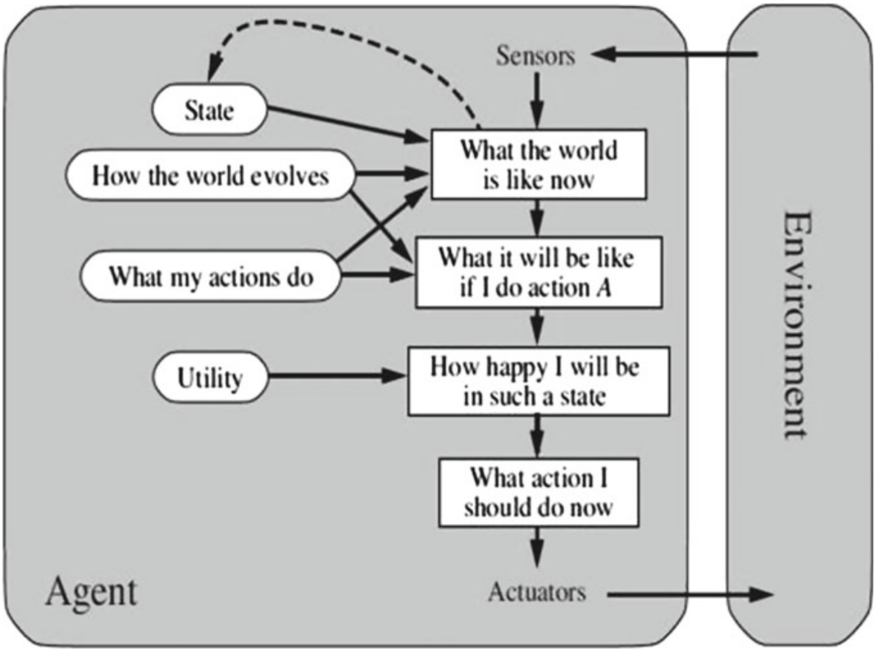


Fig. 27.1 Agent architecture [422]

These rather general schemes presenting the architecture of an agent can be implemented in numerous different ways.

For instance, one can imagine that each white box in this figure is implemented by means of an interactive information system [467, 469]. In such situations, agents perceive the environment and internal states, construct plans of actions, choose between alternative actions to be performed, make decisions whether to activate or deactivate a given action in the face of the dynamically changing environment and whether to perform specific actions (by performing these actions, agents influence the environment). Therefore, interactive information systems contain the attributes of special kinds, namely *perception attributes*, including *sensory attributes* and *action attributes*.

The use of the concept of interactive information systems for such implementations is quite transparent in the case of relatively simple agent architectures. Of course, for such relatively simple implementations, one might use many other well-known AI approaches, designed for dealing with imperfect information (e.g., neural nets, fuzzy logic rule-based systems, genetic algorithms, rough sets, decision trees). All of these methods are applicable to agents with a simple functionality, designed to carry out tasks in reasonably well recognized and stable environments.

Examples of such agents include: drone, autonomous vacuum cleaners, waiters, autonomous software agents, that monitor the risks in corporate computer networks with clear corporate security policies, or players in a variety of relatively simple

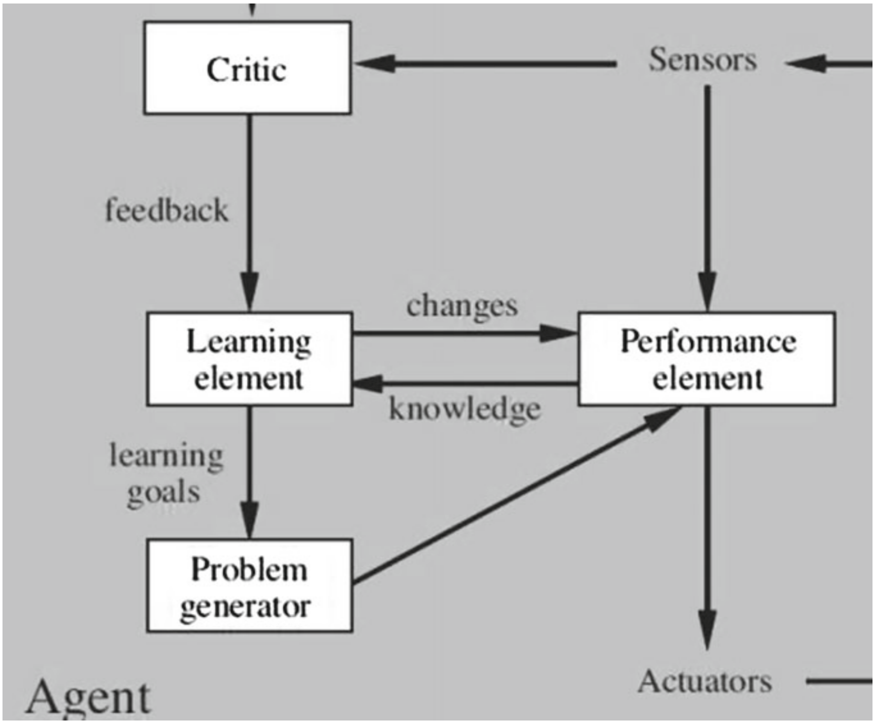


Fig. 27.2 Structure for learning agent

games (checkers, chess). The situation rapidly changes when it comes to the quality of performance of autonomous agents in solving non-trivial problems in unknown and fast-changing environments.

Artificial agents still have a long way to go, in terms of the adaptive survivability in a changing environment, in order to imitate even the simplest forms of life, like a bacterial colony.

Many scientists are especially impressed with the adaptability and intelligence observed in colonies of ants, bees, and many other insects that cooperate with each other [420]. According to a great number of these scientists, the already existing approaches to modeling thought processes are too low-level to be successfully used in constructing artificial agents, which can collaborate on solving complex non-trivial problems in unknown and changing environments.

Hence, there is a demand for the development and verification of a new approach to modeling cooperation and thought processes that will improve the quality of computing models, supporting individual or collective solving of non-trivial problems by agents.

The approach presented in this book uses the concepts of the WisTech computing models, based on the models of interactive c-granular computing. The key

concepts of the approach include such important notions of WisTech as: "hunk," "c-granule," "hierarchy of needs," "adaptive judgment," "efficient interaction plans."

Most of the WisTech concepts were discovered during joint research with Professor Andrzej Skowron. Some of the results are presented in the following papers: [248–250, 252]. The approach proposed as a result of the research is a continuation of the tradition of the Pawlak-Rasiowa School of AI [245]. Its conceptual core consists of the extended notion of (most notably by the physical components of the interaction with the agent's environment of activities) Pawlak's information systems and techniques for approximate reasoning from the perspective of rough sets [463].

Unfortunately, the WisTech concepts (especially the concept of the "c-granule") have proved to be conceptually complex. In the following chapters of the book, we will gradually clarify the meaning and use of such concepts. Initially, we will present very simple intuitions, examples, and links to some relevant concepts of computer science and AI. In Chap. 32, we will summarize ways of applying the intuitions related to the afore-mentioned term of a "c-granule" in the form of FPW. It should be underlined that FPW postulates require further clarification and more detailed development before they are applied in real-life projects. In general, a significant part of the book (especially within the scope concerning WisTech) indicates potential directions for the interpretation, development, and explanation of FPW.

In our model, we assume that **agents are parts of a physical environment which systematically change their states over time**. An agent does not have access to full information about the environment and, what is of particular importance, does not possess complete information regarding the configurations of her/his surrounding hunks. An agent can only receive signals from the environment through her/his sensors and initiate some interactions by actuators. Some of these signals are converted to values of the agent's attributes.

Through these attributes, an agent can distinguish objects and physical phenomena within a certain spatiotemporal space. Following the proposal of Heller in [201], any such physical object existing in spatiotemporal space is called a hunk (i.e., portion of matter, see Sect. 3.8).

An agent may perceive hunks by recording in her/his memory the results of aggregating the consequences of interactions between her/his sensors/actuators and the environment, in which hunks are located. The results of such an aggregation, as perceived by an agent, are known as the *properties of hunks* or *interaction properties*.

An agent may represent the properties of hunks through: expressions of her/his language, sets of expressions, theories (e.g., theories as understood by [26]), sets of dependencies determining cause-effect relations, and models of processes.

In actual applications, the properties of hunks are usually the results of very complex computational constructions. For example, they may resemble a network of cooperating hierarchical structures (e.g., neural network or network of c-granules defining a new c-granule). Classical examples of such hierarchical structures of computational constructions include the following:

1. Aggregation of data aimed at determining the property of an average height (or other statistical feature) among a given human population through the calculation of an average on the basis of determining the height of each representative (or a selected sample) of a population.
2. Mechanisms for the aggregation of pixels on a TV screen, enabling the identification and processing of the properties of hunks (recorded by a TV camera).

It is worth emphasizing that the concept of a hunk has a spatiotemporal character. As a consequence, the properties of hunks may be very complex. For example, these may include various complex models of such spatiotemporal hunks.

The spatiotemporal character of hunks applies both to the properties of physical objects perceived by an agent and to the processes related to the hunks.

The hunks perceived by an agent **may be independently perceived and investigated by numerous other agents** equipped with sensors, actuators, and aggregations techniques whose manner of operation is quite similar.

An agent may interpret various properties of hunks. To make a great generalization, we can distinguish two types of such properties:

1. **Dynamic properties.** These are those properties of hunks perceived by agents which cannot be perceived at very short intervals of time. They are formed by aggregation of properties at numerous moments of the agent's time, within some period of time or some dimension of spacetime.
   Intuitively, these are the properties of hunks which an agent cannot perceive in a photograph. However, they can be perceived in a movie.
2. **Static properties.** These are the properties of hunks perceived by agents which are not dynamic. This means that these properties can be perceived within such a short moment of time which an agent can distinguish (the borderline being a moment in the agent's time).
   Intuitively, these are the properties of hunks which an agent can perceive in a photograph.

To recapitulate, an agent may perceive the H hunk (henceforth referred to as H) through perceiving her/his static and/or dynamic properties and by aggregating the sets of properties of H.

To be more specific, an agent may perceive H through the aggregation of the following properties:

I-1 properties inherent to the mereological structure of H (i.e., agent perceives such relations as: being a part of H component, being part of the parts, etc.),
I-2 properties inherent to the parts of H at the level of its mereological structure (including the properties of parts that are parts of its components, etc.),
I-3 properties of relations between the sets of H parts,
I-4 properties of H obtained by aggregating the properties of H parts.

We may say that the property of H (formed thanks to its mereological structure mentioned above; see points I-1–I-4), is also the *property of interaction between various hunks that are parts* of H or, in short, *the interaction property*. In such a situation, H is *commonly referred* to as the *interaction scene*, whereas the hunks which constitute its parts, appearing under the aforesaid conditions (I-1–I-4), are known as the *participants of interaction* or the **interaction hunks** (i.e., ***the hunks which are in interaction***).

Some examples of interaction hunks include: planetary motion around the Sun, electron motion around the nucleus of an atom, car traffic, objects lying on a table, building constructions, as well as theatrical performances.

To put it more simply, **interaction properties are formed in the process of aggregating the parts of a given hunk and the relations between these parts. However, the *interactions* themselves constitute all the processes of physical objects which a given agent may attempt to perceive using interaction properties.**

Such an interpretation of an interaction property (or the interaction alone) may be of two-fold nature: static (e.g., a ship staying afloat) and dynamic (e.g., a rock falling to Earth).

When H is the interaction scene, at various moments of time its parts that are in interaction in the interaction scene, determined by this hunk, may be found in states that are distinguished by an agent through her/his perception capability. In such a situation, we may say that an agent perceives various *hunk configurations* of H.

Of course, agents—due to the restrictions on their sensors and actuators—are generally not capable of distinguishing all hunk configurations. Thus, the ability of an agent to distinguish hunk configurations is subject to the limitations of her/his sensors/actuators.

An agent strives to overcome these limitations by creating various new aggregations of signals from the sensors or actuators, and acquiring frequently sophisticated signals from the environment through complex and pre-scheduled interactions with this environment. For this purpose, an agent may use complex *generalized attributes*, e.g., classifiers and optimizers (which support the process of control of an agent during the construction of optimal solutions to various problems, e.g., the construction of "optimal" interaction plans).

The approach presented in this book uses the concept of the agent's *c-granule* as a starting point for constructing the proposals of such models.

Roughly speaking, one can treat a *c-granule* as a generalized attribute (or classifier/optimizer) consisting of physical networks of hunk configurations, and/or sensors/actuators, linking the agent's control and memory with some interactions in the physical environment.

Apart from the entities of the physical world surrounding it, an agent may retrieve previously stored hunk images (hunk configurations), represented by appropriate attributes or, more generally, "c-granules."

An agent can also use her/his attributes (or more generally "c-granules") to create models of attribute configuration images, which has been never encountered by her/him in the external physical reality. In other words, we assume that an agent might have her/his own metaphorical "mental world" in which she/he collects representations of the physical world as perceived by her/him in the form of c-granules, as well as c-granules representing her/his mental world (constructed on the basis of observations of available attribute value configurations).

Entities representing some meaning to an agent, which can be recognized and manipulated by this agent in her/his "world of mental phenomena," are called *infogranules*. Intuitively speaking, in her/his "world mental phenomena," an agent can use some irreducible components (*elementary infogranules*) whose semantic meanings are known to her/him and aggregate them into complex infogranules.

Below are some of the approaches to defining the concept of an infogranule:

1. Descriptors of information systems [391] are elementary infogranules and their Boolean combinations can be *complex infogranules*.
2. In a more general situation, infogranules can be interpreted as a representation (in the world of the mental phenomena of an agent) of the hunk configurations perceived by an agent in the physical world around it.
3. In some cases, infogranules can also be interpreted as a distinguished representation (in the world of the mental phenomena of an agent) of other infogranules perceived by an agent.

In a sense, the physical world observed by an agent is represented as an "image" obtained by this agent after a filtration with *metaphorical glasses*, through which she/he perceives the physical world. These glasses receive signals from the physical world. More specifically, these signals reach the sensors of an agent, where they are converted to attribute values available to an agent.

Thus, a picture is created which can be seen by an agent from the other side of the *perceiving glasses*. If the glasses can transfer only the black and white signals, an agent will not see the true colors. Configurations of attribute values, perceived by an agent through the glasses, can form images recognizable by this agent that are interpreted by her/him as infogranules.

To recognize these images, an agent may use complex supporting attributes. We assume that an agent can store these images and, to a certain extent, retrieve them or their compressed forms. Such retrieved images can be looked at through the *second metaphorical glasses*, that are conventionally called the *mental world perceiving glasses*. These glasses receive signals from the physical "brain" of an agent and her/his world of configurations, i.e., from the hunks that implement the agent's mental world.

Such "mental" hunks are called *m-hunks*. Analogously to the previous case, when it comes to the agent's mental world perceiving, we also assume that the signals from the world of configurations formed in "the brain" of an agent are sent to her/his appropriate sensors, and are therein converted to attribute values available to the agent.

The attribute values processed in such a way are then passed to an agent in the form of images (on the other end of the mental world perceiving glasses). In this case, the picture is understood more broadly than in the case of images generated by the glasses for perceiving the physical world.

In other words, through the mental world perceiving, we "see" such internal states as courage, optimism, pessimism, heat, cold, pleasure, pain, joy, sadness, saturation, hunger, and other phenomena perceptible to the mind (like inner voices, visions, transformation of mathematical expressions, or any other creations of the mental world).

We assume that an agent can simultaneously look through her/his glasses for perceiving the physical world and continue her/his thought process, e.g., leading to discovery of solutions of investigated problems, proofs of theorems or construction of new sensors for space exploration.

For example, she/he can associate the currently observed situation with other situations, previously stored in her/his memory. In addition, an agent observing the physical world can evoke certain images in her/his mental world (e.g., an emotional state of anxiety or joy associated with the currently observed physical phenomenon).

In other words, an agent sees the world through her/his *complex glasses*, consisting of:

1. *Real glasses*—through which an agent perceives the physical world surrounding it (external to her/his mental world). Through these glasses, an agent "sees" the world with her/his physical senses. She/he may also sense sounds, fragrance temperature, pressure, orand taste.
2. *Imaginary glasses*—through which an agent perceives her/his mental world (external to her/his physical world). Here, she/he may also sense her/his own thoughts, emotional states, pleasure, pain, imagination, association, uncertainty, conclusions, etc.

A certain analogy between the complex glasses and complex numbers, which also have their real (real glasses) and imaginary (imaginary glasses) parts is naturally evident.

An agent perceives the world through complex glasses by recording the changing attribute values on her/his mental display, which we refer to as her/his *complex window*. It consists of two parts: the *real window* (hunks observed by an agent in her/his environment, conveyed through her/his real glasses) and the *imaginary window* (m-hunks observed by an agent in her/his environment, conveyed through her/his imaginary glasses). *Agent's attention* supervises the implementation of the *plans of interaction* with the environment.

The "supervision" is subject to the *process of control by an agent*, based on the *adaptive judgment* abilities of an agent, identifying her/his current *prioritized needs*. Depending on the needs, the agent's glasses are focused (using appropriate sensors and actuators) on some parts of the window. At this point, it is worth emphasizing that the significance of controlling the reasoning process on the basis of judgment and fulfilling one's needs was recognized already by Aristotle [10, 475].

The complex window itself can be a network of complex window hierachies. We can propose the following metaphorical equation:

$$complex\_window = real\_window + imaginary\_window. \tag{27.1}$$

Obviously, from the viewpoint of an autonomous agent, any skill that may facilitate her/his survival in a dynamic and unknown environment is useful. In this context, the agent's ability to identify and recognize objects and phenomena of actual interest (in the windows and thus, in the environment), so that it is not driven by any illusion, is highly important.

Hence, the agent should be particularly able to recognize and pinpoint the position of objects or phenomena recorded in her/his windows. Thus, the following equation can be developed:

$$\begin{gathered} complex\_window\_location\_attribute = \\ real\_window\_location\_attribute + imaginary\_window\_location\_attribute. \end{gathered} \tag{27.2}$$

In general, window location attributes can specify the location in the real or imaginary window. In a sense, we can assume that an agent has an internal clock, in which time intervals are considered as specific location attributes. At any time interval sensible to an agent, she/he observes images, referred to as complex pictures. Such a picture has two dimensions as depicted in a metaphorical formula:

$$complex\_picture = real\_picture + imaginary\_picture, \tag{27.3}$$

where the *real picture* is a static image or a dynamic movie, perceived through the real glasses.

Analogously, the imaginary picture is a static image or a movie, perceived through the imaginary glasses. Real picture and imaginary picture consist of different versions of image changes of objects and/or imaginations, along with the locations and time intervals assigned to them. An agent observes and attempts to store the pictures on which it is focused through her/his complex glasses at a given time interval. An agent, while monitoring the external world, may notice a repetition of similar images, along with the recurring objects or phenomena, known as hunks.

Upon directing her/his real glasses at those hunks, an agent may expect to see the same hunks, which identify the location in the physical world. It may happen that some hunks will no longer be present at a given location, but an agent can still recognize it using other hunks. Therefore, to identify a location, extra identifying information is often required.

An agent, using her/his imagination (more precisely, her/his *memory*), can quite freely shape objects and their relative position. While constructing the images in her/his imagination, an agent may or may not associate them with some specific place in the physical world. It is also possible that an agent, while constructing an image in her/his imagination, may attempt to reproduce a picture that resembles, to the greatest extent possible, a physical location that we call a *scene*.

Observing different scenes of interactions in the outer physical world, an agent may assign special pictures to those scenes, which we call scene maps.

We will assume that on these maps, there exist basic objects—anchor points of reference used to determine the location—and descriptors that describe other locations on these maps, using the anchors. Along with the improvement of her/his imagination, an agent watching a given scene may decide to widen the scope of possible modifications to the hunks present in this scene.

In this way, she/he may create generic *scene* maps that resemble the concept of *frames* [339] used in AI. Let us recall that the term was first introduced in this context by Marvin Minsky [339] as a paradigm to understand visual reasoning and natural language processing. Agent imagining frames can help predict the course of a given action.

A good example is driving a car. Scene maps may also be applicable to the agent's world of mental phenomena. Since the ancient times, we have known many examples of visualizing the realm of thoughts, concepts, and emotions with the help of diagrams. A classic example is the work of Aristotle.

Also in modern times, the visualization of the world of mental phenomena plays a substantial role. Terms such as "mind map," "semantic network," etc. are widely popular. An agent equipped with scene maps can navigate more easily and, as a consequence, move around more freely in the physical world, as well as in the world of mental phenomena. In other words, an agent's ability to perform mind-steering actions and navigate in the physical environment requires complex navigation skills, both in the world of mental phenomena as well as in the physical world. Therefore, we can come up with one more metaphorical equation:

$$complex\_navigation = real\_navigation + imaginary\_navigation. \quad (27.4)$$

Complex navigation can be interpreted in such a way that an agent has attributes which support such activities as:

1. Identification of *scene maps* needed by an agent (according to the agent's adaptive judgment).
2. Identification of indicative positions of an agent, expressed in appropriate values for location attributes within a "scene map."
3. Identification of actions recommended for moving around a map in order to plan further actions.
4. Moving around in the environment.

When a map turns out to be insufficient, an agent should be able to identify or create new maps. The creation of new maps is of paramount importance in the case-based reasoning [218].

In such a situation, an agent may use a generic scene map, analyze the phenomena observed in similar scenes, and try to apply the conclusions drawn thereof to the new situation.

An agent controls her/his real glasses by executing appropriate actions in the physical environment. The response to interactions from the environment will also appear on her/his real glasses. Similarly, an agent controls her/his imaginary glasses through the execution of corresponding physical actions, carried out, e.g., by processes occurring in her/his mind and resulting in actions within the agent's world of mental phenomena (remember that mental phenomena are considered here as changes in agent's hunk configurations). Analogously, here the effect of these actions is the result of the environmental response (intuitively, an effect of some response in the mind and the effects of any unforeseen interactions with the environment). It is emphasized that the mere desire to perform an action by an agent (e.g., to look through her/his complex glasses at a direction of interest) does not guarantee that this action will be feasible. It is the environment which determines, through the response of an agent, whether the action is feasible and whether there will be any limitations or disturbances.

An agent observing the world through her/his complex glasses can at the same time perform a compression of the data, which produce complex images in subsequent time intervals. Let us not forget about the fact that an agent observes the world through a limited set of attributes, which, generally, have a fairly narrow range of descriptors. Therefore, the picture produced by the attributes of an agent is in itself a significant compression of the incoming data stream. Next, an agent captures selected features of the image, that often constitute input data to complex hierarchical attributes. Then, depending on her/his abilities, an agent is able to store those complex values of attributes, which she/he finds important, performing a compression on the input data. In this way, she/he produces images in the form of a metaphorical film that we call an "integrated agent's history."

We assume that an agent has the ability to replay in her/his imagination previously stored fragments of her/his "integrated history" to modify them and use them to foresee new situations. For instance, she/he can associate the values of some attributes with potential risk. An agent continuously keeps on interacting with the environment and, depending on her/his ability to assess situations, she/he can identify issues that are to be ad-dressed. Then, she/he attempts to solve these issues, using her/his knowledge, current states of mind, and ability to act. Very often, the solutions to new problems may be modelled based on the solutions to similar problems, stored in the agent's memory, with a necessary adaptation to the current conditions.

The process that encodes the metaphorical film known as the "integrated agent's history" can be, to a certain degree of generality, expressed in a natural way using interactive information systems [467, 469]. Interactive information systems can also be applied to a sample implementation of the metaphorical equations on complex glasses.

The aforesaid functional model of an agent, based on the aforesaid concept of the complex-glasses, does in no way pretend to simulate the functioning of the human mind. This is just like the mechanism for calculating the results of arithmetic operations on real numbers do not pretend to fully model the functions of the human mind. It is merely a model, inspired by a highly simplified approach to perceiving and mental image processing of the human beings. The aim of this simplified approach is to propose further mechanisms to build simple computing models for very simple reasoning about complex vague concepts.

The main concept discussed herein is a model of the "agent's thought" as a vehicle that converts some "thoughts" of an agent, seen through her/his complex glasses, into other "agent's thoughts." These "agent's thoughts" most probably do not have much in common with the actual human reasoning. However, we do not consider it important in this context as our goal is not to describe or explain the processes of human thinking. Our aim is to gain some inspiration from a simplified understanding of the thinking processes to propose a construction of computation models which, once improved, may be used in the future to build autonomous adaptive robots, moving around in the real physical environment and capable of solving simple problems in this environment. Taking into consideration the fact that in the simplified perception of the human thinking processes, the language, in which we think, plays a key role, we shall return to our previous considerations regarding the language.

27.2 Some Preliminaries of IGrC

In Part III of the book, we have explained the meaning and practical importance of the TPGP problem, paying particular attention to some difficulties in the struggle with FP3C, the essential precause of the TPGP problem (Chaps. 12 and 13). It can be concluded that an important aspect of our efforts to minimize the negative consequences of TPGP is the comprehension and skillful application of relevant models of interactive computing on granules, and reasoning about this computing in CSE.

It should be noted that currently, **GrC constitutes a dynamically expanding scientific field** [40, 218, 220, 307, 308, 394, 472, 549].

On the other hand, it is worth remembering that **GrC refers to traditional research on processing and reasoning about conceptual structures. Any progress in the research toward interactive computations on granules could directly initiate a significant leap in the development of our civilization** from many aspects. In some sense, the old principle of *divide and conquer* is the essence of GrC.

To illustrate the importance of GrC, let us recall that even in the ancient times, Aristotle [10, 475], among others, dealt with the mechanisms and forms for processing and reasoning about conceptual structures. This subject continued to be studied in the next centuries. Of particularly importance was the work by Gottfried Wilhelm Leibniz [9, 300, 301]. The research was continued, among others, by George Boole [61, 62]. In the mid-19th century, he proposed an algebraic structure, referred to at the time as the "algebra of thoughts." Today, it is now known as Boolean algebra.

As we remember—it was Boolean algebra that has played and continues to play an important role in the development of our civilization, paving the way for the invention of computers and programming languages.

It is remarkable that Boolean algebra found its most prominent field of application in computers only after about a hundred years after was invented. These algebraic structures are still being used to develop new key technologies. For example, they are particularly intensively used in such traditional applications as:

- Consumer devices with automatic control systems (e.g., cars, planes, ships, mobile phones, washing machines, ovens, lifts).
- Control and management systems for global and local computer networks (e.g., Internet, Intranet, corporate LAN).
- Database management systems.

It should be noted that in addition to the aforementioned, and commonly known, traditional applications, new important fields of application of Boolean algebra continuously emerge. For instance, in the second half of the 20th century, Boolean algebra became the basis for the development of software systems for key AI applications, using approximate Boolean reasoning [359, 389, 442].

To sum up, Leibniz and Boole can be considered pioneers of the modern concept of granulation. Leibniz's idea concerning the interpretation of thoughts and their processing in the form of computing processes, in the metaphorically understood arithmetic of real numbers, was also intensively developed after the advent of Boolean algebra.

Another significant step in this direction was the concept of Lindebaum-Tarski algebra [410]. In this algebra, the concept of thought (or the modern concept of granule) can be interpreted as the equivalence classes of the relationship between logical formulas. Lindenbaum-Tarski algebras are strongly connected with the canonical model of a mathematical theory, which proves the crucial role in mathematical applications.

Canonical models can be obtained, very intuitively, by performing algebraic operations on Lindenbaum-Tarski algebras. Algebraic models for classical predicate calculus, intuitionistic logic, modal logic, Post logic, and many other logics were extensively investigated by Alfred Tarski (mainly for propositional calculus), Helena Rasiowa and Roman Sikorski mainly for predicate calculus, [245, 410]. In the 70s, the Rasiowa-Sikorski algebraic models were generalized to sheaf models, using topos theories [296, 318, 507]. One such [38, 203] is the work by Denis Higgs,[1] who generalized the Rasiowa–Sikorski Boolean models to the case of category theory in 1973. In particular, Higgs defined the notion of Ω-set and established for a complete Boolean algebra B, the equivalence of the topos of B-sets both with the category of sets and maps in the Boolean extension $V(B)$ of the universe of sets and with the category of the canonical set-valued sheaves on B. Later, analogous results were independently obtained by Dana Scott and his students [38, 144].

Practitioners developing AI-related technologies were quite skeptical about the possibilities of practical application of these sophisticated algebraic tools for "processing of thoughts." In particular, their reservation stemmed from the fact that these algebraic tools concern the logic of a single agent, working with crisp concepts of predicate calculus and possessing no mechanisms for interaction and cooperation

[1] https://en.wikipedia.org/wiki/Denis_Higgs.

with other agents. In addition, this single agent usually resorts to "thought processing" mechanisms, which in non-trivial situations is often infeasible by means of modern computers.

These practitioners are convinced that developing applications in the real world requires a different approach. Such an approach should presume that the concepts related to the features of complex dynamic systems are generally vague and thus, any inference about them is doomed to be approximate. This would help human beings to reach a final conclusion with a sufficient degree of certainty in a much shorter time.

A very nice and transparent outline of intuitions, illustrating the distance between the achievements of the modern "crisp" science (e.g., quantum physics) and **the phenomena in the complex world around us** can be found in the book [158].

The above arguments stimulated a growing interest in a better understanding of granulation and its use in facilitating the construction and implementation of AI computer systems. On this basis, a series of competing approaches, attempting to deepen the understanding of the concept of granulation for processing, communication, and comprehension of vague concepts, were developed.

Today, the predominant way of understanding the concept of granulation—especially in the context of AI applications that use vague concepts—refers to the pioneering notion of a linguistic variable, introduced in 1973 by Lofti Zadeh [548] and developed further by him in [546, 549, 551]. Lofti Zadeh presented the following intuition of the concept of *information granulation*, which resulted in the creation of information granules [552]:

> Information granulation can be viewed as a human way of achieving data compression and it plays a key role in implementation of the strategy of divide-and-conquer in human problem-solving.

Also, the concept of **interactive computing is a dynamically expanding area with quite an extensive history** [169]. In [372] the key role of interactive computing in understanding complex systems was duly underlined.

The approach to interactive computing and GrC presented in this book is a continuation of the research known as Interactive Rough Granular Computing (IRGrC), which is presented in the following works: [251, 253, 448, 464–469].

Within this mainstream, we can find such works as [248–250, 252]. These papers are devoted to the concept of c-granules, which are of utmost importance for this book, as well as the basic intuitions of interactive computing on granules.

Given the importance of models of interactive computing on granules and methods of reasoning about these computing properties for this research, in the next chapters, we will present a very concise summary of these concepts. Their presentation and analysis will include the following stages:

E 1. In Chap. 28 entitled **Selected Distinguishing Properties of WisTech IGrC Models**, we shall enumerate the properties of our computing model which distinguish it from other types of computing models (e.g., Turing models of computation). We will provide an intuitive explanation of these properties, expressed in a natural language. At this stage, we shall focus on as simple and intuitive an explanation of the key properties of our computing models as possible, rather than development of a more specific formal conceptual apparatus.

E 2. The descriptions of the distinguishing properties presented in stage E1 (i.e., Chap. 28) are not detailed and clear enough from the perspective of a possible design and further implementation of these models. It turns out that the task of clarifying the conceptual apparatus for a more detailed description of such computing models, is in itself a difficult challenge. Consequently, in Chap. 31 entitled **Designing an Ontology for IGrC Models**, we shall identify the main causes of these difficulties. We will also introduce proposals of the basic objectives and principles for constructing an ontology to support the specification of interactive computing models and methods of reasoning about interactive computations.

E 3. The assumptions made in stage E2 will be illustrated in Chap. 32 entitled **Framework Postulates for WisTech (FPW)**. These postulates specify (set the direction for) the ontology in terms of supporting the implementation of the agent's control as part of (on the basis of) a WisTech computational model.

Chapter 28
Selected Distinguishing Properties of WisTech IGrC Models

28.1 Limiting the IGrC Model to Physical World

The basic assumption that governs our approach posits that **any interaction that is analyzed should be physical, i.e., implemented in the real physical world, by physical objects**. This applies to all the interactions taking place in the environment, interactions between (parts of) agents, and between (parts of) agents and the environment.

This assumption, among others, is understood in such a way that all interactions that are analyzed by us are reduced to physical phenomena.

An agent's interaction is carried out using her/his physical sensors and/or physical actuators. Naturally, an agent in her/his interaction with the environment and/or other agents is confined by her/his sensors and actuators. Somewhat more specifically, the agent's control (to an extent made possible by the *agent's attention*) is responsible for configuring and reading control parameters of her/his sensors and actuators. These sets of control parameters determine a metaphorical "window" to the physical world perceived by an agent. Hence—an agent may perceive only this part of the reality that is available to her/him through the "window(s)." As a result, any interactive computation on granules that is subject to our approach may be performed only by the phenomena and physical objects that are partially recognized and identified by an agent. They are accessible to an agent on the basis of the perception of interactions between physical objects.

Assuming that the IGrC models are limited to the physical world, we should be careful with adopting some of the approaches to the foundations of the "mathematics," "minds," and "Platonic ideas" of the agents which are existing inside the IGrC models.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_28

In the contemporary foundations of classical mathematics, Platonism[1] and formalism[2] are especially popular. However, there are also other approaches to the foundations of mathematics, e.g., intuitionism and constructivism[3] [120, 142, 174, 202, 260, 277, 318, 326, 329, 507, 523, 524] (cf. Chap. 24).

As we have underlined in Chap. 24, according to intuitionism,[4] mathematics is a creation of the mind. In other words, if we assume that computing of mind is a physical process, then mathematics can be also treated as some kind of a physical phenomenon. In consequence, the truth of a mathematical statement can only be conceived through a mental construction processes that prove its validity. The communication between mathematicians only serves as a mean to create the same mental process in different minds. For example, the original Brouwer's intuitionism is based on the agent's awareness of time and the conviction that mathematics is a creation of the free mind, which excludes both Platonism and formalism.

However, due to the widespread presence of "Platonic ideas" in contemporary natural languages, for the sake of convenience (resulting more from historical factors rather than a mere necessity), we may quite freely use the concepts of contemporary classical mathematics.

28.1.1 The Physical Nature of Interactions

The **interactive computations on granules itself is a representation of interactions between physical objects**, occurring in a physical world and perceived by agents. According to the assumptions, interactive computing on granules is carried out using hunks and hunk configurations.

Therefore, the medium of interaction is the physical layer within which the interaction occurs between physical parts that are at the disposal of agents and the surrounding physical environment. To be more precise, **computing is conditioned by interactions between physical objects and is represented by the properties of their spatiotemporal configurations as perceived by an agent**. It should be noted that, in this context, we can treat the interaction as a physical phenomenon occurring between hunks (cf. [201] and Chap. 3). Moreover, some of the properties of interactions discovered by agents will be used temporarily only. In the future, these properties may be substituted by new ones according to the new discovered knowledge in interaction with the environment.

A very trivial example of an interaction may be the interaction between a mouse and a mousetrap. If a trap with a bait is sprung and its trigger mechanism is activated,

---

[1] http://en.wikipedia.org/wiki/Platonism.

[2] https://en.wikipedia.org/wiki/Formalism_(philosophy_of_mathematics).

[3] http://en.wikipedia.org/wiki/Intuitionism,
https://en.wikipedia.org/wiki/Constructivism_(mathematics)..

[4] http://en.wikipedia.org/wiki/Intuitionism.

an inexperienced and hungry mouse, passing by the trap, will be—with this hunk, easily caught in the trap due to the hunk's configuration.

The approach to the interactive computing on granules, presented in this book, leads to the following consequences:

- The computing is performed in a physical world where relevant knowledge, e.g., the laws of physics, apply. This means that our **computing complies with the constraints imposed by the laws of physics**. In the trivial example presented above, it is the laws of physics that enable the mousetrap to trigger in motion at the right time.
  The laws of physics can be the objects of agents' perception and consequently, they can be discovered by agents. These laws may be used, by agents, to construct plans of interaction, designed to meet their needs, and to protect their important values or needs; e.g., in terms of the human perception of reality, the laws of conservation of energy seem highly important. As a result, agents taking actions in such environment will first try to heed their energy needs (including those required for the implementation of such actions).
- The **computing is performed in an open world and the whole process can be disrupted or directed into a completely unknown direction virtually, at any time, due to the unknown and/or unpredictable phenomena**. For example, it is quite possible that for some reason, the mousetrap may fail to trigger properly, and the mouse eats the bait unscathed.

A consequence of the physical nature of interaction is the assumption of an *open world* in which IGrC is performed. This assumption contradicts that of the *closed world*.[5]

As a result, this means that basic computing models used in modern mathematics (i.e., those based on Turing machines) are significantly different from those used for interactive computing on granules. Turing machines "work" in an ideal "closed" world of mathematical entities and their algorithms are not affected by the unexpected inferences with the physical world.

28.1.2 The Physical Nature of Agent's Perceiving

An agent has a limited access to the surrounding physical environment. This access is inhibited by the extent to which an agent has a control to reach the physical world (through actuators and sensors at her/his disposal). The control is gained if the agent's state of knowledge is imperfect and if the data, received by an agent both in the past and in the presence, is uncertain and incomplete. The interaction between agent's sensors and actuators and the surrounding environment enables an agent to form in her/his imagination a representation of the phenomena and/or objects that she/he encounters in the environment, and to construct their interpretations. The

[5] https://en.wikipedia.org/wiki/Closed-world_assumption.

information about phenomena, recorded by an agent immediately after it is recorded by the agent's control, is expressed primarily via the sensory attribute values. Such a description can be called an agent's *elementary sensory description*.

The agent's elementary sensory descriptions can be bounded by different mutual relations that are available to it (e.g., time effect, causal effect). More generally, elementary sensory descriptions can serve as the basis for the formation of hierarchical structures and/or networks (e.g., a graph of a project's important events). Such structures of bounded sensory descriptions, recorded by an agent, are called her/his *historical sensory observations*. To avoid any confusion, we shall refer to them briefly as *observations*. To simplify, we can say that an agent empirically perceives the environment through observation. Once recorded and adequately interpreted, observations form an important part of the agent's knowledge.

The values of sensory attributes may be accordingly aggregated and processed, by the agent's control, into more and more complex properties of observation that are potentially important to an agent. Hence, there may emerge highly complex and vague concepts, such as "warm," "cold," "fast," "hot," "good food," "comfortable environment," and "dangerous situation." This allows for a much richer (e.g., more general, but sufficient to solve the agent's problems and/or to spare the agent with unwanted details) expression for properties of the observation (rather than of the crude sensory attributes). This type of description can be called *interpretation of observations*.

Interpretations of observations may, e.g., express *agent's judgment about observation*, conveyed in the language of agent's needs and values (using hierarchical mechanisms of perception).

Observations important to a given agent, as well as their interpretations (including their judgment), can be stored in the agent's memory (and compressed or granulated if needed). Such stored observations, along with their judgments, may be used as arguments "for" and/or "against" in the future in the process of inductive reasoning when judging new situations that bear satisfactory resemblance to the previously encountered observations.

The concepts of "observation" and "interpretation of observations," along with the concept of interaction, play a fundamental role in this book and the assumptions on the construction of models for the mechanisms used by an agent to perceive the phenomena in the physical world. The approach to understanding the **"agent's perceiving", proposed in this book, is based on the simultaneously executed *PDCA cycles***[6] that communicate with one another. Each of these cycles is implemented in the form of the following phases:

Plan Depending on the current needs of an agent, establishing and/or modifying feasible and possibly effective (according to an agent) interaction plans.

Do Agent's interactive actions in the environment, performed in order to implement the interaction plans.

Check A systematic judgment of realization of those plans of interaction that have been already implemented, both in terms of their expected outcomes and from

[6] https://en.wikipedia.org/wiki/PDCA.

the perspective of the agent's current hierarchy of needs and values (based on the expected and perceived outcomes of the plans).

Act A systematic storage of current important interpretations of observations and an analysis of opportunities for the improvement of plans of interaction implemented by agents, interpretations of observation and observation techniques.

To simplify, the PDCA phases show that in order to perceive, an agent must be able to remember, plan, execute, judge, and act in such a way so that it carries out plans, and identifies and interprets (as well as judges) her/his own observations.

In some sense, a good example of implementing the approach to AI based on PDCA cycle is a well-known concept of *Case-Based Reasoning* (CBR) [218]. The idea of CBR is very natural. Namely, it is the process of solving new problems on the basis of the solutions to similar problems from the past. A typical example of CBR could be a lawyer who advocates a particular outcome in a trial on the basis of legal precedents or a judge who gives verdict using laws. In other words, a new problem is solved by finding a similar case in the past and adopting it to a new (problematic) situation. An important feature of CBR is an approach towards incremental, sustained learning since each time a problem is solved a new experience is gained, making it immediately available for the future problems. It can be done by adding a new case to the *pool of cases*. Of course, there are many practical problems with the implementation of a model based on CBR. These problems are typical for other techniques (e.g., knowledge representation, knowledge similarity relation, distance measurement, methods of reuse and manipulation on represented knowledge).

CBR is a good example that illustrates the importance and generality of *PDCA cycles*. In particular, it is a fundamental mechanism for agent's perception processes.

In our research, we assume that the act of **perceiving and perception processes are determined by interactions which enable a given agent to recognize the change in satisfiability degrees of concepts in her/his hierarchy of needs**.

In order to better explain the above assumptions concerning the approach to modeling the way in which an agent perceives, we shall provide the well-known example [366], which is a metaphorical description of the act of perceiving of a blind man walking in an unfamiliar environment.

This metaphor is summarized as follows in the book [366]:

> The main idea of this book is that perceiving is a way of acting. It is something we do. Think of a blind person tap-tapping his or her way around a cluttered space, perceiving that space by touch, not all at once, but through time, by skillful probing and movement. This is or ought to be, our paradigm of what perceiving is.

Thus, we understand the metaphor of **perceiving as a process of interaction with hunks**, as illustrated in Fig. 28.1. This figure presents an agent, trying to "complete a path" by "going straight." Suddenly, the agent encounters an obstacle, a metaphorical "fence." The agent, through appropriate interactions with the environment, can empirically verify that the concept of "an obstacle on the way" is satisfied. An agent, who is wise enough can understand that the interaction plan which has been implemented may not lead to success and that it may be necessary to develop an alternative one.

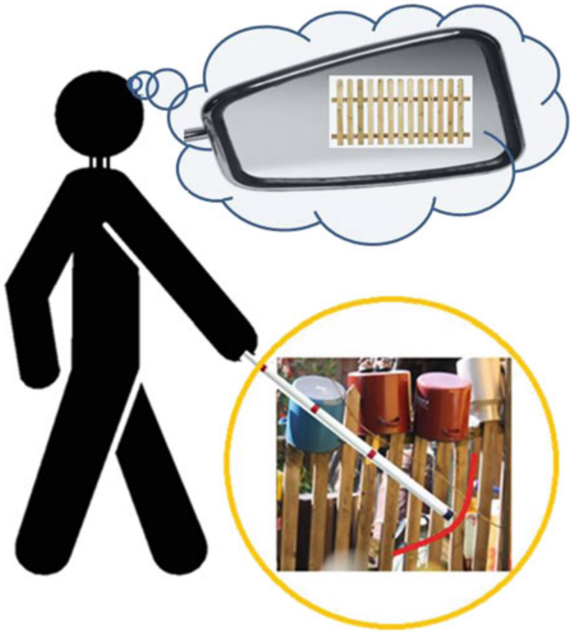


Fig. 28.1 Metaphor of perceiving process [366]

Consequently, an agent may need to replace the current priority of walking "straight" with the need to find methods to overcome an obstacle. On this basis, an agent may begin developing further plans for interactions, which lead to the creation of new plans for overcoming obstacles and their subsequent verification in terms of feasibility.

In other words, appropriate interpretations of perceived phenomena and objects in the environment are of particular importance for an agent. Reasoning leading towards the development, implementation, evaluation, and modification of subsequent interaction plans is equally important.

In order to minimize the risk of failure, an agent must be able to interpret. Interpretations are necessary for making accurate judgments regarding the conditions that trigger actions, aimed at fulfilling the agent's prioritized needs.

We may therefore assume that the agents have the ability to describe (and reflect in their memory and/or imagination) a compression of hunks and judgments on the fragments of the environment perceived by them (including observations), using concepts that they can understand and associate, through appropriate relationships, with their currently important needs and values that are to be preserved.

The more complex and subtle their structures of important concepts and techniques for reasoning about those concepts are, the more subtle their perception of the ever-changing reality that surrounds them is. Given the limitations on the scopes of attributes that are at their disposal, it can be assumed that agents are only able to approximate complex and constantly changing vague concepts, used for describing their needs and reasoning at higher levels.

28.1.3 The Physical Nature of Agent's Memory

Most generally speaking, a computer, while processing data, will employ:

- a microprocessor based control,
- a RAM—random access memory,
- a secondary memory,
- an interface with the outside world.

Analogously, we can assume that an agent, while processing data, will employ:

- agent's control,
- RAM (as depicted in Fig. 28.1 as a "mirror"),
- a memory with a slower access than that of a "mirror",
- interfaces with the outside world.

In this section, we will present a more detailed explanation of the term "agent's memory." We will also focus on the importance of Fig. 28.1 and provide examples of sample entities, processed in the agent's memory.

Entities that appear in the internal memory of our agent (i.e., on the metaphorical "mirror") are the results of interactions between an agent and the environment and interactions among the hunks perceived by an agent.

In other words, the form in which an image of the physical world is reflected on the "mirror," as well as its effect, is the result of an interaction between agent's sensors and actuators and her/his environment of activities. An agent can see, interpret, and judge this effect as her/his control interacts with her/his memory and the representation of phenomena on the surface of the "mirror." It should be noted that this effect, along with the interpretation and judgment of an agent, do not exist as separate physical entities, reflected on the surface of the "mirror." These entities are fragments of interactions that are represented in different hunks or m-hunks of an agent.

More intuitively speaking, the necessary condition which enables an agent to observe and interpret this effect is having a "mirror" (or something imitating that "mirror") and engaging in interactions that contribute to the agent's perception, interpretation, and judgment of this effect on that "mirror."

The metaphorical "mirror" presented on Fig. 28.1 is a representation of the continuously changing *agent's state of mind*, including the cache and/or agent's imagination, as well as her/his parts representing "emotional" states, which has important role in the context of discussion.

Intuitively, these "emotional states" can be treated as compressed representations of knowledge, resulting both from how agents perceive the satisfiability degree of concepts that appear in their current hierarchy of needs and from the dynamics of changes in the satisfiability degree of these concepts.

From such a perspective, "emotional states" are important to the model of interactive computing proposed in this book. This is due to the fact that in our approach, the link between the physical world of interaction and the world of agent's knowledge representation is the ability of the agent to judge (perceiving the current situation) with respect to the agent's current hierarchy of needs and her/his dynamics of change. For example, the risk of losing a house will be judged differently by an indigent, impoverished person and a wealthy, prosperous individual.

An important feature of the human mind is the ability to imagine objects and processes that do not necessarily exist in the current physical reality surrounding her/his. These imaginary entities exist in the subjective reality of the human mind. In the case of agents, these entities are stored, recalled, and eventually processed by the agent's control.

Examples of such entities include a presumed development of a given situation or abstract mathematical reasoning. They can also be agent's subjective feelings connected with the way it perceives the satisfiability degree of her/his needs. This includes the interpretation of "agent's emotions" such as pleasure, pain, euphoria, etc., referred to above.

Any entity recognizable by the agent's control, which appears on the surface of the "mirror," can subjectively exist in the "mind" of an agent (or, more intuitively, can appear on the agent's metaphorical "mirror") and shall be called an *m-hunk*. This may be treated as an abbreviation for *mind-hunk* or *memory-hunk*.

Note that any m-hunk can be formally identified with the class of all hunk configurations, aggregated on the agent's metaphorical "mirror," as the same representation, but, at the same time, as a distinctive m-hunk (or an object similar to this m-hunk to a degree satisfactory to an agent).

On her/his "mirror," an agent can imagine that the m-hunks, which represent the hunks that have been already conceived and perhaps exist in the environment, may also reflect certain interactions between the already conceived hunks. In a sense, these are the interactions which are simulated in the agent's memory.

Analogously to the definition of interaction properties as the properties of parts of hunks that have been aggregated (cf. conditions I-1–I-5 in Sect. 27.1), we can define *m-interaction properties* as properties of aggregations of m-hunks.

Of course, these 'simulated' interactions do not necessarily correspond to the actual interactions occurring in the agent's physical environment. Therefore, they are called *m-interactions*. Similarly to m-hunks, m-interactions cannot exist independently, without an agent, her/his mirror and actual interactions that trigger the m-interactions. An agent may establish and, if possible, enforce the laws governing m-interactions in the construction of new m-hunks. In other words, an agent may want to construct and accept only those m-interactions that obey the laws established by this agent.

For example, to simulate interactions in the financial markets, an agent may greatly simplify and/or distort the laws governing the behavior of the market being simulated in its models.

It is worth remembering that m-interactions are the result of actual physical hunk interactions. In this context, m-interactions are still a special case of interactions, occurring in a physical environment. Hence, not all prospectively conceivable m-interactions on m-hunks can be implemented in the external environment of an agent. In her/his memory, an agent can perform only those m-interactions that are executable by hunks available to this agent.

For instance, an agent with limited resources cannot perform m-interactions that require infinite resources. For a more detailed illustration of this example, we can conclude that if an agent exists for a limited time, then she/he will not be able to perform an addition to an arbitrary number by adding "one" to a given natural number. Thus, on her/his mirror, an agent also encodes *ways of using* for the hunks and m-hunks that she/he observes, as well as implementation techniques for interactions, described by these ways of using.

In the context of c-granules and their intuition, it is convenient to assume that the metaphorical "mirror" is divided at least into two parts.

One part reflects a fragment of the physical environment as observed by an agent. Hunks perceived by an agent, which are reflected on that part of the mirror, are, in a sense, aggregation of effects produced by the interactions of hunks, observed in the agent's physical environment of activity.

In contrast, **the other part of the "mirror" reflects images from the agent's memory which explain meaning, context, and plans connected with the processes observed in the "mirror."** This time, it is an aggregation of the effects produced by m-interactions between the agent's m-hunks. The images on the this part of the "mirror" include ideas (e.g., on scenarios for the further course of interaction) and images retrieved from the memory available to an agent. The m-hunks appearing on that part of the "mirror" can be interpreted by the agent's control as a representation of such (past and/or current and/or planned) effects of interactions, such as:

i. Past interactions, i.e., those that have been recorded by the agent's control and represent the m-hunks, which reflect the images of the physical environment or processed images stored in the agent's imagination. As an effect, they appear on the "mirror" after being retrieved from the agent's memory.
ii. Current interactions, i.e., those which represent m-hunks currently generated by an agent in her/his memory (e.g., reasoning about properties of the environment she/he observed) and registered by the agent's control.
iii. Presumed interactions, i.e., interactions that an agent presumes to happen in a given predicted context and under specific conditions for the implementation of m-interactions and/or interactions which the agent's control anticipates to trigger, once concepts known to an agent become satisfied. After these interactions have been triggered, the control will verify the conditions for their progress and completion. In the case of any significant deviation from these conditions, the control will bring them back to the original state.

In order to distinguish between these two parts of the metaphorical "mirror," let us assume that the part of the "mirror" where the environment external to the agent's control is reflected is called an *e-mirror* (*external mirror*) (cf. also the concept of "real_glasses" in Chap. 27) and the rest of the agent's memory, with direct access to the agent's control, is called an *i-mirror* (*internal mirror*) (cf. also concept of "imaginary_glasses" in Chap. 27). The whole "mirror" of an agent will be called an *a-mirror*. We allow for certain areas of the a-mirror to be occupied at times by the e-mirror and at other times by the i-mirror.

Note that by the definition, the entities recognized by an agent and present in all parts of the a-mirror are m-hunks (i.e., regardless of whether they are in the e-mirror or in the i-mirror). If m-hunks change over time through mutual interactions, we will refer to these interactions as m-interactions.

The main difference between the m-interactions taking place in the e-mirror and those occurring in the i-mirror lies in the fact that in the e-mirror, an agent can only observe, through her/his sensors, the effects of the interactions in her/his environment of activities. Of course, agents can also initiate, by means of their actuators, interactions in the environment. On the other hand, an agent can freely (within the limits of the functionality available to her/him) shape the course and impact of m-interactions in the i-mirror.

Of particular importance is the fact that an agent can establish a set of m-rules governing the behavior of m-hunks. Then, she/he can simulate interactions in the i-mirror that comply with this set of m-rules.

As part of the m-interactions, an agent's control may establish m-interactions that represent the relationship between m-hunks. Such packages of m-interactions are called *m-relationships*. M-relationships are therefore a special case of m-hunks. Agents, instead of operating on highly complex structures of m-hunks, may have control rules which assign to certain m-hunks other, much simpler m-hunks. These m-relationships among m-hunks, resulting from such a binary assignment, shall be called *m-names*.

An agent's control may discover some causal relationships occurring in its surrounding environment and formulate corresponding m-rules on this basis.

These m-rules can be combined in a finite directed graph of consecutive causal relationships, represented by m-rules designated by an agent. A graph of such m-rules can be regarded as an implementation guideline for interactive computing. For technical reasons, we assume that in this graph, there are distinct sets of m-rules (represented by sets of vertices) which support first, to start of the computation process, then in its progression, and finally in judgment of its quality, as well as possible improvements to the graph. Branching in the graph can be interpreted as non-deterministic rules.

Below, we define one such a fairly natural collection of distinguished sets of vertices, which we use for the definition of the concept of m-program. Of course, this collection can be subject to modification. Intuitively, such a network of m-rules (with sets of rules selected below) can be seen as a metaphorical representation of "software" that an agent could potentially use in certain contexts. To implement such computations, an agent, if possible, carries out or triggers them according to the rules determined in the successive vertices of the graph. In this model, we assume that an agent is capable of concurrent computing. Hence, this graph does not necessarily reduces to a linear order. An agent can observe the effects of her/his computing in the a-mirror.

By an *agent's m-program*, we mean a finite directed graph whose vertices are m-rules. We can distinguish the following types of vertices:

1. Among the starting vertices of the original graph, the following sets of vertices are distinguished:

    i. m-rules accepting a set of m-hunk configurations that set out the rules for control and triggering of actions,
    ii. m-rules accepting a set of input m-hunk configurations for computing,
    iii. m-rules for the construction of m-hunk configurations which trigger computing implemented by the graph,
    iv. consistent m-rules for computing and/or termination of computations, implemented by distinctive paths in the graph,
    v. m-rules for judging the quality of computing implemented by the graph,
    vi. m-rules for reconfiguration and improvement of m-hunks that represent the agent's m-program, so that the computing graph meets the quality judgment criteria, set for computing implemented by the graph.

2. Among the final vertices of the graph, we distinguish sets of vertices representing the m-rules which recognize and accept the computational results in a form of appropriate m-hunks.
3. Note that the concept of m-program introduced above also includes the specifications for its implementation techniques. More precisely, at the point (1.iii), we have m-rules for the construction of such m-hunk configurations that trigger computing implemented by the graph. Therefore, they are m-rules that tell an agent how to start the implementation of a program.

Speaking of computer programs, we almost always distinguish their basic concepts, such as control parameters (1.i), input data (1.ii), and the computed results (2). Therefore, these concepts will be also distinguished here. The m-hunk configurations, which are acceptable by the three types of vertices, will be respectively called: *m-control parameters*, *input m-data* and *computed m-results*. The features (1.iii) and (1.iv) specify the rules for starting and stopping an m-program. In contrast, points (1.v) and (1.vi) are meant to enable an m-program to adapt and improve their quality of functioning.

The implementation of an agent's m-program as interactions that implement m-rules within that m-program (or those that suspend the implementation of these rules) is called *m-computing*. An m-computing can be carried out in the agent's physical environment, in which case it is called *e-computing*, or it may be simulated in the agent's memory. Analogously, we refer to it as *i-computing*.

With the concepts of m-program and m-computing, one can adopt a considerable part of the apparatus used in AI on the grounds of our considerations.

For illustration, let us mention a list of sample concepts that play an important role in constructing models for IGrC.

From the point of view of AI applications, the fairly basic concept of a classifier is of particular importance. In our considerations, by an *m-classifier*, we understand an m-program that assigns certain m-hunks to other m-hunks, which might be important for the agent's control. Let us assume that an agent has a hierarchy of values for the evaluation of the results obtained from the classifier and this hierarchy grows over time. This concept is called the classifier's *m-judgment*. From a formal point of view, this classifier's judgment can be also treated as an m-classifier that calculates the appropriate confidence degree of the satisfiability of the input classifier with the input m-hunk.

For illustration, we can consider an *m-classifier* of *m-type* that assigns to every m-hunk its *main m-type*, which is also an m-hunk. It is convenient to have some binary relations between various types (e.g., according to taxonomy). Then, we can assume that an *m-classifier* of *m-type* is represented by an m-program that computes the characteristic function of the subtype relation. In such a situation, we generally assume that if an m-hunk is of a subtype of a type τ, then it is also of type τ. For example, the type 'man' is also a type of 'mammal.' Thus, the m-hunk representing John Smith is a man, so it is also a mammal.

28.1.4 The Physical Nature of Agent's Control

An *agent's control* consists of physical beings that control the interactions representing some *cause-effect relationships* and preserving some invariants of interactions among physical beings from the physical world. The cause-effect relationships establish some kind of cause-effect rules for behavior of control.

The implementation of interactions is performed in the physical world and can lead to the creation of structures consisting of physical beings, used for the activation of interactions. The agent's control has the ability to initiate some interactions following an identification of some effects of these interactions. This ability is called the *agent's perceivability*.

The cause-effect relationships establish some cause-effect rules governing the behavior of the agent's control. By this, we mean that agents can use these rules as guidelines for their decisions.

Intuitively speaking, the cause-effect rules of the agent's control can be treated as the *decision rules* of an interactive information system [467, 469]. They can be of the general form:

if a condition W is satisfied, **then take** the decision D.

It means the decisions are associated with triggering actions which, once properly executed, are expected to yield results verifiable by an agent. It is worth emphasizing that the condition W and decision D occurring in the decision rules can be trivial or highly complex.

For example, when it comes to trivial conditions, we can consider simple signals transmitted by sensors (e.g., fast-growing temperature), while simple decisions can be the reactions of actuators to these signals (escape from the fast-growing temperature).

A slightly different example of a very simple decision rule for a bacterium may be the following metaphorical decision rule:

If you feel hungry for meal X, then move head toward the greater concentration of substance Y.

Obviously, the effectiveness of such simple behavioral rules needs to be verified in real life.

In contrast, an example of a highly complex set of conditions can be a set of input attributes for market risk assessment or the evaluation of a patient's condition. These include interaction plans to counter the perceived market risk or a treatment plan for a given patient.

The examples of decision rules presented above indicate that in the case of decision rules used by an agent, it is important to find a way to effectively verify whether a condition set "W" is satisfied. If so, it is equally important to effectively verify whether the following conditions are met:

1. Is the decision set, recommended by the rules which satisfy the necessary conditions, feasible in the current situation of an agent? In particular, is the set of rules to be triggered inconsistent (that is, do we have to simultaneously perform several conflicting decisions recommended by this rule set)? It is worth to note that the set of recommended decisions can be consistent, but at the same time infeasible (e.g., due to limitations of an agent and/or constraints imposed on her/him by her/his environment of activity).
2. If a rule set is not feasible (especially inconsistent), how to resolve conflicts and how to select the optimal decision set? In particular, which criteria, preferably simple, should be used to find a decision set to be executed (e.g., weighted voting rules, consultation with a domain expert)?
3. Is a selected decision set justified in relation to the agent's current hierarchy of needs?

In real life situations, especially in analyzing large data sets, in the management of complex projects, as well as in a variety of situations, we often deal with infeasible and, at the same time, inconsistent recommended decision sets in which each decision, when considered separately, is feasible. For example, when evaluating investments, we can often have many arguments *for*, *against*, and *neutral* for particular decisions.

In the absence of any rational arguments clearly indicating which of the conflicting decisions should be taken, we often resort to making decisions on the basis of our intuition and/or beliefs.

This kind of a decision-making process has been proposed by D. Kahneman and A. Tversky in "Prospect theory" and presented in [263].

Let us recall that, as we have discussed in Sect. 9.3, according to [176], the failure of financial systems in the 21st century is caused by the failure of risk management paradigms. Greenspan in [176] emphasizes the emerging stronger influence of behavioral economics.[7] Especially, he underlines the importance of the Prospect theory, developed by the Kahneman and Tversky[8] [263].

---

[7] http://en.wikipedia.org/wiki/Behavioral_economics.

[8] http://en.wikipedia.org/wiki/Prospect_theory.

Regardless of the modality (e.g., whether based on rational criteria or on conviction), an agent should not avoid taking decisions (she/he can even trigger a set of recommended decisions that are not feasible). In project management, a failure to take a decision under the pressure of time may often turn out to be the worst and most expensive decision.

Therefore—in such situations—in addition to a set of normal decision rules, an agent should have "higher order rules," i.e., "rules about rules and contexts of their use." These rules should, e.g., support the feasibility study about the judgment of the rule sets, the selection of rules (from a rule set) to be applied, and the resolution of conflicts between rule sets selected for the implementation. The selection of rules to be applied should also be based on judgments concerning the feasibility of these rules, evaluated from the perspective of the agent's hierarchy of needs.

Higher order rules may also lay down the principles for the production of new rules. For example, having at our disposal a set "Z," containing the lower order rules of the form: $\alpha_i \rightarrow \beta$, where $i \in I$, we have a higher order rule, defining the techniques for the production of a new rule from "Z" by generating: $\bigvee_{i \in I'} \alpha_i \rightarrow \beta$, where I' is, according to the specifications of the meta-rule, a subset constructed from I. The construction of I' can be the result of agent's reasoning, learning, and experience or/and, can occur accidentally.

Basic decision rules, from which we begin to generate higher order rules, are called the *first-order decision rules*.

The rules, that allow an agent to make decisions based on the analysis of a set of recommended first order decision rules and rules for making new first order rules shall be referred to as the *second order decision rules*.

Analogously, you can define the *third order decision rules*. Namely, these are rules for making lower order rules (which in this context include first and second order rules), techniques for conflict resolution in a set of lower order rules, and judgments concerning the feasibility (including consistency) and the rationale for the execution (in the context of agent's hierarchy of needs) of the recommended set of lower order rules.

From a theoretical point of view, this induction step can be repeated infinitely many times. Thus, in this way, we can define decision rules for any order n, where n is an integer or, more generally—n is a countable ordinal.[9] Such constructed new decision rule shall be called briefly a decision *meta-rule*. We will refer to them as *m-rules*. Thus, m-rules include both simple first order decision rules as well as decision rules of any higher order.

It is worth noting that m-rules of higher orders often have great practical importance. For example, in multiplayer games, it is crucial to understand strategies used by other players, and hence the second order m-rules are important. However, one of our opponents may notice a second order m-rule used by a player. In such a situation, this opponent may develop a third order m-rule to deal the concerned second order m-rule.

[9]http://en.wikipedia.org/wiki/Ordinal_number.

Using the above concepts, we can define *rule-based agent's control* as a strategy for using and improving a set of m-rules, which are implemented using hunks available to an agent and which allow an agent to trigger, deploy and supervise (her/his own and delegated) decision-making processes, with particular emphasis on the execution of action.

These actions concern the processes of interpreting, understanding, judging and responding to interactions, occurring in the agent's environment of activity.

While considering various aspects of the rule-based control for "intelligent" robots, let us recall an important property, which is the adaptability of control, as a crucial characteristics of "practical wisdom." The issue captured the attention of Aristotle in antiquity and is emphasized in [504]:

> Aristotle's man of practical wisdom, the phronimos, does not ignore rules and models, or dispense justice without criteria. He is observant of principles and, at the same time, open to their modification. He begins with nomoi—established law—and employs practical wisdom to determine how it should be applied in particular situations and when departures are warranted. Rules provide the guideposts for inquiry and critical reflection.

Rule-based control also includes meta-rules for the modification and creation of new rules. Thus, rule-based control also allows you to implement various strategies for the adaptation of control rules.

When it comes to rule-based control, in the case of the agent's control by causal meta-relationships, we can implement multiple strategies for the adaptation of the agent's control. This is due to the fact that m-rules can modify and judge m-rules of lower orders. The basis for such judgments and modifications may constitute their usefulness and effectiveness in supporting an agent in achieving designated goals.

The implementation of these mechanisms can be carried out in a number of ways through nature based computing [115]. For example, this can be done using genetic algorithms and/or other meta-heuristics [115, 483, 544]. In other words, it is relatively easy to implement the adaptive nature of the aforesaid approach to the adaptability of the agent's control, using natural computing techniques. This implementation can be carried out through the optimization (by means of meta-heuristics) of the rule-modifying meta-rules.

28.1.5 The Physical Nature of Agent's Perception

28.1.5.1 Perception as the Integration of Acting and Thinking

Figure 28.1 constitutes the starting point for our considerations regarding the way in which the term of "perception" is understood. The drawing can be found in Sect. 28.1.2. The figure presents how an agent perceives, and according to the following paradigm: **perceiving is a way of acting** [366].

The term of "agent's perception" is far more complex than the term contained within the chunk "agent's perceiving." However, perceiving can be treated as an elementary and simple form of perception.

In general, *agent's perception* mean the **identification, classification, interpretation, and organization of sensory information in order to represent, comprehend, judge, and understand the interactions in the environment**. An agent uses the *agent's windows* for perceiving and perception. Based on the results of her/his perception, an agent modifies the *interaction models* and then, prepares and implements (in the PDCA cycle) the *interaction plans*.

In other words, an *agent's perception* consists of the agent's ability to aggregate, comprehend, and judge the parts of phenomena perceived in the physical world, accessible through agent's windows, on the basis of:

i. Construction, verification and/or adaptation of *agent's interaction models*.
ii. Relevant execution of *agent's interaction plans*.

It means that during the *agent's perception* process, an agent's control uses "agent's thoughts" in numerous ways. The following forms of using "agent's thoughts" are especially important for the agent's perception:

i. Initialization and/or modifications and/or termination of "interaction plans" in agent's environment. In particular, the "interaction plans" can be very simple (trivial) actions initiating interactions in agent's environment, or they can be agent's habits as reactions on some situations in the environment.
ii. Computing and/or storing (registering) and/or retrieving the values of perception attributes.
iii. *Granulations*, by which we mean interactions which constitute any type of processing of "agent's thoughts." Granulations include the following types:
    a. Construction, aggregation, decomposition, and destruction of "agent's thoughts."
    b. Storing, retrieving, and reconstruction of "agent's thoughts."

By moving on from the illustration of the agent's perceivability, presented in Fig. 28.1 (cf. Sect. 28.1.2), to the agent's **perception**, we shall take into account the above-mentioned aspects of the *agent's perception*.

In simplified terms, this is limited to the representation of the processes related to the "comprehension" and execution of "interaction plans," in the "agent's mind." This representation shall include the representation of the research and analysis related to the conditions occurring within the scope of the "interaction plans" within the context of the changing needs of an agent. The "interaction plans" mentioned above, are related to the following forms of agent's activity:

- metaphorical cane-probing of the environment,
- comparing the actual results of "probing" to the expected results of "probing,"
- SWOT analysis, tailored to the agent's ongoing needs,
- adaptation of the agent's hierarchy of needs, based on the SWOT analysis,
- development and/or modification of the relevant "interaction plans," taking into account the ongoing, priority needs of an agent along with the SWOT analysis conditions,

- launching and executing further "interaction plans," along with the cycles, the aim of which should be to fulfill these plans, using a PDCA scheme.

The above-mentioned processes are represented in the brain of an agent by hunk configurations. From the point of view of the agent's functioning in an unknown environment, it is particularly important to take into account the **changes perceived by an agent, especially those that have a direct impact on the needs fulfilled by an agent**. A great emphasis should be placed on the **changes that were unexpected and those that may impede the implementation of the interaction plans** in a given environment.

Let us remind that Fig. 28.1 presents a blind agent fulfilling her/his need of making a forward movement. The agent, by moving forward, systematically verifies the possibilities for the next movement. In order to achieve her/his goal, the agent carries out movement using a cane by which she/he checks whether no obstacles are present. If, at any moment, the interaction between the cane and the physical world it indicates the presence of an obstacle in front of the agent, it has a potential to become a very important piece of information for that agent. In a consequence, a change in the configuration of the agent's brain appears. **This change constitutes a trace in the agent's memory—and this trace represents an obstacle emerging in front of the agent**. The trace results from perceiving, by the agent, a phenomenon which is treated as an abnormal result of the cane reaction. The cane is a device that recognizes the route taken by the agent. The trace may be called, agreeably, *an index of change* or basically, *an index*. Figure 28.2 presents this index with a green arrow indicating the point on a metaphorical mirror, the purpose of which is to visualize the status of memory and imagination of the agent.

The agent, that was able to notice the obstacle, shall examine it more thoroughly, along with her/his surroundings. The agent shall verify the scope within which the obstacle hinders the implementation of interaction plans and find ways to overcome these obstacles. In the situation the simplest, of the one that is presented by Fig. 28.2, is the verification of proper plans of interaction within the scope of examining the obstacle. Thus, the agent will use a pointer in her/his mind. This pointer shall interconnect the index of change with the:

1. Information, imagination, and associations (placed within the agent's memory) related to the obstacle, the plans for examining and avoiding the obstacle or for introducing changes in the primary plans. Should a need and possibility of acting arise, this information would be specified by the agent within the scope of the future research.
2. Examined parts of the obstacle located within the area of the agent's activity and hunks within the area of the obstacle.
3. Interactions between the "cane" of a blind agent and the obstacle which broaden the agent's knowledge of the obstacle and of possible ways of overcoming that obstacle.

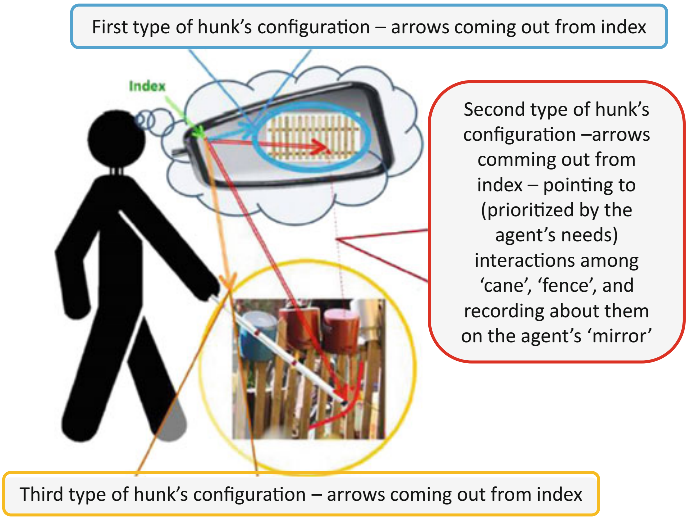


Fig. 28.2 Hunk types and association of the hunk types with the arrows, defining the interactions intended or/and expected by the agent

Summing up the above discussion, we may assume that the **change index** (cf. Fig. 28.2) is a starting point which branches out into three arrows.

A-1. The arrows indicating the **"images"** in the agent's memory—proper m-hunks, objects and processes examined by the agent's control (hunks and m-hunks). Figure 28.2 presents the agent's memory in a form of a mirror. The images of the phenomena examined by an agent are presented in a form of images on the surface of a mirror. They constitute configurations of hunks and their interactions that are distinguishable by an agent, representing the information, imaginations, interaction plans, terms and associations that are decodable/encodable by an agent. These refer to the objects and phenomena examined by an agent. Figure 28.2 presents the arrows which indicate, on the surface of a mirror, the images related to the obstacle in a form of a "fence" and her/his surroundings. The arrows may also indicate the m-hunks replicating the properties of "fence" perceived and examined by an agent, such as hardness, temperature, and dynamics of changes. Secondly, the arrows may also indicate m-hunks which contain the interpretation of the situation (its context and meaning) that occurred in the agent's environment as visualized in the mirror. The context of the situation is particularly important in terms of the hierarchy of agent's ongoing needs—including the needs resulting from the plans that were implemented.

It **may also be possible to examine m-hunks** and links through agent's control. For example, an agent may try to solve some kind of problem. In this case, the m-hunk status which is being examined may represent the current state of a solution to the problem. The m-hunk may also indicate (via an "image" in the agent's memory) other places in the memory that contain specifications of a problem that is being solved, specifications of the criteria that need to be met by a solution for this problem, as well as the results of the agent's judgment regarding the degree of approaching the solution criteria. In this situation, the arrows indicating such an "m-hunk" image constitute of the type of arrows referred to in section A-1. The m-hunks that are being examined may also be related to other situations, such as the properties of the physical processes "simulated" in the agent's memory or the construction of solutions to a given problem (including mathematical problems).

Summing up, the discussed **arrow-type consists of the "images" simulated in the agent's memory, reflecting the hunks that are being examined, along with the images of the m-hunks**.

A-2. The arrows indicating the prioritized **links between the objects that are being examined and their images, including the related processes**. The links constitute a hunk configuration that makes it impossible to carry out the *linking interactions*. These interactions interconnect the beginnings of the links, being the imaginary appearances of the hunks in the agent's mind, with the endings of the links, being the hunks and interactions examined by an agent.

In other words, this means that the *linking interactions* **interconnect the world as seen in the "mirror" (m-hunks in the agent's mind) with the world in which the objects and processes examined by an agent are present (hunks and/or m-hunks)**. They make it possible to transfer the results of interactions happening in two different environments:

i. m-interactions taking place in the agent's mind—between m-hunks that represent the imaginary objects on the surface of the mirror, reflecting the objects and/or processes examined by an agent, which are currently under the agent's scrutiny,
ii. interactions (and/or m-interactions) taking place in between the hunks (or/and in between the m-hunks) within the environment in which an agent is active, being the objects and/or processes currently examined by an agent. In our case—these are the interactions with the fence.

The linking interactions ("cane probing" of the fence) are controlled by the agent's control and are responsible for transferring the results of interactions between the "cane" and the "fence" to the m-interactions of the m-hunks and the other way round. They are mak it possible to transfer the results of the m-interaction on the agent's mirror, initiating more and more interactions between the "cane" and the "fence." The aim of these interactions is to broaden the knowledge and judgment of an agent regarding the important objects and/or processes that take place in the environment within which a given agent is active.

An agent may examine her/his m-hunks, m-interactions and the links between them. Thus, the interactions mentioned above may also constitute the endings of the links.

A-3. The arrows indicating the **objects and processes examined through the agent's control, along with the "images in the memory,"** i.e., hunks and m-hunks. Thus, this type of arrows refers to the objects and processes indicated by the *link endings*.

In Fig. 28.2, the arrows indicate hunks that are being examined in a form of the "fence" that is present in the physical world, its surroundings and properties that are recognized by the agent's control. It is worth noting, as it was noted in section A-1, that the control of an agent may also be used to examine the properties of some m-hunks, m-interactions and links. Thus, the arrows of this type may indicate hunks that are being examined, as well as m-hunks, m-interactions and links that are being examined.

Figure 28.2 is a graphic representation of these three types of arrows that have been mentioned above.

28.1.5.2 Integration of Acting and Thinking by Links Using Linking Interactions.

The perception and interaction mechanisms described in Sect. 28.1.5.1 (including the conditions: A-1, A-2 and A-3) indicate that "linking interactions" are of key value in the process of the agent's perception. This role is also illustrated in Fig. 28.2.

The main intuitions related to the "linking interactions" are shown in Fig. 28.2, which presents the perception of a blind agent who is in the middle of a process of identifying the obstacle in a form of a "fence." The interactions are being carried out mainly through the process of controlling the "cane" interactions, which transmit the signals between the following two environments:

li-1. **Environment of the researcher's mind: the so-called "mirror."** It is an environment available to the agent's control, embedded within the memory of an agent. Within that environment, an agent records and interprets the effects of interaction between the cane and the fence, and controls further movements of the cane.

In other words, the linking interactions are the interactions which support the mutual communication (transmission) of the results, obtained by means of an interaction between different environments of hunks.

li-2. **The examined environment**—i.e., the metaphorical "fence" and its surroundings. This is a physical world which is partially available to the agent's control via the effects of interaction between the agent's sensors and actuators and the environment (interactions between the cane and the fence).

In other words, the *linking interactions* are the interactions which support the mutual communication (transmission) of the results, obtained by means of an interaction between different environments of the hunks.

LI-1. **Hunks and interactions placed within the "agent's mind"** are represented in Fig. 28.2 with the use of m-hunks and processes of m-interactions, happening on the surface of the "mirror." These include: replication of the effects of the ongoing perception of the environment, context and state of the art, components of planning and judgment made by an agent, representation of the structuring, launching, monitoring, embedding and adjustment processes related to the interaction plans.

LI-2. **Hunks and interactions with objects and processes** that are being examined, illustrated in Fig. 28.2 as "cane probing" of the agent's environment (e.g., the "fence"). It shall be strongly emphasized that **this type of interactions is indispensable for the execution of complex tasks**. This results from the fact that it is impossible and unnecessary for an agent to create correct and detailed interaction plans for complex tasks. Thus, systematic adaptation of the plans, throughout the process of their implementation, is required.

As the term of "linking interaction" plays such a significant role, below, we present a proposition of how to expand its scope of meaning. In order to do that, we shall assume that X is seen as a configuration of the agent's m-hunks, while Y is a configuration of the hunks (or configuration of the m-hunks). Moreover, we may also assume that if the contexts suggest which X and Y we are referring to or if an unambiguous determination of X and Y is of no relevance to our considerations, instead of using the term *"link between X and Y,"* we will simply use the term *"link."*

A link between X and Y (cf. also Sect. 32.3.2 and Fig. 32.1) is the configuration of hunks, being at the disposal of an agent, that make it possible to execute the *linking interactions* with the properties as follows:

L-1. Interactions determined by the link between X and Y may occur in one of the active states:

i. *Active link*—a link functioning in line with section L-3.
ii. *Pending link*—a set of interactions consisting of hunks that are ready to be used by an agent in order to quickly launch the interactions indicated by the agent's control.
iii. *Suspended link*—a link that does not work in line with the expectations of the agent's control.
iv. *Archived link*—is a representation of the link that is referred to as a *saved link*, that is one which is saved in the memory potentially available to an agent. The saved link may be reconstructed by an agent.

L-2. The link may be introduced into one of its states described above:

i. Directly by the control of an agent.
ii. Indirectly by coded (through the agent's control) hunk configurations which put the link into an active status (using the signal arranged with the control system).
iii. Some other, random situations.

L-3. In the active state, the link differentiates between three *types of link interactions*:

i. *Beginning of the link*—constitutes interactions represented by the m-interactions of an agent, involving the m-interactions with X. They may be used to send and receive the signals (i.e., the processes that go along with the interactions of the hunks that are transmitting the changes in the configuration of the hunks, respectively in Y and/or X direction) from/to the ends of the link (involving Y). The interactions involving receiving and/or sending the signals are interpreted as follows:

a. Signal-sending (initiated by the agent's m-interactions in which it is involved—m-interactions with X) initiates the interactions that contribute to the transmission of the link signals to the endings of the link. The results of these interactions are recorded by the interactions that build up the endings of the link, no matter Y be a hunk (not an m-hunk), or m-interactions, or an m-hunk. Should Y be a hunk (not an m-hunk), then the endings of the links act as the *agent's actuators*.
The interactions between the endings of the link created in this way may stimulate further interactions in the environment of the agent's activity, particularly, they may reach other agents.

b. Signal-receiving is a result of the signal transmission process, originating at the ending of a link. These signals are initiated by the interactions that are building up the endings of a link if Y is defined as a hunk. In such a situation, the link endings may act as the *agent's sensors* or they may constitute the *implementation of the agent's m-hunks*. The interaction between the endings may be generated by the interactions occurring within the environment and/or by an agent and/or some other agents. In the case when Y constitutes an m-hunk, the transmitted sources come from m-interactions that involve Y.

ii. *Endings of the link*—these are hunks and their interactions and/or interactions that represent the m-interactions of the m-hunks (involving Y)—they represent the interactions that aim at *sending the signals* and/or *receiving the signals* from/to the beginning of the link.

iii. *The processes related to the transmission of the link signals*—hunk interactions that make it possible to transfer the signals between the beginning of the link and its endings.

The links themselves may be divided into two separate groups:

- *Implementing links*—the ones that constitute an implementation of the m-hunks (cf. this section, the L-3b—link definition). The hunks that are directly involved in the interactions, that are the building blocks of these links, are referred to the *hunks implementing the link*.
- *Communication links* are the links that constitute the implementation of the agent's sensors or actuators (cf. this section, the L-3—link definition). When Y, present in the link definition, is a hunk (or an m-hunk, accordingly), then the hunks (m-hunks, accordingly) that are directly involved in the interactions (or, -accordingly,

in the m-interactions), namely the endings of the hunks, are referred to the hunks indicated by the link (*m-hunks indicated by the link*, accordingly).

28.2 C-granules as Computational Building Blocks for WisTech

28.2.1 C-granules as Components of Perception via Interactions

Figure 28.2, in Sect. 28.1.5.1, describes the so-called "index of change" as a starting point for three arrows, representing different types of hunk configurations. This index plays an important role in constructing, integrating, and initiating agent's thoughts and interactions, that are interpreted by an agent. Once the agent's thoughts are integrated and initiated, further activities, data processing and agent's actions, are performed, and during which the initiative is taken by the "thought" that has been initiated, until it is replaced by another thought. In simplified terms, functioning of the active "thought" is limited to the interaction between the agent's control and other "thoughts," as well as the agent's environment of activity.

In order to focus on the concept of a "thought" (separated from the index of change that initiates a given thought and integrates it)—we have to reformulate the conditions A-1, A-2, and A-3 described in Sect. 28.1.5.1, in a way that it will **omit the arrows from the index of change**. Thus, we are able to distinguish several conditions, described below as B-1, B-2, and B-3, that define the following three types of components which form the "thought structure:"

B-1. *Soft-type hunk configuration, consisting of the* "images" in the agent's memory (i.e., proper m-hunks of the processes and objects examined by the agent's control, i.e., **hunks and m-hunks**).
Figure 28.2 presents some states and phenomena in a form of images, projected on the surface of the mirror. They are the configurations of hunks and interactions that are distinguishable by an agent, representing the information, imaginations, interaction plans, terms, and associations that are decodable / encodable by an agent. These refer to the objects and phenomena examined by an agent. M-hunks in a soft configuration also include the m-hunks that represent the properties of the "fence" perceived by an agent, including its hardness, temperature, and dynamics of changes. They may also include the interpretation of context expressed by m-hunks, where the interpretation is carried out by the agent's control. The context refers to the meaning of a given situation within the environment visualized by the mirror. The context of the agent's situation is particularly important in the light of the hierarchy of her/his ongoing needs—including the needs that emerged after the implementation of plans.

Analogous to A-1, in the case of B-1, we may also assume that the “agent's control” may be used to **examine m-hunks**.

B-2. *Link-type hunk configurations, consisting of the prioritized* links between the objects that are being examined and their images, along with the **related processes**. The links constitute a hunk configuration that makes it possible to bring about the *linking interactions*. These interactions interconnect the beginnings of the links, being the imaginary appearances of the hunks in the agent's mind, with the endings of the links, being the hunks and interactions examined by an agent.

B-3. *Hard-type hunk configurations*, consisting of the **processes and objects examined by the agent's control**, i.e., hunks and m-hunks. Thus, these are the objects and processes indicated by the *link endings*.

In the example presented in Fig. 28.2, it is the “fence”, its surroundings, and the properties of the surroundings that are recognized by the agent's control.

It is worth noting that, as it was noted in section B-1, the agent's control may also be used to examine the properties of some m-hunks, m-interactions, and links. **Thus, the arrows in a hard-type hunk configuration may indicate both the hunks that are being examined as well as m-hunks, m-interactions, and links**.

We should also note that the above-mentioned conditions B-1, B-2, and B-3 are independent from the graphical representation of an agent in Fig. 28.2 and her/his form. As a result, in order to enhance the clarity of our visualization, illustrating the framework structure of the perception process, as well as the agent activation process, we may imagine that an agent in question is symbolically represented as a three-layered box with hunk configurations of a proper type, in line with the conditions described above (B-1, B-2, and B-3).

Such an approach towards the perception and agent activation processes leads to a clearer visualization of their structure. In this way, we may arrive at Fig. 28.3, which, like a framework, illustrates the *c-granule* concept.

The conditions described above (B-1, B-2, and B-3) refer to the processes of perceiving and initiating an action by an agent (e.g., during the process of identifying and overcoming an obstacle in a form of the metaphorical fence) in a form of the hunk interaction processes. This stems from our assumption that the processes in the physical world, that are recognized by an agent, i.e., the hunks (of this agent), are represented in her/his brain by means of proper hunk configurations (using m-hunks) perceived by the agent.

An agent perceives the hunks by interacting with them (e.g., she/he measures the portion of their electromagnetic spectrum), using available sensors and actuators. An agent who is color-blind, sees the world in a different way than an agent who can differentiates the light spectrum's tonal range. An agent, on the basis of the available interactions, calculates the attribute values resulting from these interactions. These values are placed and processed in the agent's metaphorical RAM memory, which can also be treated as a hunk. Thus, in simplified terms, the models of perception and agent activation processes both may be considered as the processes of interaction,

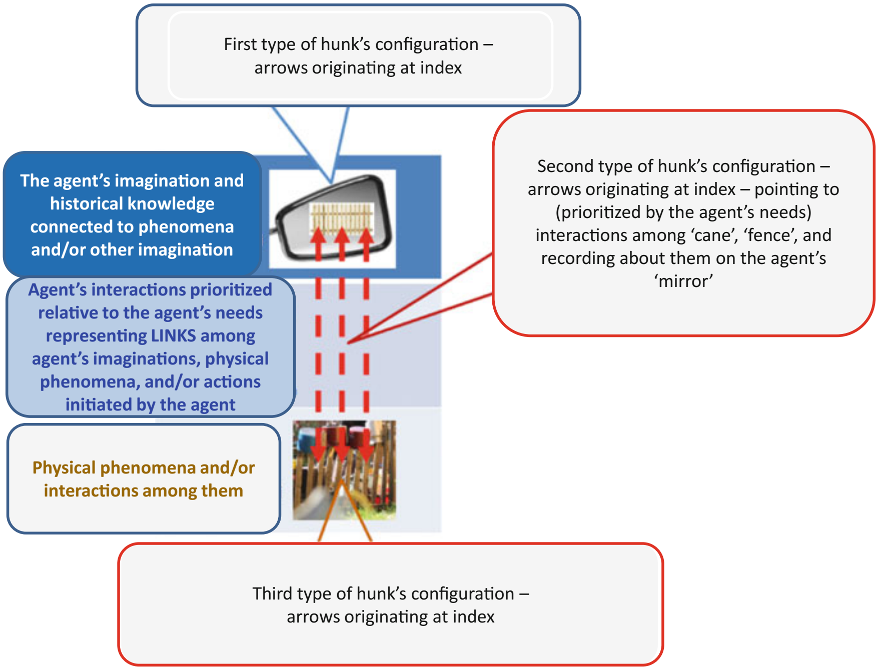


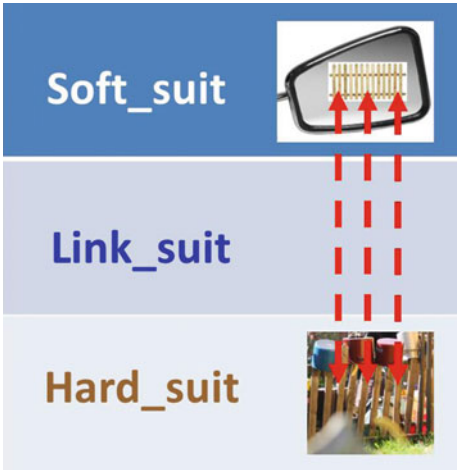
Fig. 28.3 Illustration of the c-granule term. It is worth noting that the links may initiate and transmit the results of interactions from the ending to the beginning of the link and other way round

embedded within the configuration of hunks. These interactions take place both among the hunk configurations and between an agent and her/his environment of activities. In this context, it is worth mentioning that Fig. 28.3 constitutes a symbolic image of the actual physical processes.

When it comes to the terminology related to c-granules, instead of using terms such as "soft-type hunk configurations," "link-type hunk configurations," and "hard-type hunk configurations" (as it was in the B-1, B-2 and B-3 conditions), we usually use shorter names, e.g., *soft_suit* (*link_suit* and *hard_suit*, respectively). These names are related to the terms "software" and "hardware," both of which are commonly used in the field of IT. In the context of this terminology, Fig. 28.3 becomes more comprehensible once we name the subsequent layers in line with the conventions described above, i.e., **hard_suit, link_suit**, and **soft_suit**. In this way, we may arrive at Fig. 28.4. which illustrates the concept of a c-granule.

Of course, Fig. 28.4 has a very general character. It roughly represents the layers of the c-granule's structure of the architecture. Thus, the drawing should be complemented with a brief summary of the way, in which we understand the c-granule layering. Summing up, we may state that the structural and physical outline for the c-granules consists of three layers, all of which may be described, in simplified terms, as follows:

Fig. 28.4 Concept layers of the c-granule architecture



D-1. **Soft suit**—these are the configurations of hunks representing the content of "thoughts/emotions," judgments, interaction plans and agent's reasoning (within a given c-granule). Some of these representations are transferred from the ongoing imagination to the other data, carried in the agent's memory.
In case of the soft_suit's component, this may include the specifications of the algorithms, interaction plans, imaginations, emotions, manipulation methods, methods for the implementation of plans, etc.
In line with the terminology adopted in Sect. 28.1.3, the soft_suit components, recognized by an agent, are referred to the *m-hunks*, while the interactions recognized by an agent are referred to as the *m-interactions*.
In the case of the soft_suit, it is possible to create m-hunks that represent the results of interactions. They are formed once the results of other interactions are transmitted by the links (e.g., from the environment). For example, the links described in D-2 may transmit such results of interactions from the sensors. What is more, m-hunks may, using their own interactions, initiate such a transmission to actuators by means of links.

D-2. **Link_suit**—these are the configurations of hunks that make it possible to carry out linking interactions (cf. Sect. 28.1.5.2). The beginnings of the links are embedded within the soft_suit, while the endings may be present in a form of hunks, interactions, m-hunks, m-interactions, and, finally, links.
These links indicate and clarify the meaning of a "thought" (represented by c-granules) by providing relevant examples of terms represented by these links, their clarification, and interactions that were undertaken (including the ways in which an agent uses terms represented by c-granules). The links may also enable an agent to carry out her/his "granular" computations by activating her/his sensors and/or actuators (in order to obtain some values of the attributes and/or initiate some action within the environment). Hence, links indicate hunks and/or m-hunks.

For example, if an agent uses a c-granule to represent the “agent’s thought” about a “tree,” we may represent this “thought” in a following way: an agent imagines, using an m-hunk in the soft_suit, an image of a tree and is in a possession of a link that indicates the actual tree that is present in the agent’s surroundings.

Other links may also be connected to the “agent’s thoughts” (actually, with a relevant c-granule) concerning a “tree.” For example, these links may indicate other trees and images of numerous trees embedded in the memory. These may also be the links that indicate the agent’s (or other agents’) actions related to using a tree. The links may be additionally marked with a variety of labels and parameters (e.g., the control parameters). For example, the labels may specify priorities which reflect the current importance of active links for a given agent. They may also describe the level of clarity/intensity of the given hunks and/or m-hunks; they inform an agent to what degree of accuracy/certainty a given term is recognized or to what degree we recall/remember that term.

We should also note that link’s endings in particular may indicate the hunks included within the soft_suit and/or link_suit.

D-3. **Hard_suit**—these are the configurations of hunks indicated by the links included in the sof_suit. As it was in the case of the D-2 condition, here we may also note that m-hunks from the soft_suit may appear within m-hunks from the hard_suit and/or links from the link_suit.

The outline presenting the physical structure of c-granules described above (cf. sections D1, D2, D3) will be complemented with an outline presenting the functioning of c-granules. A variety of aspects of such an outline is presented in the subsequent parts of this book. We should start with those components that play a key role in the functioning of c-granules. In order to do that, we will use schematic drawings—Figs. 28.5, 28.7 and 28.6.

Figure 28.5 illustrates the following functional components of the c-granule G (henceforth referred to as G).

1. An ellipse located at the top of the drawing represents the C concept implemented by G.

    a. The representation of the C concept implemented by G includes the representation of *various aspects of the C concept* available to G. These include particularly such aspects as ways and contexts of using C, imaginary concepts related to C, intuitions, action plans, opinions of experts and other forms of knowledge related to C.
    b. The C term is represented by soft_suit m-hunks, which are specifying relative to G (G-*specifying*) its various aspects.
    c. *The aspects of the term* may have a variety of degrees when it comes to their certainty and credibility.
    d. The *specifications* of the individual *aspects of the C term* can be expressed both in a vague, natural language, which uses terms represented by c-granules, and by non-vague terms.

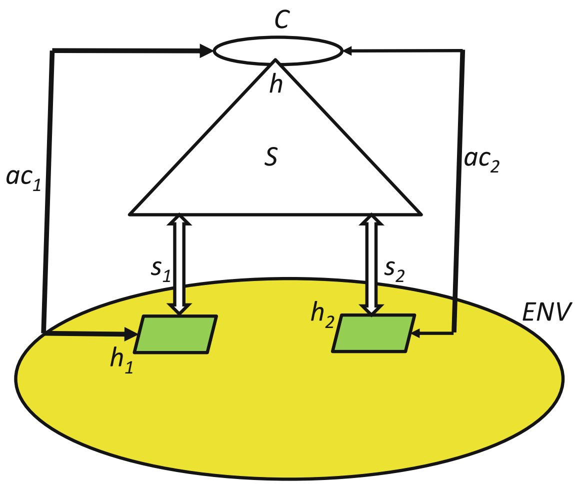


Fig. 28.5 Simplified concept of c-granule G operation

For example, we may imagine that C is a term which expresses a risky situation on the road, monitored by an autonomous drone.

2. The ellipse located at the bottom of the drawing, marked in yellow, physically represents the environment of hunks (ENV), part of which is available to the links of G (including sensors and actuators).
3. Among the links present in the link_suit of G, s_1 and s_2 sensors along with ac_1, and ac_2 actuators can also be found. These links allow interactions with the hunks within the environment. They are represented by the green h_1 and h_2 parallelograms.
4. h_1 and h_2 parallelograms symbolize the environment that is currently within the reach of G, representing the C term with adequate active sensors and actuators. The active scope of the agent's environment is referred to as the *agent's windows*.
5. The links (sensors and actuators) of the c-granule can be embedded within the scope of the hunks that are directly available to the hunks. It is also possible that the links may be "expanded" to reach an environment which is more distant (this refers to, for example, the Internet signal transmission, vehicle control on Mars, observation of the space or detection of the elementary particles).

   The S-triangle represents the *data synthesis* scheme, in the case of which the sensor-gathered data is used, including the data coming from s_1 and s_2 sensors. Usually, this is a hierarchical classifier that calculates the complied degrees of the C term (within the scope of its validity in the assessment scale used by G). For example, S may calculate a variety of approximations for the C concept (lower concept approximation, upper concept approximation and the boundary of the concept). It may also calculate the degree to which various criteria of the term are being complied with.

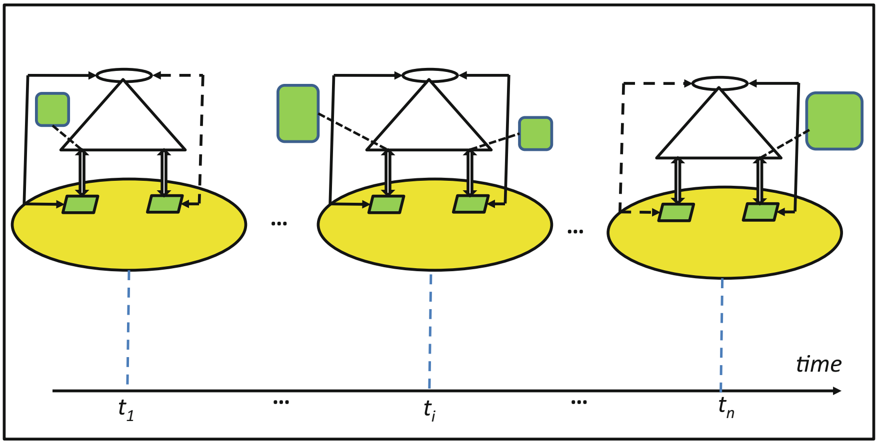


Fig. 28.6 Illustration of a change in the activity and role of the c-granule "windows"

6. As a result of the calculations performed via the *S synthesis scheme*, *h* hunk emerges, which is being shown at the top corner of the triangle that represents *S* the synthesis scheme. The current version of the *h* hunk, defined in this way, is referred to the *current result of the S synthesis* leading to the approximation of the *C* term.
7. The *current result of the S synthesis* is available to the m-hunks in the soft_suit, *G-specifying the aspects of the C term*. On this basis, the *C* term specification may recommend the c-granule *G* to initiate the ac_1 and ac_2 operations.
8. The c-granule *G* may also activate other sensors and actuators. Moreover, the c-granule *G* may also control its own sensors and actuators by directing them towards other hunk areas available to the agent's environment. In this way, the c-granule may activate or deactivate the *agent's windows.*

The simplified concept of the c-granule *G* should be treated as a rough approximation of the selected intuitions related to the complex c-granule concept.

Another set of very important intuitions is connected with illustrating the dynamic of changes within the c-granule windows. These changes may be the result of the new needs, identified by the agent's control. The new needs may cause the c-granule to activate and/or deactivate sensors and actuators, leading to the activation, and/or deactivation of the windows or introducing changes to the windows. For example a given window may have a character of a "sensor window" (a window which acquires sensor data), while after a change, its role may evolve to that of an "actuator window" (window which initiates the interactions within the environment). Consequently, the schemes of synthesizing the data (acquired by sensors) may obtain a variety of data from a variety of windows. They may also initiate a variety of new interactions. The situation is well illustrated by Fig. 28.6.

Usually, the agent's control has many c-granules at her/his disposal. These c-granules may cooperate and/or compete with each other. In general terms, we may

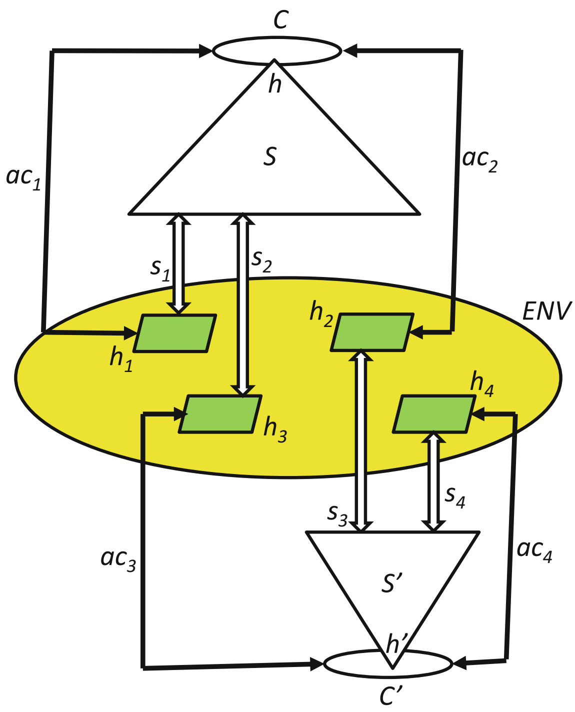


Fig. 28.7 Illustration of cooperation and/or competition between two c-granules having access to the same "windows"

say that c-granules are playing a certain type of a game with one another. They have a game-space at their disposal—the environment within which an agent is active. An access to that environment is ensured by the signals transmitted by the c-granule links, grouped and arranged in the c-granule windows. Thanks to these links, a c-granule is able to monitor and/or initiate interactions within the framework of such a game. In such a situation, it may turn out that the actuators and sensors of two separate c-granules are functioning within a single window in a given environment. The cooperation and/or competition between two c-granules having access to the same windows is illustrated in Fig. 28.7.

While discussing the properties of c-granules, one should remember that hunks constitute the basic building blocks for c-granules. They may be considered to act as space-time blocks. Thus, **c-granules may also be considered as space-time elements.**

Intuitively, referring to Fig. 28.2, this means that the "metaphorical fence with pots hanged over it," the "fence's image in the mind of an agent, without the perception of the pots", as well as link-constructing "interactions" should be all treated as space-time entities. Thus, the metaphorical "fence in the agent's mind" is considered to be more of an "agent's movie about that fence" than a static image of the fence.

An agent, on the basis of the available attributes, may recognize the fact that the fence that is being observed was transformed into an object which is no longer recognized by this agent as a fence. For example, the fence could have been destroyed and replaced with two piles of boards, with a safe passage in between the piles or with unsafely protruding nails.

From the point of view of an agent who has a need of entering into the area behind the fence, a new situation arises—it may be at the same time represented by a c-granule that represents the "index of change." In other words, c-granules may indicate, by means of links, other c-granules and they may also be transformed into new c-granules.

28.2.2 Context of Using C-granules in Perception Functioning Schemes in WisTech

In line with our understanding of the two concepts discussed in Sect. 28.1.2 and illustrated in Fig. 28.2 (Sect. 28.1.5.1), "perceiving" and "perception," we can presume the following:

The agent's perceiving and perception processes are determined by interactions, which allow an agent to recognize and change the satisfiability degrees of concepts present in her/his hierarchy of needs.

A special role in these processes is assigned to "thinking" processes implemented by an agent. These processes are illustrated in the figures referred to above, as processes illustrated by m-interactions on the "mirror" (cf. Fig. 28.2) of an agent.

The approach to modeling "thinking" processes proposed in this book uses a key mechanism in the form of *linking interactions*. They link the agent's "thoughts" with the agent's environment. In this context, in Fig. 28.2, the perception process is illustrated by imposing a control over the interactions that link "thought" with physical objects which are being tested. In this case, linking interactions are implemented by interactions which transmit signals between the "mirror" and the fence, and the other way round, through the "cane." This allows for an exchange of interaction results between the two interaction sites:

1. **Interactions in the agent's "mind"**—intuitively, these are the interactive processes that occur on the agent's metaphorical mirror.
2. **Interactions with the objects and processes that are being tested in the agent's environment of activity**—these are the interactions of the "fence" with the surrounding environment, in particular, with the agent's "cane."

In this section, we will discuss the context of using the c-granule concept in the perception functioning schemes in WisTech, in the form of two graphic schemes:

1. A contextual outline presenting the structure of packages for perception functioning modules and interactions implementing this perception (cf. Fig. 28.8).
2. A contextual outline presenting the functioning of perception and its dynamics cycle, along with interactions implementing this perception (cf. Fig. 28.9).

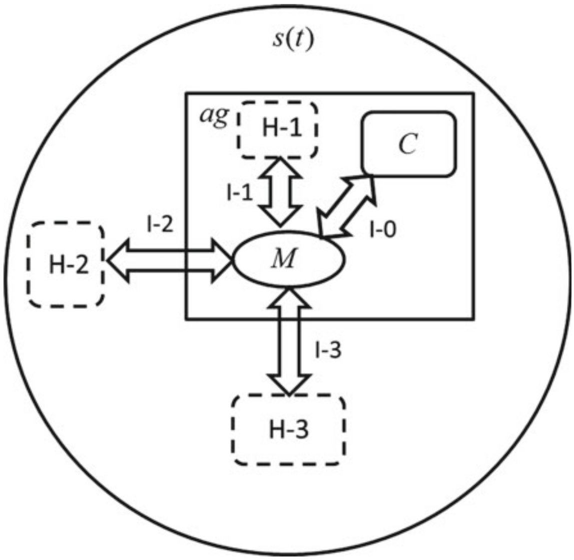


Fig. 28.8 Outline presenting the packaged structure of functional modules relevant to perception and interactions implementing this perception

Figure 28.8 outlines the packaged structure of functional modules relevant to perception and interactions implementing this perception. The figure introduces an agent, denoted as *ag*, who has many hunks at her/his disposal. The hunks that have been distinguished are: C, M, H-1, H-2, H-3, I-0, I-1, I-2 and I-3. This figure can be treated as a picture, taken at a given time *t*, of selected *ag's* functional modules, in which the world around the agent is in the state $s(t)$. For the sake of our further considerations, we will pay special attention to Fig. 28.8 and the role of c-granules and linking interactions that allow for the implementation of IGrC computing. Figure 28.8 contains the following components:

- $s(t)$— state of the world at a given time *t*, a part of this world is the agent's environment of activity, i.e., hunk at *t*, representing the physical object (world) perceived by an agent *ag*,
- *ag*— *c-granule*—hunk structure implementing the agent's interaction processes which are directly available to an agent (*ag*),
- C— *c-granule*—hunk structure implementing interaction processes for the agent's control,
- M— *c-granule*—hunk structure implementing the agent's interaction processes, initiated by the agent's control or by other c-granules (with consent of the agent's control)—"agent's thought" implemented in her/his mind, which also is also a hunk,
- H-1— *c-granule*—hunk structure implementing the agent's interaction processes, initiated by the agent's control or by other c-granules (with consent of the agent's control)—supplemental "agent's thought" (examples of c-granules' example) implemented in her/his mind and which supports the implementation of the M "thought" M,

H-2, H-3— *hunks*—hunks structures implementing the agent's interaction in processes for representing the objects and/or processes involved in the agent's activity (perception, research, usage and/or interactions enforced by the processes occurring in the environment),

I-0— *hunk* implementing *linking interactions*—hunk structure implementing the agent's interaction processes in order to transmit the representations of results obtained through:

- * interactions within agent's control, and
- ** interactions within agent's "thoughts" represented by M,

I-1— *hunk* implementing *linking interactions*—hunk structure implementing the agent's interaction processes in order to transmit the representations of results obtained through:

- * interactions within agent's "thoughts" represented by M,
- ** interactions within agent's "thoughts" represented by H-1,

in our computing model, we assume that a "thought" (i.e., c-granule) can process and/or create another "thought" (c-granule); in this context, the interactions described as I-1 (which connect two c-granules, marked in the figure by H-1 and M) depend on whether the M and H-1 c-granules will properly perform their computations and whether they will communicate with each other via I-1.

I-2— *hunk* implementing *linking interactions*—hunk structure implementing the agent's interaction processes in order to transmit the representation of results obtained through:

- * interactions within agent's "thoughts" represented by M,
- ** interactions within the activity scope of the H-2 hunk located in the agent's environment of activity.

I-3— *hunk* implementing *linking interactions*—hunk structure implementing the agent's interaction processes for the transmission of representations of interaction effects between:

- * interactions within agent's "thoughts" represented by M,
- ** interactions within the activity scope of the H-3 hunk located in the agent's environment of activity.

Figure 28.9 presents the dynamics cycle of how perception works. The structure of its functional modules is illustrated in Fig. 28.8. To describe the dynamics of changes within this cycle, we will use four states (phases of the cycle) in their order of occurrence:

CC-1 At time t (at the agent's time scale), the agent's surrounding world is in state $s(t)$. By judging the current needs of an agent, the agent's control determines that in the near future, a new agent's "thought" (i.e., a new "c-granule"), which in the figure is indicated as M, will be required.

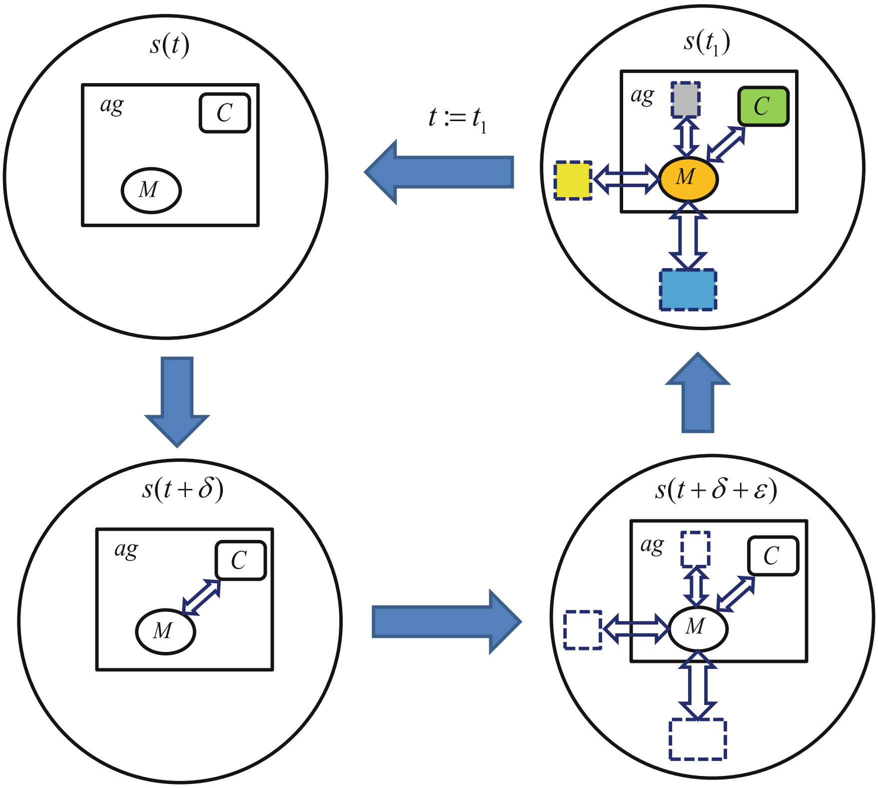


Fig. 28.9 Four phases showing the functioning of the agent's perception and its dynamics cycle along with interactions implementing this perception

CC-2 At time $(t + \delta)$, the agent's surrounding world is in state $s(t + \delta)$. The agent's control retrieves (from the accessible memory) the M "thought" (i.e., c-granule) or it may initiate a process by which M is created. Then, the control stimulates an interaction with M and M is activated. Once activated, M, in the subsequent phases, will perform computation. To this end, it should have prepared the necessary infrastructure, based on the current needs, i.e., supplementary "thoughts" (c-granules) and hunks, and, if necessary, be prepared to activate sensors and/or actuators assigned to it.

CC-3 At time $(t + \delta + \varepsilon)$, the agent's surrounding world is in state $s(t + \delta + \varepsilon)$. The agent's control supports the M "thought" (c-granule) in preparing the necessary infrastructure (created by c-granules shown in the figure) to initiate computation implemented by interactions among c-granules. In particular, based on the current needs, other supplemental "thoughts" (i.e., c-granules) as well as necessary sensors and actuators used for interactions in the agent's environment of activity, may be activated.

CC-4 At time t_1, interactions are activated and at this point, they are implemented by the M "thought" M (i.e., c-granule). In general, they are interacting both with the hunks located outside a given agent, as well as with other "thoughts" (i.e., c-granules) at the agent's disposal. While computing, it may become necessary for the agent's control to trigger new "thoughts" (i.e., c-granules) and in this way, through CC-5 phase, initiate a new phase described as CC-1.

CC-5 The expression $t := t_1$ means that we repeat the whole cycle for a new "thought" (c-granule) of M at time t_1, i.e., we initiate the CC-1 phase at a new moment of time.

The description in Fig. 28.9, presented above, provides a general view of phases CC-1, ..., CC-5 that describe the functioning of the agent's perception and the interactions implementing this perception. This description does not encompass their prospective applications. They concern the possibility of implementing the PDCA cycles that could support the functioning and optimization of the agent's control.

Below, we have sketched such PDCA cycles and their possible implementation based on the phases described above (CC-1, ...,CC-5 in Fig. 28.9). In our considerations, we refer to Fig. 28.1 which depicts the perception process of a "blind" agent in the process of identifying an obstacle in the form of a "fence."

Let us assume that the "blind" agent is in the state C-4, completing her/his task; in particular, one can consider that she/he is checking the possibility of further movements. If the agent's judgment concerning the results of interactions between the "cane" and the physical world indicate the presence of an obstacle which impedes further movements in the current direction, then the agent should highlight this observation, initiate an investigative action, and undertake applicable corrective actions.

Let us note that in this process, it is crucial to focus on the situation during which an agent encounters the obstacle. In our model, we assume that in such a situation, a new hunk configuration, which we call an *index*, appears in the agent's mind. In Fig. 28.2, this new m-hunk (index) configuration is marked by a green arrow. Intuitively, this index records a "trace" of perception's results in the mind of an agent, judged by the agents as a presence of an "obstacle." Therefore, in a sense, in the agent's mind, there occurs a specification of a new "thought" (i.e., c-granule) as part of M. Such a "thought" retrieved from "memory" and/or constructed a new appears in the CC-1 phase. Usually, the specifications of "thoughts" concern the techniques for the execution of interaction plans, including:

a. *An initial assessment of a change (its form and scope) in the expected reaction to "cane probing" in the agent's environment of activity*. For example, the agent should recognize that there, indeed, is an obstacle whose form and scope prohibit the continuation of the agent's interaction plans.
b. *Establishing and initiating an interaction plan which explains in more details the changes and consequent deviation from expectations*. Therefore, it is important to prepare and initiate corrective actions for the agent's interaction plans (paths).
c. *Planning and preparing for a possible initiation of the corrective actions for the interaction plans (path) that have been implemented*. These preparations ought to be synchronized with explanations.

In Fig. 28.9 in the CC-2 phase, this “thought” (c-granule denoted by M) constitutes a part of the memory hunk over a certain interval of the agent’s time. An agent constructs it anew or retrieves it from the memory in the CC-5 phase. Next, in the CC-2 phase, an agent activates this “thought” (c-granule) in order to control the further interactions of the “cane.” The M “thought” (c-granule), once activated, verifies the interactions present in its mechanisms and prepares for their initiation (in particular, she/he prepares available sensors and actuators). These tasks are completed in the CC-3 phase. Then, the agent’s control (or another c-granule) launches the interactions initiated and controlled by the “thought.” This constitutes a new phase known as CC-4.

Depending on their abilities, agents in the CC-4 phase can concurrently check for multiple possible directions of verification. Other phases can also be implemented concurrently and/or independently. In other words, we should bear in mind the possibility of implementing many PDCA cycles. Some of them may share common elements. Especially, the CC-4 phase may include many overlapping cycles.

To sum up, the agent’s interactions associated with “tapping” the obstacle (in a CC-4 cycle) **allow the agent to systematically expand her/his understanding of new conditions and/or needs in a changing environment of activity.** This is particularly important for the following reasons:

i. An agent’s adaptive and current judgment related to her/his current SWOT situation, based on her/his current hierarchy of needs.
ii. Planning and modifying interaction plans for an effective fulfillment of the agent’s current priority needs. The basis for such planning and modifying includes:
    a. the effect of the current SWOT judgment in the agent’s current situation and
    b. current interaction plans in effect.
iii. Preserving, in agent’s memory, important and recurring m-hunks, especially patterns of processes (e.g., causal processes that an agent can use to support the implementation of her/his priority of needs) and their properties. This applies above all to processes resulting as important consequences according to the agent’s current hierarchy of needs and their dynamics of change.
iv. Searching for new and more efficient techniques for computing important features. This includes discovering and learning new concepts that support agent’s activities, such as:
    a. Better recognition and understanding of the agent’s current needs and their hierarchy.
    b. Development of effective interaction plans that lead to fulfilling agent’s priority of needs.
    c. Effective adaptation and implementation of partially known interaction and dynamically changing environments.
v. Constant verification and expansion of knowledge along with agent’s hierarchies of needs, expressed in a language of concepts available to an agent.

28.2.3 Examples of C-granules

Conceptually, the simplest intuitive example of a c-granule appears to be that of a sensor. A sensor can be very simple, e.g., it can be used only for measuring the temperature of a specific physical object at a given location and at a given time, or it may consist of multiple measurement devices (e.g., it can be a set of instruments to monitor weather or simple industrial cameras). Generally, such sensors have control parameters for their procedures of measurement and data retrieval. This can be, for example, changeable settings of a camera or the duration of measurement. The results of successive measurements can be saved and presented in a table which represents an information system. An example of such a c-granule is presented in Fig. 28.10.

Of course, the situation when we have only one (even complex) sensor is a special case of a more general situation when we have few attributes whose values are computed on the basis of sensory measurements. Then, on the basis of these values, we can compute another attribute or a set of attributes that may be used for computing the decisions of agents. In this way, we obtain yet another example of a c- granule, which is a more complex example than that of a c-granule that represents a single sensor. This example of a c-granule represents:

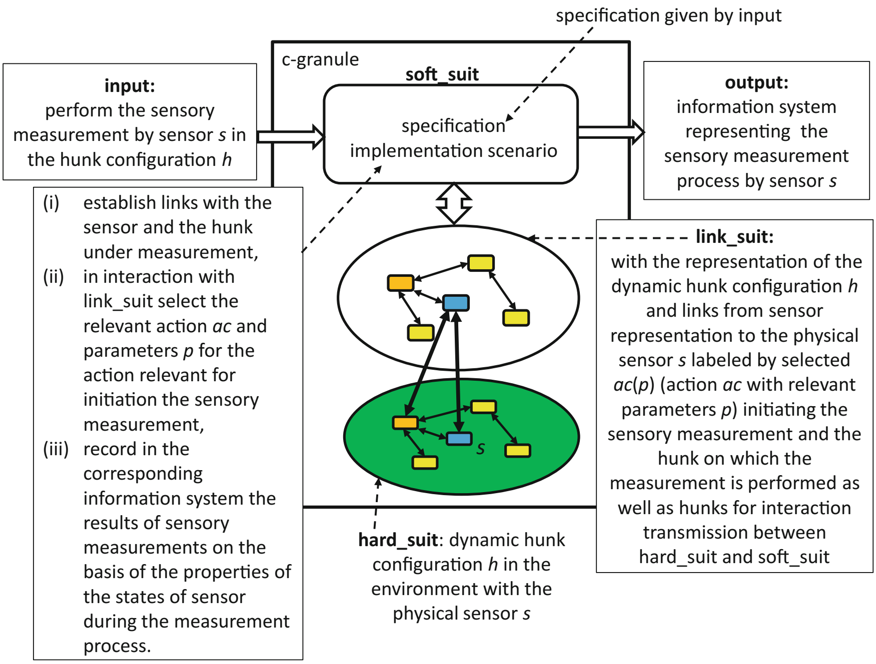


Fig. 28.10 Sensory measurement

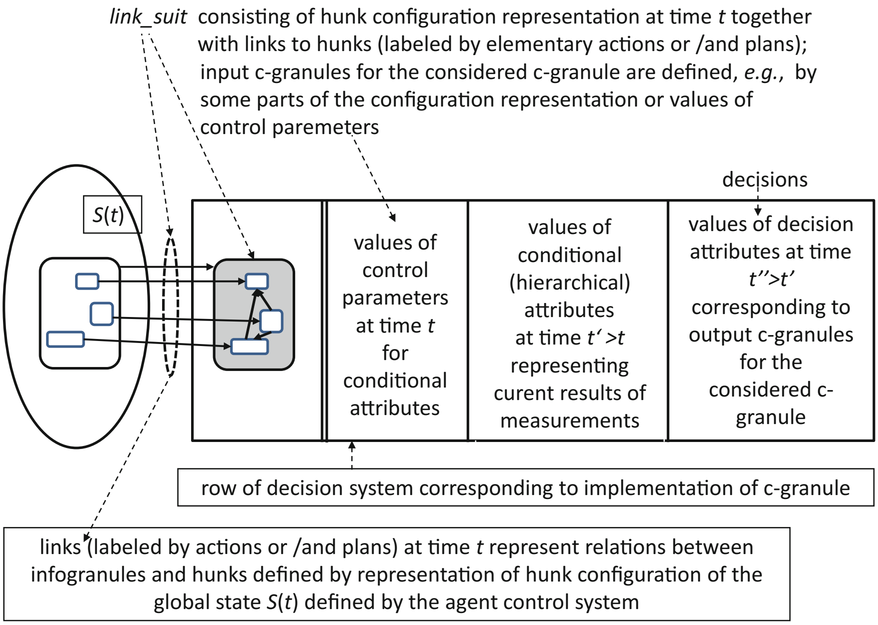


Fig. 28.11 Row of decision system corresponding to c-granule implementation

1. a set of control attributes for sensor measurements,
2. a set of conditional attributes, computed on the basis of sensor measurements,
3. a set of decision attributes.

This is illustrated in Fig. 28.11.

Of course, the situation presented in Fig. 28.11, where we have to deal with measurement within a single environment of objects, is by itself a simplification. In practice, we usually have to cope with multiple environments of objects. Information about them is presented in the form of a table with multiple rows (e.g., for different time intervals), as in Fig. 28.12.

What is more, generally, a single table is not enough in practice. Such tables are very well known in relational databases or Pawlak's information systems [386, 391]. In the case of such applications, models with only one table are usually not optimal for modelling the functional logic of a base. Neither are they optimal for many types of computations (e.g., those induced by SQL queries in transactional databases). It is then necessary to use multiple tables join operation using complex relationships (constraints) [456].

An analogous situation arises in the case of decision systems. Here, the structure of many systems, in particular their hierarchical structure, "cooperating" with each other is of paramount importance. Intuitively, we can imagine such a structure as a tree. In this tree, leaves correspond to data from sensors ("sense organs"), while subsequent "tree branches" correspond to join of tables that store the feature vectors

a_1	a_2	a_3	a_4	d_1	d_2
a	1	2	1	1	3
a	1	2	3	1	4
1	2	3	9	1	3
a	1	2	3	1	3
a	6	5	3	0	3
9	7	9	?	0	4
2	9	7	9	0	4

...

Fig. 28.12 Table representation of the interactive decision system

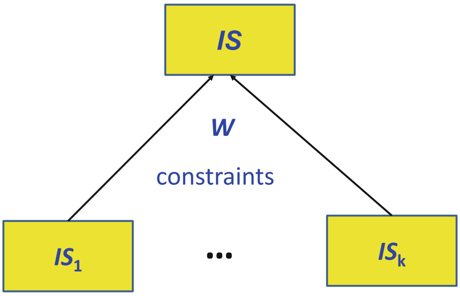


Fig. 28.13 Join of information systems

of structural objects, which satisfy the join conditions. Finally, there is the root of the tree which represents the main table that integrates knowledge from lower levels. A schematic diagram illustrating the "join" relationship is shown in Fig. 28.13.

Join tables in practical applications are not limited to only two decision systems, but they are usually composed of multiple decision systems. A network of decision tables joined by multiple constraints may serve as a basis for the implementation of complex c-granules, which perform complex interactive computations in the environment. An example of such a complex situation is illustrated in Fig. 28.14.

The above-mentioned examples of c-granules are largely based on the concepts of an information system and a decision system which were developed by Pawlak [386, 391]. Therefore, if we were to simplify the whole procedure, the essence in the construction of such c-granules was based on adding layers of link_suit and hard_suit to infogranules derived from tables and/or networks of join of tables. In a similar fashion, one may construct examples of c-granules using other approaches to AI. For example, this may include neural networks, decision trees, genetic algorithms, or specifications expressed in fuzzy logic.

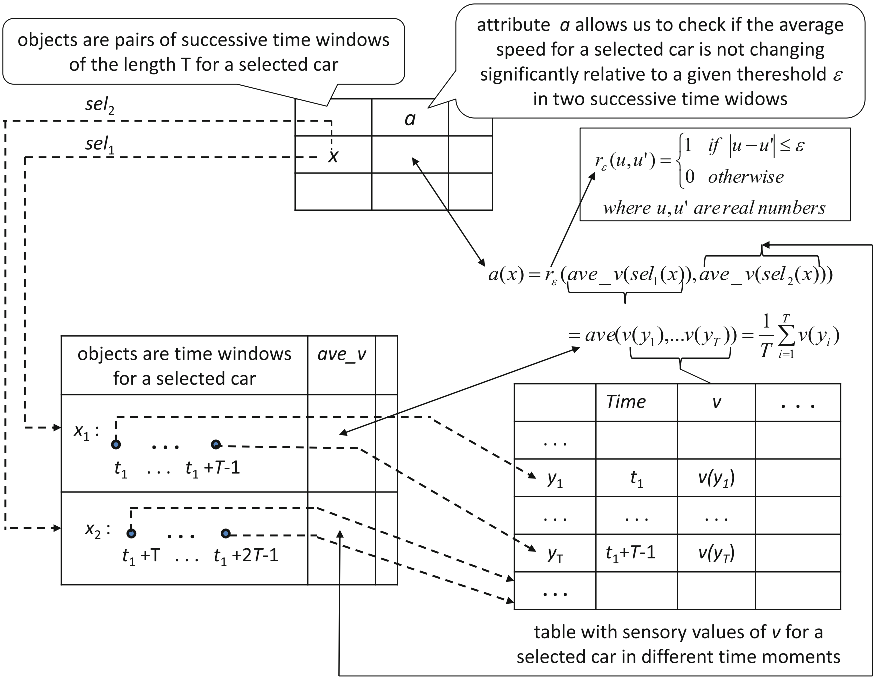


Fig. 28.14 An example of a complex attribute

28.2.4 *Interactive Granulation: Aggregation, Decomposition, and Transformation of C-granules*

28.2.4.1 Transformations of C-granules by Interactive Granular Control

The functional framework of c-granules discussed in Sects. 28.2.2 and 28.2.1 shows that a c-granule may transform different types of c-granules (including input and control c-granules) into other c-granules.

For example, output c-granules can become input c-granules or control c-granules for other c-granules. In general, the processes of creation (using granulation or degranulation), transformation, and decay of c-granules can conventionally be called *interactive granulation* or, in short, *granulation.*

The examples of simple c-granules were presented in Sect. 28.2.3. There are, however, also some instances of increasingly complex agent's c-granules. Their construction is based on information systems and decision systems. In the previous sections, we started off with describing simple sensors (ones that may measure the value of a single attribute) and then, moved on to discussing the vectors of attributes, tables, and finally, a hierarchical network of join tables (cf. Fig. 28.14).

This method of constructing increasingly complex c-granules based on a simple implementation of information systems or decision system is one of the most elaborated ones. Depending on the degree of her/his sophistication, an agent can be equipped with different methods of granulation.

Agents, through their interactions between control system and c-granules can also activate, execute, modify, and deactivate other c-granules. These interactions can also circulate their properties (transmit and receive). Therefore, the typical basic methods of granulation include: component construction and aggregation, granule composition and decomposition, representation, access, selection, activation, execution and deactivation, communication (e.g., approximation of granules received from different sources or sent to other receivers) and modification.

The process of **granulation is carried out in the physical world**. It means the interactions are understood as physical processes. These processes are subject to the laws of nature and limitations resulting from these physical laws, as well as to the constraints imposed by an agent through the interaction control. The existing laws of nature are very important—they **essentially decide about the "to be, or not to be" of the actual computing models** implemented in the real life. By the same token, operations on c-granules must also face the constraints which stem from both the laws of nature and any intention planned by an agent. More specifically, an agent may happen to improve and enforce additional constraints, which in some way may optimize the course of interactions against some criteria that are important to the agent. However, often, the constraints are not fully known to an agent (the owner of hunks). An agent is trying to learn them iteratively and then stores them in the form of appropriate c-granules.

Let us recall that constraints, understood as the laws of nature, are usually **ignored in theoretical computing models**. As a result, these models, designed without any regard to the laws of nature, are often unusable in reality. It is worth to note that limitations derived from the laws of nature do not always impede the flow of interactive computation.

Quite the opposite, the **laws of nature often contribute to the optimization of solutions produced by interactive computations**. In other words, these limitations happen to make physical processes asymptotically converse to optimal hunk configurations. Classical examples of this kind of processes include crystal, chemical reactions yielding more and more stable substances or the evolution of living organisms.

In the approach proposed in this book, we assume that an agent is equipped with the *c-granules of interaction control*. This c-granule is composed out of the c-granules supporting the agent's control in interactive granulation processes. Of course, interaction control c-granules are themselves subject to the laws of nature. However, on the other hand, they should follow the constraints and recommendations of the agent's control.

The above remarks concerning the interaction control for c-granule processing are summarized in Fig. 28.15.

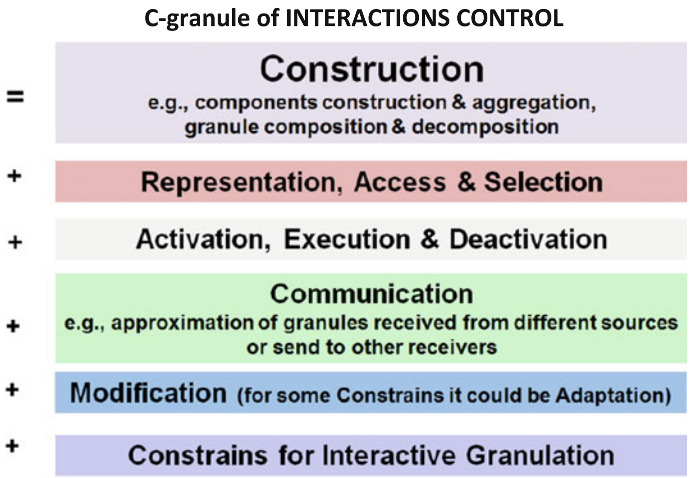


Fig. 28.15 c-granule of interactions control

An agent, having an advanced system of the interaction control for c-granule processing, carries out an INTERACTIVE GRANULATION process, i.e., she/he creates and processes c-granules. INTERACTIVE GRANULATION processes depend also on processes occurring in the environment and on "material" processes assigned to INTERACTIVE GRANULATION and accessible to an agent.

This remark can be summarized in a metaphorical equation (cf. Eq. 28.1).

$$\begin{aligned} INTERACTIVE\ \ GRANULATION = & \\ & c - granule\ of\ INTERACTIONS\ CONTROL\ + \\ & linked/input/output\ \ c - granules\ \ + \\ & environmental\ processes. \end{aligned} \tag{28.1}$$

This metaphorical equation is presented only for the purpose of a quick review. Hence, it does not reflect all nuances contained in the INTERACTIVE GRANULATION concept.

In order to deepen the intuition of this important concept, let us notice that the interactions that model basic aspects of the human thought processing constitute a starting point for the construction of IGrC models.

Our metaphor constitutes a simplified perception of a human "thought" as an INTERACTIVE GRANULATION process within the "brain" of an agent *ag*. Let us illustrate the idea by the example of interactive granulation, which leads to the creation of a name for interactions preconceived by an agent *ag* (cf. Fig. 28.16). To do this, we assume that an agent possesses a metaphorically understood "brain", the states of which are represented by configurations of hunks that constitute that metaphorical brain. She/he is involved in interaction processes (IP) with the

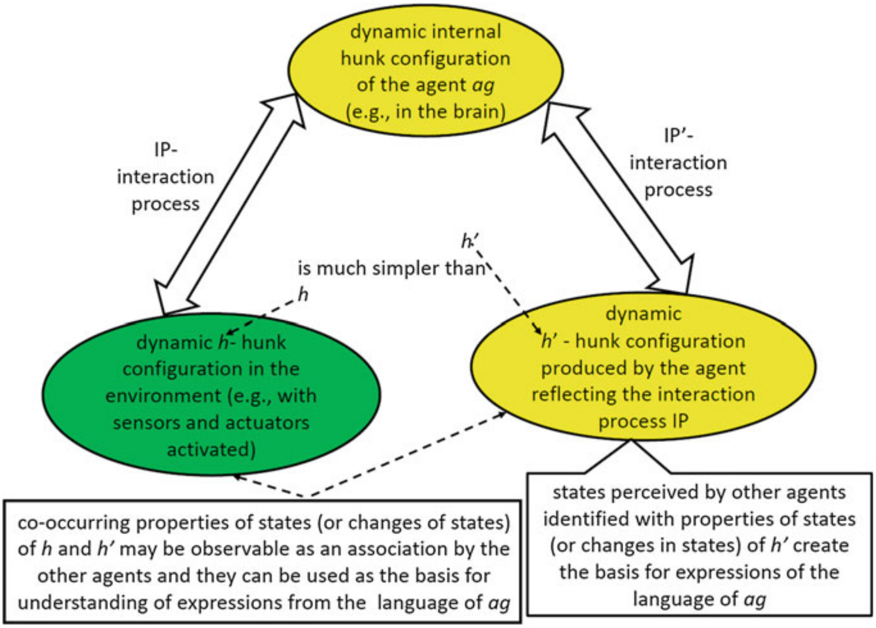


Fig. 28.16 Name creating as an example of an interactive granulation

surrounding environment (cf. Fig. 28.16), where an agent *ag* is observing a hunk *h* in a configuration recognized through her/his physical world glasses. In effect, the brain of the agent *ag* launches an interaction process IP' with the environment. IP' is a reflection of the previously mentioned IP. IP' then initiates a hunk configuration h' which is an image of h through the agent's *ag* physical world glasses. The hunk h' constitutes a compressed form of h and can be, in a sense, considered as a name for an expression in the *ag*'s language for the h hunk. The co-occurrence of (properties of) hunks h and h' may be perceived by another agent ag' and can be used as a basis for creating names of h by ag' with the usage of (properties) of h'. The agent's perception of hunk structures, constituting expressions in her/his language, leads also to the creation of the grammar of these languages. An agent learns these grammatical rules through interaction with the environment. In a sense, such a situation is similar to the concept of a triangle of reference [10, 371, 421]. It is depicted in Fig. 28.16.

28.2.4.2 Aggregation and Decomposition of C-granules

The following paragraph illustrates the conceptions and attitudes towards specific examples of interaction-based granulation—aggregation and decomposition of c-granules.

The examples of c-granules presented, among others, in Sect. 28.2.3 constitute the starting point for our present considerations. These examples include both very simple c-granules, as well as more complex ones. The fairly simple examples of c-granules include attributes that are easy to calculate, such as:

cg1 Sensor-based measurement carried out by an agent (e.g., of a temperature, in an easily accessible place).
cg2 Results of interactions remembered by an agent that took place in the past (e.g., injuries caused by exposure flame).
cg3 Reaction that stimulates agents to escape from a place in which they identified risks that are well known to them (e.g., bacteria running away from a location in which it has recognized a dangerous substance).

In the examples provided above, the term of an "attribute," used to refer to a peculiar type of a c-granule, is understood in a wider scope than in the typical approaches towards the information systems [384, 386, 389–391]. The difference stems from the fact that the present case contains more components on the basis of which the attribute value within a given object can be computed. Precisely speaking, in a classical approach to information systems, this value is by definition represented by the following components:

- Indication of an object described by the value,
- Indication of an attribute described by the value.

In the case that is being discussed, other indications exist, including:

- Indication of the mechanism used to compute the value of an attribute,
- Indication of the parameters that control the computation mechanism,
- Indication of the mechanism used to reach an object and prepare it for computations,
- Indication of the mechanism used to carry out the necessary measurements and computations.

Of course, the level of difficulty depends on the skills of an agent who owns a given c-granule. The more complex and sophisticated the agent's control, sensors, and actuators are, the greater number of c-granules may be interpreted as "simple" by her/him.

Complex c-granules representing complex processes include:

cg1' attributes of risk assessment related to the behavior of the financial markets [176], or attributes determining the conditions for undertaking the decisions related to the techniques of acquiring the resources (e.g., decisions made by people or bacterial colonies),
cg2' "thoughts of an intelligent agent," e.g., covering the technologies of constructing modern, geothermal power plants,
cg3' plans and techniques used to implement a complex CSE-type projects and the discovery and development of new AI technologies.

The above examples illustrate some significant differences between the complexity and variability of simple and complex c-granules. Considering the fact that complex c-granules are created through the aggregation of simpler c-granules, the above examples also present the scale of our lack of knowledge and effectiveness within the scope of c-granule aggregation.

It is worth to note that **in our approach, aggregations of c-granules are also treated as larger c-granules which may interconnect input c-granules with the required c-granule types in order to create c-granules of a more complex nature.** The c-granules created in this way represent aggregations. They may constitute very complex processes which in turn may lead to the creation of complex c-granular networks. In some sense, the agent's mind is treated as a tool for modelling the functioning of her/his brain.

The grammar of a natural language may be considered as an aggregation of c-granules. Usually, such aggregations are also quite complex. Boolean operators constitute a simpler example of an aggregation. They join the sentences with the use of logical conjunctions, such as: conjunction, alternation, implication, negation, or modal operators. Thus, for example, the agent may have a c-granule at his disposal, where this c-granule realizes $\otimes$, the operator for the aggregation, which for the input data of the c-granules (with acceptable input c-granules type for $\otimes$): g_1 and g_2, creates new $g_1 \otimes g_2$ c-granule, with the latter one being an aggregation of the former two. The $\otimes$ symbol used in this context may be understood as an aggregating c-granule, thus referring to one of the binary logical conjunctions listed above.

Another example of a c-granule aggregation, which is quite popular in relational databases, is visible in the join table joining operations.[10] They are based on the connection of lines from two tables that go in line with the specified conditions.

Considering the above examples of aggregations, we may differentiate between aggregations that are fairly simple to execute for a given agent and ones that are easily repeatable (e.g., logical conjunctions, join tables), as well as aggregations that are very complex and difficult to repeat (such as the constructions of implementation plans in the case of complex CSE-type projects or the development and creation of a paper which discovers the Einstein's general relativity theory).

Aggregations of c-granules enable an agent to increase the effectiveness of her/his reasoning by making it possible to process, through the agent's control, the c-granules that are more complex, to which the control may refer, given the fact that the control does not have to refer (multiple times) to each c-granule that has been aggregated.

In some sense, the *decomposition of c-granules* is an operator which is opposite to aggregation. For example, the decomposition of a text into chunks comprehensible to an agent and/or separating some fragments needed by an agent from a text (providing the answers to some burning questions) may all be seen as examples of decomposition. Another example is an attempt to solve a problem by an agent, with the use of the "divide and conquer rule" method.

---

[10] https://en.wikipedia.org/wiki/Join_(SQL).

Analogous to aggregation, decomposition is also treated as a c-granule. Some c-granules of decomposition may be relatively easy to apply, whereas others may be very complex and difficult to use, like in the case of aggregation.

Decomposition supports the agent's control in discovering and/or separating other c-granules from a specific c-granule, in the case of which the construction is not performed intuitively using parts that contribute to the final shape of that c-granule. At this point, we should remember that an agent does not always have a capacity to reproduce the building blocks of an aggregated granule.

It should be emphasized that a skillfully applied aggregation and decomposition may lead to the emergence of very subtle c-granules that may be optimal in terms of the required criteria. However, the creation of such c-granules is usually very difficult. Sometimes, this task may be simplified by the skillful application of knowledge which is field-specific and related to the given activity. However, this does not guarantee the efficiency of an optimal granulation adjustment (to the specific profile of granular application). The next sub-section deals with this problem in a more comprehensive manner.

28.2.4.3 Efficiency of Optimal Granulalation Adjustment, as One of the Most Important Goals of IGrC

In order to clarify the term "efficiency of optimal granulation adjustment," we are going to use the example of a typical problem with tailoring the level of detail related to planning the specific requirements of a given project. The nature of the problem is finding an optimum compromise between the effort needed to create a project implementation plan and the effort undertaken to implement, specify and adapt this project (often through costly changes) so that it fits the changing conditions related to the implementation of any initiative.

We shall remind the reader that in Sect. 28.1.5.2, "Integration of Acting and Thinking by Links using Linking Interactions," under LI-2 condition, we noted that there are no options and justified reasons for constructing correct, over-detailed plans of interaction for complex tasks (e.g., planning a route through traffic congested midtown of a large city). The lack of such option and justifiability includes the consequences stemming from the following facts:

A. Each agent **has a limited number of resources at her/his disposal**. In other words, none of the agents can afford wasting the available resources through the construction of over-detailed and irrelevant (in terms of the agent's needs) plans.
B. Regardless of the level of detail, in the case of fairly complex initiatives, **no plans made by an agent can take into account all changes occurring during their execution**.

This is also an important conclusion resulting from the practical experience gathered during the process of implementing the CSE projects, which may be referred to as the metaphorical laws of physics, including the Heisenberg's uncertainty principle[11] or the second law of thermodynamics.[12]
For example, due to the incompleteness of the information, it would never be possible (and there is no need) to plan the implementation of a project on an overdetailed level. At some certain level of detail, further elaboration is unjustified since throughout the process of implementing a project, we get to know the details which were practically unpredictable at the stage of constructing the interaction plans (and these details have a relevant impact on the way in which we fulfill our plans). However, **in order to successfully implement a project, it is important to set and follow the basic architectural adaptation rules of implementation**. The examples of such rules for CSE projects are discussed in the further chapters of Part IV, namely: "CSE: Case Studies."

On the other hand, the **interaction plans that are being developed cannot be too general** due to the fact that plans that are too general are of no use and they often lead to catastrophes within the scope of the project implementation.

The remarks presented above lead to an important and difficult to realize observation regarding the techniques and forms of creating the interaction plans for complex tasks, including the CSE-type projects. This important fact may be expressed as below.

By planning the execution of complex tasks, it is reasonable to create, in an effective way, a compromise between the effort needed to go over the next steps of elaborating on the interaction plans that are being constructed and the effort needed to examine the environment, adapt the plans that are to be implemented, and introduce necessary changes to them.

Creating a compromise between the costs of creating elaborate plans and the effort needed to adopt these plans constitutes an important example of practical uses of the computational models for interactive granular computations (including those implemented with the use of c-granules).

Let us assume that the ag agent's task is to solve some problem, which we refer to as P. According to our approach, an agent will do her/his best to construct a solution to the problem by using relevant interactive granulation processes. Usually, in the case of complex, practical problems, numerous potential granulation methods, that may have an impact on solving the problem P, exist.

On the other hand, an agent has no sufficient resources at her/his disposal, required to search and verify all potential granulation methods in order to identify the optimal *granulation*. What is more, in practical terms, such a search may turn out to be much more costly than the application of granulation that is far from being optimal. Sometimes, well-experienced managers—in such situations—claim that **not taking a decision at the right time is the most costly decision that can be made.**

[11] http://en.wikipedia.org/wiki/Uncertainty_principle.

[12] http://en.wikipedia.org/wiki/Second_law_of_thermodynamics.

In this way, we can see that an agent has an overriding problem with which she/he needs to deal, namely:

Problem of Efficiency of Granulalation Adjustment.

The nature of the problem is the difficulty to indicate the most effective granulation method which will meet the specific profile of the problem P and comply with the current conditions of the agent ag—this will stimulate an agent to construct (at an acceptable cost level) and systematically adapt the "optimal" (at a level which is satisfactory for this agent) interactive granulation that will enable her/his to "optimally" (at a satisfactory level of approximation in terms of the optimum solution) reach the solution of the problem P.

One of the basic goals pursued by IGrC is to deliver the tools that could provide support for the effective solving of the "Problem with the Efficiency of Granulation Adjustment."

In this context, one of the special cases is related to the effective construction of interaction plans, with the use of optimally tailored granulation. In simple terms, this means that—using a limited number of measures—we need to answer the following question: **what is the proper level of accuracy (granulation) for the interaction plan that has been developed, which would maximize the chances for its effective implementation and compliance with the given class of requirements?** In order to answer such a simplified question, we may apply the following intuitive recommendations, expressed in the language of the IGrC models that are being examined:

IP-1 Level of detail for a given interaction plan depends on the timescale of an agent and the complexity of the organization, which is the subject of the plan. The greater the scale and/or organization are, the less specific the plans of the tasks that are to be executed should be. Such an overriding interaction plan should be—according to the needs—adapted, specified, and/or modified on the basis of the progress in its implementation, made within a given scope of time, and/or the fulfillment of this plan at lower levels of organization. The level of detail of an interaction plan should be the highest for plans that are to be implemented in the nearest future. Then—according to the needs—the level of detail for the subsequent plans may start to decrease.

IP-2 The more knowledge and technology we have at our disposal, which may be used in order to predict the course of a task that has been planned, the more we can afford to develop plans with a greater accuracy.

PI-3 Structure and decomposition of an interaction plan that has been developed, related to the execution of a complex task, should account for the conditions that may accompany the execution of this task. Particularly, the decomposition of a task and its division into smaller parts should take into account the possibilities within the scope of credibility and ability of the "subcontracting" agents, who are to support the partial tasks. It should be also remembered that:

i. The more "wisdom" (and knowledge) and the better technology are available to "autonomous intelligent subcontracting agents," performing the partial tasks, the more general the plan of the main task may be.
ii. The smaller the possibility of communication and cooperation there is, for the agents who carry out partial task, the higher the level of detail, contained in the specification of conditions required for the execution of this particular task, should be. In particular, this concerns the interaction rules that boost the efficiency of interagent communication and cooperation.
iii. The more chaotic the conditions required for the execution of a given task are, the more detailed the specification of these conditions should be in describing both the interaction rules related with the flexibility and resistance of performing a given task in relation to the unexpected changes and the rules of interaction that govern the implementation of these changes.

The recommendations presented above, related to the intuition of dealing with one of the basic purposes of the IGrC, stem mainly from the practical experience, gathered through the implementation of the CSE type projects.

Certainly, developing proper algorithms that could facilitate the automatic generation of interaction plans, providing solutions to a given class of problems, would be quite beneficial in numerous applications. This is the basic and still unfulfilled dream of the AI enthusiasts. In order to achieve at least some approximations of solutions of this goal, IGrC language rules should be applied. In our approach, the language of IGrC is based on the key notions of a c-granule and linking interactions.

28.2.4.4 Problem of Computational Building Blocks for WisTech

> Like the abstract theory of syntax, formal logic without a powerful procedural semantics cannot deal with meaningful situations.
>
> Marvin Minsky [339]

In some sense, the "Problem with the Efficiency of Granulation Adjustment" features a special case of a more general problem. The problem is that when one has obtained a proper specification of a given problem in a natural language (or in the language of an agent, when the agent uses vague concepts), it is hard to construct interactive plans for computations that could lead to the solution of this particular problem. Of course, it is expected that some optimization criteria for interaction plans will be satisfied.

At the current stage of IT application development, small, simplified chunks of the natural language are used as an interface. These chunks may perform a simplified processing of the semantic content, expressed using the natural language (e.g., in books) and/or via the content available on the Internet.

These small linguistic chunks are usually limited to very simple grammatical rules that aggregate simpler linguistic structures. The attempts aimed at processing the semantic content of complex texts still seem to be unachievable for the contemporary IT technologies [82]. According to some scientists, developing computer software that could fluently use the natural language, which would be impossible to differentiate from the human language, would constitute a great milestone in the development of the AI technology.

This problem was directly formulated back in 1950 by A. Turing [510] and is well known under the name of the "Turing test".[13] In fact, the idea of the "talking machines" emerged much earlier. For example, back in 1637, Descartes, in his book entitled *Discourse on Method and Meditations on First Philosophy* [111], was rather skeptical about finding a potential solution to the problem of natural language processing in the future:

> How many different automata or moving machines can be made by the industry of man [...] For we can easily understand a machine's being constituted so that it can utter words, and even emit some responses to action on it of a corporeal kind, which brings about a change in its organs; for instance, if touched in a particular part it may ask what we wish to say to it; if in another part it may exclaim that it is being hurt, and so on. But it never happens that it arranges its speech in various ways, in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do..

Despite the fact that the words above are around 400 years old, still, many AI practitioners share these Cartesian views, that by no means it is possible to create machines that could overcome the barrier of the freedom of communication, with the use of the natural language.

Currently, numerous research centers are intensively working on closing the gap and transcending the barrier related to natural language processing. At the same time, a vast majority of scientists shares Turing's view that passing the Turing's test would constitute an empirical confirmation of the basic, expected feature of AI—the intelligent usage of the natural language.

For many scientists working on AI, the quotation from Minsky is stimulating [194, 339]—according to him, the key to success is to develop the "powerful procedural semantics" which can "deal with meaningful situations." From the very start of the research in this field, a search for the basic building blocks, that could be processed within the framework of the **powerful procedural semantics** [194, 339], has been carried out.

One of the first proposals regarding such "building blocks" emerged in a form of a "linguistic variable," which is a term introduced by Zadeh, back in 1973 (cf. [548]), along with the term of Information Granule [552]. The search for such building blocks was continued, within the frameworks of a variety of attitudes towards granular computations [394]. A lot of researchers agree that further research in this field is required. Such views were expressed by the Turing Award recipient, L.Valliant.[14] He claims that:

> A fundamental question for artificial intelligence is to characterize the computational building blocks that are necessary for cognition.

[13] http://en.wikipedia.org/wiki/Turing_test.

[14] http://people.seas.harvard.edu/~valiant/researchinterests.htm.

One of the prospective goals pursued by IGrC is to deliver the computational models that would bring us closer to the effective processing of the natural language, particularly within the scope of complex vague concepts processing, taking into account the approximate reasoning about these concepts. The concept of a c-granule described in this book and the concepts proposed to develop the IGrC models constitute actions in that direction, which are continued by the works initiated in [248–250, 252].

28.3 Adaptive Learning of Concept Approximations to Support Agent in the Decision-Making Process

28.3.1 Learning Through Memorizing "Important" Patterns

In order to clarify the mechanism of learning the approximations of concepts that support the decision-making process in which an agent is involved, we may use an example described below.

Let us imagine that an agent has no knowledge of the term "wasp" and, at the same time, she/he knows the term of "the pain in the hand" (occurring, for example, after a wasp sting). More an agent notice and record the fact that a cause-effect pattern occurs quite often (insect on the agent's body $\longrightarrow$ pain in the place where the insect is), that pattern would be more and more efficiently recorded in the agent's memory. In some sense, the agent is going to be a subject of the classical Pavlov's conditioning.[15]

The quicker an agent records such a relationship, the greater need she/he feels to defend herselves/himself from that pain and the earlier she/he is able to judge the situation while being in an environment in which there might be a risk of being stung by a wasp. Thus, adaptive judgment plays the key role in developing a mechanism that makes it easier to manufacture and use new terms by an agent.

By its very nature, the mechanism is to identify, select, remember and recognize either important (in the context of assessing the adaptive judgment) processes that take place in the environment or the anomalies in regard to the "normal" status within the agent's perception. Thanks to that mechanism, an agent may construct more and more perfect approximations of a term which is important to her/him. For example, once a relationship is established by an agent ("insect on the body of the agent" $\longleftrightarrow$ "pain in the place where the insect is"), some other relations may be noted ("something buzzes and flies around the agent" $\longleftrightarrow$ "insect on the agent's body").

An agent, by gathering more and more experience, may create c-granules/attributes that are more and more perfect, which enables her/his to recognize insects that may cause bad consequences.

[15] http://en.wikipedia.org/wiki/Classical_conditioning.

Thus, the classifiers constructed by an agent (c-granules) may be more and more sophisticated.

These classifiers may be used by an agent to recognize some important properties of the environment—the risk of trouble and/or the possibility of attaining additional resources. In other words, when one refers to the agent's classifiers which support the mechanisms of the "adaptive judgment," it means that an agent, by observing the environment, interprets the processes occurring within the surroundings and on the basis of the interactions, verifies whether some term-related requirements are met at a satisfactory level (including conditions related to the situations which are dangerous for an agent). Usually, these include complex, vague concepts which are time-variable and which may be approximated solely in an iterative way by an agent, depending on her/his needs.

When a term associated with danger is present (e.g., an aggressive wasp emerges within the agent's surroundings), we may additionally assume that an agent has proper patterns of plan regarding the further development of the situation, such as:

a. an agent does not undertake any action and suffers (according to her/his hierarchy of needs and beliefs),
b. an agent scares the wasp away and avoids the consequences.

Using her/his adaptive judgment (assuming that no other important events occur—e.g., a car approaching an agent), an agent will initiate "plan b" which entails the initiation of some action (e.g., scaring the wasp away). Then, an agent would try to execute a proper action. By initiating that action, an agent expects a new term to be carried out (e.g., the wasp goes away).

After the action is initiated, it may turn out that it was not successful. Then, depending on her/his knowledge and emotional status, an agent may change her/his decision. Thus, at the beginning of the process, an agent has no specific plan of performing the actions described above. These actions are indeed related to the systematic perception of the agent's environment, which is being carried out through interactions (Fig. 28.2).

In simplified terms, an agent performs her/his actions in the PDCA cycle (cf. Fig. 28.9). One should remember, that the *execution* phase is not related to some complex plans, rather it is connected with some simple actions (e.g., steering wheel movements in a car). It is also not related to small adjustments implemented on the basis of the confrontation and the actual effects of the actions in comparison to the expected results.

It should be emphasized that in this simple example, featuring a wasp, we use some very complex, hierarchical terms which are tailored to the perception of the events, taking place in the agent's environment.

For example, "scaring the wasp away" is a complex set of hierarchic plans of interactions with the environment, developed by an agent. Once an agent observes and evaluates, using her/his *adaptive judgment*, any danger that seems to be more impending, a given interaction plan may be quickly replaced.

Another important observation that may be drawn from the analysis of the wasp example is related to the game theory. Thus, we may imagine that an agent plays a game with nature and on the basis of the repeatable patterns (e.g., wasp on a hand $\longleftrightarrow$ pain and suffering), an agent, in an adaptive way, learns to play the game. Thus, it is possible for her/him to adapt and recognize dangerous situations in the quickest manner possible and select optimal plans of counter-acting the results of these dangerous situations. However, the process of learning terms that could be used as the classifiers of less and more risky situations is very difficult.

28.3.2 "Machine-Like" Learning Process—Interactive Granulation

Agent's "thoughts" (represented by c-granules) are used by the agent's control mainly for the construction and judgment of the "for," "against," and "neutral" arguments before an agent undertakes or revokes an action. For example, the effect of such a judgment may be an assessment of the degree to which a given condition is satisfied.

The satisfiability of a given condition may stimulate an agent to design (action) plan, which fulfills of the agent's current priority needs (in this case, one of the priority needs was to construct an interaction plan). In this context, the agent's "thought" (c-granule) can be treated as an entity consisting of three layers:

(th-1) a representation of substantive content, context, and beliefs which build an agent's "thought" (i.e., a c-granule) and which can lead to the verification of conditions, specification of preparations, necessary to perform particular actions, and/or initiation of interactions (by an agent),

(th-2) a hierarchical network of "windows" (which change over time as a result of judgments issued by the control), through which the agent's control—by means of available physical instruments (sensors and actuators)—can perceive, explore, and record fragments of the physical reality and interfaces for the participation in interactions therein occurring (including those needed for the communication with other agents),

(th-3) the physical world—the processes observed and activated by an agent through interactions outside of the "windows" and the world, where a given agent operates.

For a more detailed elaboration on these intuitions—let us imagine that a blind agent (cf. Fig. 28.2) carries out her/his interaction plans, embedded in the current context of her/his hierarchy of needs. An agent expects her/his interaction plan to satisfy certain conditions (which can be represented by c-granules). Furthermore, let us suppose that an agent unexpectedly found herself/himself in an unfamiliar environment, i.e., outside the metaphorical "windows," where there are images that an agent could not predict. In this case, an agent is "surprised" to find herself/himself in an environment different from what she/he expected and described using her/his "thought," (i.e., appropriate c-granules).

Of course, finding differences between the effects of interactions—those expected (in the past environment) and actual (in the current environment)—is only possible on the basis of their appropriate interpretation (by an agent) and judgment. This is achieved by testing the similarity between the representation of the actual environment and the representation of the environment recorded by an agent as well as their similarity to the "expected" environment. This judgment (i.e., *judging c-granule*) is usually implemented by an aggregated network of interrelated appropriate auxiliary judging c-granules. These c-granules perform interrelated conformance tests and evaluate their course (some of the c-granules can be interrelated). In particular, judging c-granules use knowledge accessible to an agent in order to test the interpretation of the interaction effects. In this way, each **judging c-granule generates and aggregates arguments "for" and "against" a positive test result**.

An agent operates in an environment of which she/he has only partial knowledge, which changes over time. Accordingly, an agent using approximations of complex vague concepts, often found in the arguments "for" and "against" the satisfiability of these concepts, must expect conflicting interpretations and frequent occurrence of both kinds of arguments, "for" and "against" a positive test result. We assume that judging c-granules constitute a decisive mechanism in such conflicts as they take into account the context of the situation in which an agent is involved (including her/his current hierarchy of needs).

An agent, having observed the discrepancies between the situation in which she/he was unexpectedly found and the situation she/he had imagined, does not know **what her/his next step should be.** This is when the agent's control activates a "thought" (i.e., a c-granule) which represents this state of surprise and the agent's knowledge about this surprise (including her/his knowledge about the causes and motivations of the surprise). She/he also launches another thought (i.e., c-granule), which includes adaptive schemes of interaction plans (based on adaptation strategies of an agent) which enable an agent to immediately recognize the environment and update her/his needs. On this basis, an agent constructs the most effective interaction plan possible, driven by her/his priority needs towards a more accurate evaluation of the environment (e.g., by "tapping the cane," "smelling the odors" and other "features" of interaction with the agent's environment of activity) in order to take an appropriate action (plan).

The quality of the agent's activity in the environment depends largely on the quality of her/his sensors and actuators, as well as the agent's ability to aggregate information—in the form of complex hierarchical structures, representing *complex hierarchical attributes*. The result of this process of learning is an update of SWOT c-granule, which describes the agent's current situation and c-granules responsible for the agent's current hierarchy of needs. As a result, a c-granule which represents the hierarchy of the agent's priority needs may be formed, together with proposals for some further schemes of interaction plans, designed to meet the agent's priority needs.

As medical practice shows, first we have to perform the diagnosis. Based on the diagnosis results, we can propose the appropriate therapy. Analogously, we can proceed in the process described above, **a factor that determines the success of designing good quality SWOT c-granules and c-granules for the agent's current hierarchy of needs is the quality of complex hierarchical attributes, defined**

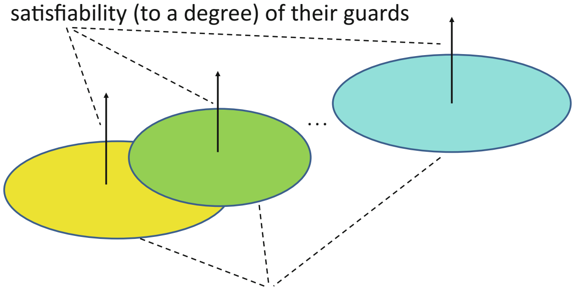


Fig. 28.17 An agent is playing a game with the environment. The game is defined by complex vague concepts labeled by actions. The concepts are used as "guards" for initiating actions, played by an agent against the environment. The actions are activated on the basis of the satisfiability degree of the concepts labelled to them. An agent uses interactions in the adaptive discovery of approximations of complex vague concepts. Moreover, an agent should learn relevant cause-effect relationships that occur among these concepts, e.g., for resolving the conflicts between different arguments "for" and "against" related with the satisfiability of different concepts

by appropriate c- granules. Therefore, the discovery and construction of these attributes are particularly important for the effectiveness of an agent.

Based on a cost-benefit analysis and other reasoning processes, an agent judges which option of a given interaction plan is worth pursuing and executing. The agent's control constructs c-granules that specify conditions which must be met for an agent to launch interactions and/or it constructs c-granules that define the conditions which must be met during the implementation of the interaction plan and at the time of its completion. An agent should have some recommended schemes for interaction plans in case there are some significant deviations from the conditions expected for the completion of these interaction plans.

In our review of the basic intuitions for using the c-granule concept in the context of WisTech, we emphasize the importance of interactions with the environment and their evaluation by identifying, satisfying, and adapting agent's needs, executed by the agent's control in PDCA cycles.

To simplify, the basic intuitions for using the concept of c-granule in the context of WisTech comes down to: **constructing, launching and adapting c-granules, judging the satisfiability degree of different conditions in various contexts of the interactions that are being triggered. This judgment is implemented by the construction of arguments "for" and "against" and—depending on the context—their further aggregation. The computing results obtained from judging c-granules should help an agent to construct, launch, and effectively adapt (according to an agent) interactions plans that lead to the fulfillment of the agent's current priority needs.** A brief summary of this idea is shown in Fig. 28.17.

The colorful ellipses in Fig. 28.17 depict complex concepts to which we assign their satisfiability degree as a judgment result of the process of judgment. These concepts define the conditions which are necessary to trigger an action (or, more generally, the interaction plans). The arrows which come out from the ellipses represent actions triggered by the judgment, confirming that the satisfiability degree of the concept behind a given action is sufficient. We assume that an agent has appropriate c-granules for judging the satisfiability degree of these terms, as well as c-granules judging on this basis how to choose the best action, from the agent's point of view, to be performed on the basis of these degrees.

When an agent is sensitive to the effectiveness of her/his actions, she/he may observe that if she/he uses more sophisticated and more effective (in terms of the satisfiability degree) methods to verify the conditions for action launching, which also take into account the agent's needs, her/his capacity to effectively operate increases. This underlines how important it is for an agent to discover and learn concepts and causal relationships that appear in her/his complex "games" with nature. The reward in this type of games can be the fulfillment of the agent's need (in a satisfactory way). An example of such a need may be an effective fulfilment and extension of the agent's priority needs.

It should be emphasized that the above presentation of the basic intuitions for using the c-granule concept in the context of WisTech is preliminary. The purpose of this outline is only to provide some foundations for a more in-depth discussion of c-granules and WisTech. Nevertheless, it shows how complex **a conceptual apparatus required for WisTech is (in particular when it comes to the concept of a c-granule and the diversity of c-granules in the context of WisTech).** For example, given the incompleteness of information itself, this apparatus should include concepts necessary for the specification, implementation, verification, adaptation, and development of such processes as:

a. identification of the current situation and the characteristics of its most important properties from the perspective of an agent's hierarchy of needs,
b. discovery of the appropriate context in which the current situation (taking into account the past and future, risks, costs, benefits or risks and opportunities) ought to be considered,
c. the recognition of any similarities to the previously analyzed situations and interaction plans that were implemented (both observed and simulated),
d. detection and differentiation of significant deviations from the expected and anticipated interaction results,
e. identification of any important unexpected changes in the situation recognized by an agent.

In real complex projects, it is also essential to use a complex language to make appropriate adaptive judgments regarding the above-mentioned concepts and their degree of satisfiability, in particular for the resolution of conflicts between the arguments "for" and "against." The language should express different concepts, used to induce new concepts, and the properties of the phenomena observed by an agent which will be, as predicted by an agent, useful in her/his further actions and cooperation with the societies of agents. This idea was described in more details by Pearl and Zadeh. It constitutes the motto of this book.

It is remarkable that all the attempts **made in the recent decades to create AI techniques, based on a fully automatic learning, representation and processing of concepts—in the case of real-life complex projects—have proven to be far from satisfactory**. Classic examples of such attempts are the research on reinforcement learning [490] and various flavors of natural computing [420].

This is mainly due to the difficulty of dealing with the complexity and variety of complex vague concepts that must be discovered, sufficiently well approximated, and efficiently processed. **Search spaces for a relevant approximation of such concepts are practically infeasible, i.e., existing data mining and AI methods as well as the current and foreseeable hardware technologies do not accommodate a successful search in reasonable period of time.**

For example, a classical approach to finding the reducts of attributes to generate decision rules (used, for instance, in the construction of rule-based classifiers) leads to the computationally expensive problem of the search for prime implicants [453].

More generally, we can say that any approach based on the ideas which constitute the so-called conceptual foundations of "logic gates and switching networks" in the architecture of modern computers, conceptually initiated in the mid-19th century by Boole [61, 62, 72] and developed in the 20th and 21st centuries as part of the research on Boolean functions [72, 359, 389, 442] and prime implicants and reducts [360, 450, 453], usually lead to NP-complete or NP-hard problems.

Given the Boolean functions of n variables and its huge size of space (the number of such functions is equal to is 2^{2^n}), there is a need to look for essential, with regard to a given problem (e.g., constructing a rule-based classifier), functions on subspaces of all Boolean functions. These subspaces first need to be discovered, and then, the Boolean functions relevant to the target problem may be searched for. These searching algorithms are of high computational complexity.

In order to better visualize the complexity of these algorithms, let us note that in the case of relatively simple and, at the same time, non-trivial practical problems, it is often necessary to use from a few dozen to a few thousand of Boolean propositional variables. On the other hand, the size of the space of n-variable Boolean functions for $n = 9$ is approximately $1.34 * 10^{154}$. Now, let us recall that the estimated number of hydrogen atoms in the observable universe is much smaller, with "only" about 10^{80} (cf. concept of observable universe[16]).

Thus, due to the high complexity of approximated concepts (e.g., using rule-based classifiers) and technological constraints, we should **expect no significant progress when comes to the problem, such as above. Such a problem needs full automatic learning of the adaptive classification of complex concepts, using traditional data mining systems, text mining systems or rule-based expert systems**, which follow the technological patterns of such systems as MYCIN [73, 209], logic programming [135, 282, 284], or **any other traditional AI technologies developed in recent decades** [422].

[16] http://en.wikipedia.org/wiki/Observable_universe.

As a consequence, there are considerable differences between the reality and expectations concerning the AI support in solving numerous practical problems (where complex vague concepts are present).

High complexity and diversity of concepts present in non-trivial application problems, means that in the foreseeable future, **real support in solving many practical problems** will prove possible mainly with the use of **interactive dialogue systems between the users (e.g., domain experts, scientists) and a system** [5, 27, 363]. In particular, this concerns discovering and processing approximation of complex concepts. As part of these IT application trends, there emerged new areas such as "cooperative problem solving"[17] or "collaborative intelligence".[18]

In interactive dialog systems, **user societies work towards joint problem solving, with the support of a network of mutually interacting computer systems and cyber-physical systems (CPSs).**[19]

The essence of the user's work in such a system is the fact that she/he would be able to formulate and refine hypotheses verifiable by the system using dialogue with other users and other cooperating computing machines. The verification of these hypotheses by the dialogue system can improve the discovery of a given problem and thus, provide the user with a better understanding of the problem as well as with tools for solving it. At the same time, the conceptual apparatus and knowledge stored in such computer systems should be developed through the dialogue with experts and other computer systems and possibly through planning and implementation, together with experts, of automated experiments. The functional specifications of such systems in any application field (e.g., discoveries in biology laboratories) are fairly well illustrated in the following [64]:

> [...] Tomorrow, I believe, every biologist will use computer to define their research strategy and specific aims, manage their experiments, collect their results, interpret their data, incorporate the findings of others, disseminate their observations, and extend their experimental observations—through exploratory discovery and modeling—in directions completely unanticipated.

At the moment, the modern knowledge and technology are not advanced enough to build complex, high quality interactive dialogue systems in the manner described above. There are however some attempts to implement different elements of this idea. One such example is RoughICE—an interactive dialog system for designing and developing ontology[20] [5, 27]. These activities require even further efforts to fully embrace the vision set forth in the quotation provided above and, consequently, guarantee their wider use in practice. **Further development of IGrC models based on WisTech provides a great chance for a significant progress in this area, particularly in the field of effective support for interactive problem solving**. It can be expected that the role of the central concept of WisTech-based IGrC, which is the concept of a c-granule, will drastically increase.

---

[17] https://en.wikipedia.org/wiki/Cooperative_distributed_problem_solving.

[18] https://en.wikipedia.org/wiki/Collaborative_intelligence.

[19] https://en.wikipedia.org/wiki/Cyber-physical_system.

[20] http://www.mimuw.edu.pl/~bazan/roughice/?sLang=en, http://protege.stanford.edu/.

28.4 Language of Agents and Communication

> These words [of Augustine cited in Chap. 26], it seems to me, give us a particular picture of the essence of human language. It is this: the individual words in language name objects—sentences are combinations of such names. —-In this picture of language we find the roots of the following idea: **Every word has a meaning. This meaning is correlated with the word. It is the object for which the word stands**.
>
> [...]
>
> But what is the meaning of the word "five"?–No such thing was in question here, only **how the word "five" is used**.
>
> [...]
>
> **the meaning of a word is its use in the language. And the meaning of a name is sometimes explained by pointing to its bearer**.
>
> L. Wittgenstein [539]

28.4.1 Private Language and the Agent's Infogranules

28.4.1.1 Meaning and Types of the C-granules

The motto of this chapter refers to Wittgenstein's concept [539], which perceives the meaning of a given expression in the following manner:

1. Pointing to physical objects (i.e., **pointing to its bearer**) in order to define the meaning of a given expression.
2. Setting rules for the use of a certain word in a given context of usage, e.g., with other expressions (i.e., **the meaning of a word is its use in the language**).

These concepts are developed by Wittgenstein in [539], where he introduced the concept of a "language game".[21]

Analogously, one may use Wittgenstein's approach to explain a given concept of an agent's language using a c-granule g. For example, we can do it in the following ways:

1. Indicating hunks (i.e., **pointing to its bearer**) using *linking interactions* from the c-granule g. In this context, the role of the "linking interactions" in the functioning of c-granules is presented in Fig. 28.9 and described in Sect. 28.2.2. What is more, Fig. 28.16 in Sect. 28.2.4 illustrates the process of recording the "pointing to its bearer" information by an agent (through *linking interactions*) and its role in the creation of c-granules.
2. Setting "the rules of the game" for "using" the c-granule g (or m-hunks present within the structure of a c-granule) in the "games" that are being "played" by means of the agent's control. For example, these "games" may include:

[21] http://en.wikipedia.org/wiki/Language-game_(philosophy), http://postmoderntherapies.com/word.html.

a. Game with the environment, presented by Fig. 28.17 in Sect. 28.3.2.
b. Game of cooperation and/or competition with other c-granules, presented in Fig. 28.7 in Sect. 28.2.1.

In Sect. 28.2.4, *viz.*, “Interactive Granulation: Aggregation, Decomposition, and Transformation of c-granule,” we have recognized the possibility that one type of a c-granule may be transformed into another type via interaction. The process of interaction may even lead to the emergence of completely new c-granules. During the process, a c-granule which acts upon other c-granules should have an ability to recognize a specific type of c-granules that stimulate the process of transformation. What seems of particular importance is that this very c-granule should be able to determine whether its own c-granule, the one which is to be transformed by the other c-granule, meets the conditions necessary for “entering” the transformation process.

For example, a c-granule which aggregates the vocabulary into logical sentences should recognize whether the “input” expression (phrases of a sentence) meets the conditions necessary for creating new, fully logical sentences (phrases of a sentence). In order to illustrate this, we may assume that we want a room painter to “paint a room in a building” and perform other related works. At the same time, we cannot ask the painter to “paint the numbers less than 100.” Until we define the way of interpreting “the numbers less than 100” in the physical world, such task has no sense and, as a consequence, cannot be executed.

This example shows that it is not always reasonable to aggregate random vocabulary items (phrases of a sentence), using random grammatical structures. Without verifying some specific properties of c-granules that are referred to as *“c-granule types,”* the transformation may not have any sense. This remark illustrates the role of the *c-granule type* term in the approach to IGrC, proposed in this book. To be more specific, we may say that c-granules may be aggregated and used once they fulfill certain requirements (particularly, when they are characterized by a proper *c-granule type*).

In practice, the relationship of a c-granule sub-type, which is a partially ordered set, is also very important.

In some cases, it is assumed that the relationship between various types is represented by a more-general mathematical structure. Sometimes, it is assumed that such a relationship is represented by a proper category [39, 296, 318, 329]. However, the relation is always transitive, both in the case of a partially ordered set as well as in the case when the relation is present in a form of a category.

Usually, a variety of c-granule types exists. In such situations, it is assumed that each c-granule has its smallest type, which is referred to as the *main type of a given c-granule*. Then, this c-granule also belongs to each of the over-types, that are located hierarchically higher than a sub-type. For the sake of convenience, we should assume that the name of a given c-granule within its main type evokes no ambiguity.

The agent’s control may use a c-granule selection mechanism for c-granules with identical names (or of the same type). For example, the results of such a selection may stem from the context in which particular names of c-granules are used and from the scope of information available on this c-granule. The control has an ability to perceive the sub-type relationship, for example, with the use of m-interactions and m-hunks (e.g., the m-classifier of the m-subtype).

Names and types of c-granules make it easier for the agent's control to properly process c-granules. The agent's control, before it starts to process c-granules, verifies their types in order to determine whether they can be qualified for a given processing operation. On the other hand, using of solely the names (instead of using the whole elements) of c-granules increases the processing effectiveness. This results from the fact that often, the agent's control may confine itself to using a sole name or other properties of a c-granule, the name of which is known, instead of performing, frequently cost-ineffective, operations aimed at recovering and processing the entire c-granule. What is more, it should be emphasized that when it comes to practical applications, the ability of accessing a c-granule through its name, which may be used as some kind of an indicator, is also very important. It directs us towards the memory available to an agent or the space where the form of a given c-granule, recorded by an agent, might be saved and stored. The idea constitutes a basis for the CWW paradigm (Zadeh [378, 550]).

For our convenience, we will assume that an agent is equipped with some special c-granules devoted to performing operations on other available c-granules and their names. These operations may include: adding a name, renaming, establishing a new name, recreating a c-granule with a given name, setting some property on the basis of the knowledge related to a given c-granule, etc. It is worth remembering that the approach presented in this book assumes that these operations are performed with the use of hunks.

Names, types, and other properties of c-granules may all be computed by means of proper decomposition operators.

28.4.1.2 Significance of Agent's Expressions: Pragmatics of the Past, Current Interaction Context and the Practice of the Future

> **On the Definition of Logic.** Logic will here be defined as formal semiotic. A definition of a sign will be given which no more refers to human thought than does the definition of a line as the place which a particle occupies, part by part, during a lapse of time. Namely, a sign is something, A, which brings something, B, its interpretant sign determined or created by it, into the same sort of correspondence with something, C, its object, as that in which itself stands to CIt is from this definition, together with a definition of "formal", that I deduce mathematically the principles of logic.
>
> Charles Sanders Peirce [395]

In the early modern period, between the 19th and 20th centuries, Peirce [395] noted that the term of the "meaning" of particular expressions does not simply point to a binary relationship between the properties and objects that bear those properties. He noted that this relationship is also largely dependent on the context of the situation, in which a given sentence is expressed (or in which the related message is transferred), and on the practical consequences resulting from the meaning of a given expression.

In this way, he prompted others to examine the term, which is known today as the Peirce's semiotic triangle.[22]

It is worth noting the fact that when it comes to the context of meaning of particular expressions, Peirce was particularly interested in practical aspects—in this way, he was able to initiate a trend in philosophy and logical science, known as pragmatism.[23]

We should also remember that both the agent and her/his c-granules are always placed in some context, which often has a great impact on the interactions that are carried out. Hence, it is very important for the agent's control to recognize and understand the *current contexts of the agent's interactions*. Usually, it is assumed that this context is composed of the following factors:

a. Current and important (according to the agent's judgment) effects of perceiving the environment in which an agent is active.
b. Important (according to the agent's judgment) results of perception that are associated by an agent with similar environments in which she/he used to be active (in the past).
c. Current priority needs of an agent (including the needs resulting from the interaction plans carried out by the agent).
d. Important (according to the agent's judgment) abilities of an agent within the scope of launching and/or modifying interaction plans.
e. Current results of the SWOT analysis and properties of a situation recognized by an agent.

From the point of view of the ongoing tasks undertaken by an agent, establishing the ongoing context of interaction in a precise manner constitutes a great challenge, which is connected with the notion of adaptive judgment, introduced in the previous chapters. The current interaction context is, to a great extent, a **consequence of the past interactions and those interactions that are taking place at the present moment**.

Examining the meanings of expressions used by agents in a given situation (including the way in which agents expressed these expressions), from the point of view of the current situation (including the scenarios pursued to reach a given situation), is related to the term of the *pragmatics of expressions*, which is understood as the **contribution of the "context" to the meaning of expressions**.

On the other hand, by considering the meaning of a given expression in the future context, one may interpret the meaning from the perspective of general **"practical consequences,"** stemming from the meanings of expressions.

In simplified terms, taking into account the above remarks, one may state that pragmatics is related more to the contribution of the past and present of a given **situation, and the way in which a given expression is used, to the meaning of this expression. On the other hand**, pragmatism is also linked with the more practical

[22] http://plato.stanford.edu/entries/peirce-semiotics/,
http://visual-memory.co.uk/daniel/Documents/S4B/sem02.html.

[23] http://plato.stanford.edu/entries/pragmatism/.

consequences that may occur in the future, **all of which contribute to the meaning of a given expression**.

At this point, it is worth to recall the fact that the terms pragmatics and pragmatism have broader meanings.

By pragmatics, we understand the field of semiotics which deals with the relationship between a symbol, the linguistic expression of this symbol, their user and the impact of the context on the meaning of this symbol. These dependencies are illustrated in a famous quote which is recalled at the beginning of an introductory article on pragmatics, included in the *Stanford Encyclopedia of Philosophy*[24]:

> When a diplomat says yes, he means 'perhaps';
> When he says perhaps, he means 'no';
> When he says no, he is not a diplomat.

28.4.1.3 Private Language of an Agent Determined by Semiotic C-granules

Agent's control, by using a variety of c-granules, may at the same time *interpret the meaning of these c-granules by referring to the meaning of other c-granules*. These interpretations may concern both the entire c-granules as well as their parts, which are also present in the form of a c-granule.

By interpreting the meaning of one set of c-granules, using other c-granules, we reach a point where c-granules are no longer interpreted with the use of other, already interpreted c-granules. These "final" interpretations constitute hunk-indications.

We do not assume that all c-granules have their own interpretations. One of the most important conclusions to be drawn from the above discussion is the fact that interpretations of c-granules may depend on numerous related aspects from the past, present, or future.

From the point of view of an agent who uses, e.g., a c-granule G_1 to achieve her/his goals, another c-granule G_2 may be very important. It is referred to as a *context-specific interpretation of a G_1 c-granule*. A c-granule G_2, in any current moment of an agent's activity, assigns to a c-granule G_1 and the current interaction context C of the interaction. This interaction is prioritized (according to the needs stemming from the current context of an interaction) links, which provide the agent's control with examples demonstrating such aspects of a c-granule G_1 as:

1. Patterns, that show an instance of a c-granule, which is a part of the c-granule G_1, within the C context. These patterns are embedded in situations (which comply with the context of using a given c-granule) and scenarios concerning the development of these situations, along with those results of the scenarios that may be potentially important for an agent (both in terms of increasing and decreasing the agent's satisfaction).

---

[24] http://plato.stanford.edu/entries/pragmatics/.

2. The interaction plan patterns that define potentially important ways of using the c-granule G_1 (or a part of it) in the context C, along with other resources available to an agent in the current context, which enable an agent to achieve her/his goals.
3. Expressions, that constitute "names" and alternative names of objects, described by G_1 in the context C, along with the contexts and ways of using and archiving these "names."
4. Interaction plans that enable an agent to recognize objects that are the G_1 interpretations, placed in the C context.

At this point, we should mention the SG c-granule of an agent, which constitutes an aggregation scheme of c-granules, featuring two c-granules, G_1, and G_2, at the input, where:

- G_1—is any c-granule with a contextual interpretation,
- G_2—is a contextual interpretation of G_1.

At the output of the aggregation scheme, SG is an integrated c-granule $SG(G_1, G_2)$, featuring two c-granules at the input: G_1 and G_2. This special SG c-granule is referred to as an *aggregation of a c-granule and its interpretation*. The $SG(G_1, G_2)$ c-granule, which results from the aforesaid aggregation, is referred to as a *semiotic c-granule*.

The name of the c-granule which is created as a result of aggregating another c-granule with its interpretation stems from the fact that such a c-granule encompasses basic aspects involved in a semiotic[25] examination, conducted from the point of view of an agent who is in possession of the above-mentioned c-granules. The examination includes such aspects as syntax, semantics, pragmatics, and broadly understood relationships, on the basis of which one may determine the references to the examples of a given term, represented by a c-granule, contexts in which the term appears, and ways in which it is used in a specific context, along with the resulting practical meaning, it carries for an agent.

The above digression, echoing the notions of pragmatics and pragmatism, illustrates a great potential significance of the term, introduced under the name of a semiotic c-granule.

The class of all semiotic c-granules used by an agent is referred to as a *private language of an agent*. Of course, the private language of an agent evolves with time.

The term of a private language, as we understand it, refers to the way it was used and examined by Wittgenstein, e.g., in [539]. It is characterized there in the following way:

> The words of this language are to refer to what can be known only to the speaker; to his immediate, private, sensations. So another cannot understand the language.

28.4.1.4 Infogranules

Analogously to grammar rules used in the natural language, when it comes to the private language of an agent, we may assume that an agent has a distinguished

[25] https://en.wikipedia.org/wiki/Semiotics.

class of aggregation operations, which aggregate semiotic c-granules in a way in which a new semiotic c-granule is formed. This type of aggregation is referred to as a *semiotic aggregation*. Semiotic aggregations are processing specific semiotic c-granules. Often, these constitute specific cases of more general *semiotic aggregation schemes*. These schemes operate on semiotic c-granule types rather than specific c-granules. On the other hand, once they are complemented with c-granules that have acceptable types at the input, a new integrated semiotic c-granule may be created.

Semiotic aggregation schemes may be very complex. For example, we may imagine an example in which an agent has a camera at her/his disposal, sensors of which record single pixels. Each of those pixels may be treated as a pair, consisting of a c-granule and its contextual interpretation. On the other hand, the aggregation of all these c-granules into one comprehensible and contextual interpretation is usually not a trivial task. Luckily, we also have many simple examples of semiotic aggregations that illustrate the intuition of the term.

Boolean operators, for example, constitute simple examples of an aggregation. They conjoin sentences with the use of either logical conjunctions, such as conjunction, alternation, implication, negation, or modal operators. In other words, once we have two sentences at the input, p and q, along with the interpretation of their meaning, it is easy to construct a $(p \;\; or \;\; q)$ sentence, along with a proper interpretation of its meaning, by applying a semiotic aggregation scheme to p and q. This example leads to other very important, in terms of their practical applications, terms. Namely, in the above example, we have used a fairly simple semiotic aggregation scheme.

Usually, an agent has a distinct class of semiotic aggregation schemes for the selected c-granule types of her/his own private language. These are general and repeatable computational schemes. Such a class of c-granule aggregations is called the *elementary grammar of the agent's private language*. For example, the elementary grammar for sentential calculus of some mathematical theory is composed of schemes in which mathematical sentences of this theory are connected using the Boolean logical conjunctions. Here, a mind-boggling question is how such schemes may be created as a result of evolution. Some works in this field are currently carried out by Steels [481, 482].

The c-granules which constitute a part of the agent's private language and which cannot be decomposed within the meaning of the elementary grammar of this very language are referred to as the *elementary c-granules*. The c-granules responding to sensor-descriptors are a classic example of elementary c-granules.

In the IT applications, a significant role is played by elementary c-granules or by the c-granules that were made out of elementary c-granules by means of hierarchical aggregation, using the elementary grammar of the agent's private language. This class of c-granules is briefly referred to as the *infogranules*.

28.4.2 Language of Communication Among Agents

The starting point for presenting the language used by agents to communicate is the private language of agents. Some mechanisms related to the emergence and development of this language are very useful for the co-creation of joint communication language with other agents. For example, Sect. 28.4.1.3 refers to the significance of recording, by an agent, causal relationships that are important for her/him, in order to create new c-granules (terms).

Intuitively speaking, when some processes perceived by an agent, in a repeatable way, leading to the same results, or similar judgment of these results, an agent may remember these results and associate the relationship that has occurred (cause $\longrightarrow$ effect) with some later phenomena. **Then, we may expect that once an agent records the presence of a similar result, she/he may, with the use of reasoning, predict a similar result**. Analogous mechanisms may be also present in the case of perceiving or observing hunks and interactions related to other agents. **If some behavioral patterns of an agent—the sender of a message—almost always lead to similar interaction patterns (e.g., concerning the behavior of a sender), perceived by other agents—the recipients of the message—we may speak of simple forms of communication**.

Thus, in the approach proposed in this book, simple forms of communication are executed in the following manner: the sender, using her/his interaction patterns, encodes the form and the content of a message and enables other agents to access this form and content. An agent who is properly receiving the message should be able to identify and recognize the form and the content of the message that is being conveyed to her/his. The receiving agent is particularly challenged when it comes to the proper reception of the message content. The basic problem here stems from the fact that usually, the "recipient" has no direct access to c-granule structures of other agents (including the "senders"). What is more, the receiving agent has no knowledge of (time-variable) contextual interpretations of c-granules possessed by the sending agent. The control of the "receiving agent" is trying to reconstruct the content contained within the message of the "sending agent" with the use of skills related to contextual interpretations of her/his c-granules.

Given the fact that every experience of an agent is different, in the case of complex terms it is almost certain that **the contextual interpretation of a c-granule recreated by the "receiving agent" is different from the contextual interpretation of a c-granule conveyed by the "sender."** This—in an obvious way—easily leads to misunderstandings and has negative consequences. Processes related to creating requirements specifications of the field experts, who will be the future users of a given CSE system, and of the analyst-engineers, who gather and document knowledge, both constitute classical examples of such misunderstandings. In such situations, a dialogue, conducted by means of certain interactions between agents, is very important as it minimizes the risk of such misunderstandings. Of course, the greater the scope of the joint contextual c-granule interpretation is, the more effective the dialogue becomes. The effectiveness, understood as the effectiveness of the communication activities, may be improved by a skillful approximation of messages by means of rough sets.

Negotiating the contextual interpretation of a given c-granule by its receivers and senders is an iterative process, which requires both agents to be involved in. The smaller the scope of the jointly agreed-on contextual c-granule interpretations is, the more difficult the process becomes. In extreme cases (when that scope is minimal), the process of negotiating the meaning of the content received by the "receiving agent" from the "sending agent" may be described in line with the below assumptions.

In some cases, the "sending agent" initiates interaction plans tailored to these situations, the result of which is the creation and/or the distinction of certain hunks—these hunks are briefly referred to as the *sender's artifacts*. The "receiving agent" perceives, places, and potentially remembers the co-existing situations and behavioral patterns of the "sender," along with the artifacts that were created and/or indicated by her/him. When the "receiving" agent considers the co-existence of situations and artifacts to be important, she/he records this relationship and assigns her/his own c-granule to it (also, defining its name and type). The greater the agreement scope of interpretation is, particularly when it comes to the *protocol of negotiating the joint understanding of names expressed in common communication language*, the more effective the process presented above, the aim of which is to agree on the meaning of the content sent by the sender, is.

Obviously, perception and learning processes related to such interactions, that are the components of the above-mentioned cause-effect relationships, require a lot of patience from both agents due to the iterative process related to the joint understanding of those relationships, encoded in semiotic c-granules. In other words, the communication ability, in our approach, consists of the following elements:

a. Agent's perception of the behavioral patterns (agent's expressions), represented with the use of c-granules related to the changes caused in the environment by other agents-senders (or hunks) with which the agent in question interacts.
b. Interpretation of the behavioral patterns (expressions sent by agents-senders) by an agent who is the recipient of these patterns.
c. Proper use of the properties inherent to these behavioral patterns (initiated either by the sending agents or hunks) in the decision-making process of the receiving agent and in the actions undertaken by that agent later on.
d. Properties of the behavioral patterns of the sending agent, observed by the receiving agent, which may modify the hierarchy of needs of the receiving agents and may, among other results, stimulate an action which is expected by some of the sending agents. The ability to identify certain hunks that are interpreted as agents and the co-existence of these agents with important properties of the co-existing interactions (e.g., interaction patterns) may play a significant role in the process.

Such communication mechanisms may be observed in the case of very simple organisms, such as bacteria, more complex ones, such as ants and bees, or even the most complex ones, such as the human beings. Moreover, an agent, by observing these causal relationships, may be willing to express and transmit her/his thoughts or feelings (c-granules) to another agent. In this way, an agent (hunk) may try to use her/his body language or other patterns of behavior to transfer the information about her/his intentions and actions to be carried out in relation to some other agent. On the other hand, an agent who is observing the messages being transferred with the use of the body language may consequently execute these actions. Besides, she/he can be also capable of remembering the body language and her/his own reactions to it. In this way, the sender may predict the actions that will be undertaken by the other agent (hunk). Efficiently performing this mechanism constitutes a foundation for creating the rules that could lead to the emergence and development of, e.g., a communicative protocol of agents. In simplified terms, one may claim that the communication between agents is carried out using to the following skills of the agents—both recipient and sender:

a. Ability to encode internal c-granules in a form of the transmitted interaction patterns (expressions produced by an agent).
b. Perception of the behavioral patterns (agent's expressions).
c. Interpretation of the behavioral patterns (agent's expression) in a form of c-granules by the receiving agent, who interprets the c-granules of the sending agent.
d. Internal processing of c-granules in order to undertake actions resulting from the conclusions, which were drawn from the interpretation of these expressions (c-granules).

One of the most important assumptions in our discussion of language, both private and communication one, stems from its evolution. The language transforms as experiences, knowledge, and needs of agents—users of that particular language—change. The communication language used between agents may enable them to create a new intelligent "team" agent, who will pursue her/his own needs within a given environment, along with the needs of its "constituent" agents, for as long as possible. Such an agent is referred to as a *meta-agent*. Thus, in such a situation, if we treat the agents that cooperate with each other as a single meta-agent, the language used to carry out the communication between these agents may be treated as the private language of meta-agents. This proves a significant role and impact of the private language concept in the process of understanding the language of communication used between agents who cooperate well with each other (agents pursuing a common hierarchy of meta-needs, which constitutes the basis for complex meta-judgment and meta-interaction, carried out in the environment of a meta-agent).

It is a great challenge to understand and create computational models that could simulate the transformation of: first, a society of agents into teams of agents who cooperate more efficiently and second, teams of agents into teams of meta-agents supporting the process of solving a given class of problems. Many attempts have been undertaken within this scope, in such fields as natural computations, learning without a teacher, or self-organizing computations [59, 110, 130, 136, 161, 184, 287, 293, 311, 343, 365, 405, 420].

Chapter 29
The Wisdom Equation and C-granules

29.1 Interpretation of the Wisdom Equation as a Basis of C-granules

In Chap. 13, entitled "Fundamental Precause of CSE Crisis (FP3C)," we described the interaction strategies for adaptive learning, with a particular emphasis placed on learning of "intelligent patterns of behavior" by the colonies of bacteria.

Within this context, in Chap. 13 we have mentioned about to the so-called Wisdom Eqs. 8.1, 13.1 [244, 245, 247]. It emphasizes, in particular, on the role of an agent's skill of adaptive judgment. It is a factor that makes it possible to connect the world of physical interactions with the world of knowledge available to an agent. According to our assumptions, this knowledge is encoded in a form of a hunk-configuration available to an agent.

In some sense, thanks to the prioritized links, the relationship between interactions and knowledge may also be noticed in Fig. 28.3. In this regard, let us recall the metaphorical Wisdom Equation of an agent (or a group of agents). We have assumed that this equation is related to an agent, who is active within the X domain of activity at a given moment (meaning a "micro" period of time which is short enough) or within a larger time period t. Then, for this agent, the metaphorical equation may be created in a form presented by Eq. 29.1, which takes into account the moment or the time period t:

$$\begin{aligned} &\mathit{WISDOM}\,[X,t] = \\ &\quad \mathit{INTERACTIONS}[X,t] + \\ &\quad \mathit{ADAPTIVE\ JUDGMENT}[X,t] + \\ &\quad \mathit{KNOWLEDGE}[X,t]. \end{aligned} \tag{29.1}$$

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_29

Referring to the B-1, B-2, and B-3 conditions analyzed in the Sect. 28.2.1, the individual components of the Eq. 29.1 may be interpreted as follows:

C-1. *KNOWLEDGE*$[X, t]$ is the knowledge which at the moment t is potentially available to the agent (e.g., in the agent's memory, on the external carriers, and/or from other agents). It is represented by m-hunks and m-interactions that implement proper knowledge structures, and by adequate mechanisms (e.g., of reasoning and/or learning) is used to develop new knowledge, with particular emphasis placed on the knowledge related to the domain of activity X. The knowledge in question concerns, among others, the following aspects:

- Interaction plans leading to understanding the meaning of knowledge recorded in the memory an agent, which is currently in use (particularly in reference to the hierarchy of needs and beliefs pursued by this agent).
- Interaction plans representing a variety of adaptive judgment processes using historical and current knowledge, as well as judging the ongoing perception processes.
- Methods for representing (and recording) c-granules of knowledge on physical carriers and in the agent's imagination.
- Physical carriers of knowledge used for transporting c-granules of knowledge, along with paths that make it possible to reach these physical carriers,
- Agent's interaction plans that enables her/him to gain access to the physical carriers of knowledge, decode data records and place proper c-granules of knowledge within the agent's imagination,
- Interaction plans for adaptation of implemented plans and/or generation of new interaction plans.

C-2. *ADAPTIVE JUDGMENT*$[X, t]$ is a fragment of a process that establishes or maintains the prioritized relationships (implemented with the use of links) among the following parts of the process:

- Phenomena perceived by an agent at the moment t, both within the physical world (i.e., hunks) and within her/his imagination (i.e., m-hunks).
- Associated to these phenomena m-hunks of knowledge that enable an agent to interpret, understand, and judge the processes occurring within her/his environment, particularly from the point of view of fulfilling the needs and beliefs pursued by this agent.
- Agent's decisions concerning the initiation of her/his interaction plans at the proper moments, including the following activities:
  - Agent's "conclusions" resulting from her/his judgment of the situation, formed as a result of the establishment of a relationship between the perception of the physical world and the perception of the agent's knowledge in the time period t.
  - Proposals for the expansion and/or modification of the agent's,

KNOWLEDGE[X,t].

– Proposals for the improvement of the processes related to the,

ADAPTIVE JUDGMENT[X,t].

C-3. *INTERACTIONS[X,t]*—all physical interactions at a given moment (or within a period) t of time related to the domain X of an agent (intuitively speaking, by this we mean the processes taking place in the physical "body" of an agent, along with the processes of interaction which take place in the surrounding environment and which may have an impact on an agent).

By comparing the above description of the Wisdom Equation with the relevant elements of Fig. 28.3, one may note that some relationships exist between the B-1, B-2, and B-3 conditions and the C-1, C-2, and C-3 conditions. The term of "prioritized links" constitutes a key element of these relationships.

Let us recall that link priorities are based on the perception of the environment through the agent's control with the use of links. It is based on the assessment of the level to which the basic needs of an agent were fulfilled. **The nature of the adaptive judgment may be, in simplified terms, viewed as an interaction-based adaptive construction of networks of links (carried out with the use of other links) between the phenomena perceived by an agent and the priority needs pursued by her/him together with adaptive mechanisms of reasoning about dynamic properties of networks of links.** The agent's control assigns subsequent links to the interactions that have been planned on the basis of the agent's needs, context, judgment, and agent's capacity to construct and implement interaction plans. At this point, it is quite important, especially for the process of judgment, to systematically adapt the procedure to the changing needs of an agent, her/his level of knowledge, and the environmental conditions. This adaptation should also encompass the reconfiguration and/or construction of new links, with a purpose of being used by new mechanisms of the adaptive judgment. Thus, even in this context, the links play a significant part, especially when it comes to the construction of links that indicate priority links, tasks, problems, and phenomena.

In other words, one of the main aim of adaptive judgment may be understood as a process of establishing and/or modifying a set of links among knowledge and/or interactions (cf. Fig. 29.1).

Continuing the above conclusion, if we assume that the agent's "wisdom" reflects her/his abilities in terms of effective problem-solving, aimed at fulfilling her/his needs, one may present the metaphorical illustration of the c-granule concept in a slightly different form, with the use of Fig. 29.2.

The figure illustrates the intuitive assumption according to which a c-granule may be treated as a c-granule-relative wisdom of an agent or the other way round—that the wisdom of an agent is an aggregation of wisdom contained within all her/his c-granules.

We should note that the layers in Fig. 28.3 are of a different height than those presented in Fig. 29.2. In this way, we are able to emphasize the fact that the **relationships between layers presented in** Fig. 29.2 have a "vague" character.

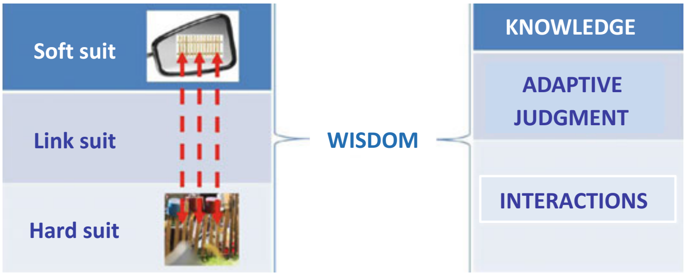


Fig. 29.1 Illustration of the conceptual relationships between the Wisdom Equation and the c-granules

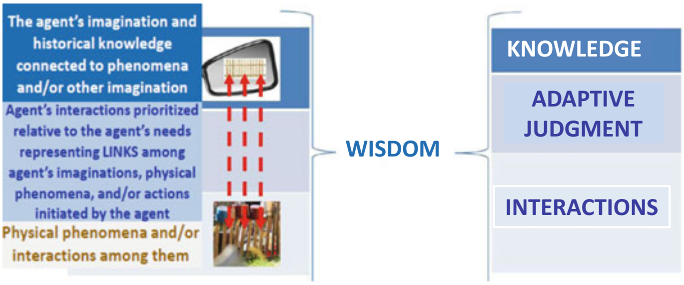


Fig. 29.2 Illustration of the conceptual relationships between the Wisdom Equation and the c-granules (cont.)

29.2 Wisdom Equation in the Context of CSE Projects

The metaphorical Wisdom Eq. 29.1 can be applied both to a single agent and/or to a society of agents, attempting to act as wisely as possible in any field of application referred to as X. For instance, X can denote activities related to the implementation of a complex project. Taking into account that X can be applied to any time interval, we can omit the time t variable in the equation. Thus, we can symbolically express the equation in a simpler form as follows:

$$WISDOM[X] = \\ INTERACTIONS[X] + ADAPTIVE\ JUDGMENT[X] + KNOWLEDGE[X]. \tag{29.2}$$

In accordance with the above-mentioned intuition, the equation states that the essence of the agent's (or of a society of agents) wisdom in the field of application known as X, lies in the ability that an agent (or a society of agents) can improve the judgment (including the analysis and evaluation) of phenomena perceived by an agent through her/his complex-glasses (cf. Sect. 27.1) in an adaptive way. These phenomena occur simultaneously in two worlds:

- the world of knowledge about X, which is contained in the mental world of an agent (or agents) and can be found on various storage media,
- the world of interactions carried out by interactive processes in the physical environment where an agent or a society of agents act.

In the metaphorical Wisdom Equation, all three components are highly important. However, from the perspective of effective functioning by an agent in an unknown environment, of particular importance is the ability to take a relevant interaction plan. In other words, speaking from the perspective of the agent especially crucial is $INTERACTIONS[X]$ and/or $KNOWLEDGE[X]$. The relationships can be treated as some kind of metaphorical "bridges" which identify and connect important interactions on one "side" of the river and important, useful knowledge on the other side. This "bridge" is:

$$ADAPTIVE\ JUDGMENT[X].$$

The "bridge" can be considered from the perspective of images, seen by an agent (via her/his complex-glasses) as an enhancement to the model for interactive computations on granules (intuitively corresponding to thoughts). In simplification, one can imagine that the adaptive judgment enhances the general model of interactive computations on granules by the following c-granules:

1. *The current c-granules of the agent's adaptive hierarchy of needs and values, which can intuitively be thought of as specifications (with appropriate c-granules—the most currently expected objectives to be reached by an agent as a result of the completion of her/his interaction plans).*
2. *An aggregation of current c-granule, equipped with complex-glasses consisting of a set of c-granules, associated with the "images" appearing in these complex glasses.* We assume that this aggregation should encompass (in the form of a metaphorical filter) the current c-granule of the agent's hierarchy of needs and values. The set of c-granules equipped with complex-glasses contains c-granules which put forward proposals, often contradictory to each other, for actions (or interaction plans) to be performed by an agent (as well as prohibitions on action/interaction plans, which an agent should not take).
3. *A selection of a set of the next actions (or interaction plans) to be performed by an agent on the basis of these suggested by the current c-granule associated with the complex-glasses and their consequences.* It should be noted that this task is difficult due to the fact that in practice, the current c-granule associated with complex-glasses often contains contradictory proposals regarding the action/interaction plans. The selection of these actions should take into account the suggestions of the current c-granules of the agent's hierarchy of needs and values. Actions can be performed by an agent in parallel. One of the most important actions is to direct the agent's complex-glasses on a phenomenon which, according to her/his judgment, are or may become important. Of course, if an agent decides it is worth and feasible to do so, then she/he may consult relevant fragments of knowledge with other agents. This may help her/him to find

an optimal solution of the problem with contradictory or uncertain suggestions concerning selecting the next action.

4. *A meta-judgment (judgment about judgment) of the currently implemented judgment from the perspective of the current c-granule of the agent's hierarchy of needs and values.* It includes proposals and actual corrective actions that can to improve the actual judgment.

The outline of the concept of $ADAPTIVEJUDGMENT[X]$, presented above, which is an enhancement to the interactive calculation model on granules, constitutes a setup, implementation, and adaptation of the relationships (links) in many types of agent's c-granules. These links are primarily between the mental world of an agent (including her/his needs, values and recommended actions) and the physical world of interactions, in which X is embedded (i.e., the objects and physical phenomena in the environment of X). In this approach, we assume that the agent's wisdom on X consists of, among others, her/his ability to select (to establish a link to) a relevant chunk of knowledge on X, which, in the agent's opinion, will allow for resolving of conflicts between contradictory c-granules.

These selected chunks of knowledge generally help an agent realize how uncertain her/his assumptions about the relationship between the observed phenomena and their judgment are. As a consequence, agents rely on such a judgment to decide which chunks of knowledge are to be used. In other words, the agents' judgment consists of making 'a priori' assumptions for "certain" types of reasoning about their established knowledge.

For example, while computing technical parameters of a building, experienced engineers will use Euclidean geometry instead of non-Euclidean Riemannian geometry and classical physics instead of relativistic quantum mechanics.

Thereby, experienced engineers use "intuition" which resulted in a judgment that Euclidean geometry and classical physics constitute—in their opinion—a "good enough" approximation of reality (as far as their calculations related to construction are concerned). By this simple example, it is clear that different chunks of knowledge can yield different, contradictory actions. An agent should select the chunk of knowledge that can support her/him in deciding which actions to undertake next. This illustrates the desired property of adaptive judgment—the ability of an agent to collect, at a right time, major arguments "for" and "against" undertaking a given interaction plan.

Having collected these arguments, adaptive judgment proceeds to their integration and, at a right time, makes the final decision as "for", "against" or "neutral" As we know, **very often, the lack of any decision is the worst decision to take**.

As has been noticed earlier, the metaphorical Wisdom Equation can be used in many different contexts, denoted as X. For instance, if we substitute "Data Analytics (DA)" for X, the equation yields a conclusion that wisdom, within this scope of $WISDOM[DA]$, is made up of three components:

1. $INTERACTIONS[DA]$—implemented in the world of physical representations (e.g., on magnetic media) and implementations—techniques for data stream processing,

2. $ADAPTIVE\ JUDGMENT[DA]$—this is the world of judgments made by domain experts (including the qualitative judgment of the quality of discovered knowledge, e.g., approximations of concepts), capable of interpreting phenomena on the basis of data and the evaluation of these data, in particular. Using adaptive judgment, one may also determine: Which assumptions and models are better? What's important and what's not in the knowledge discovered by data analysts? What kind of conflict resolution mechanism should be used in order to optimally fulfill the agents needs? What is next to be done?
3. $KNOWLEDGE[DA]$—this is the world of the mental phenomena of observed by a data analyst, representing the knowledge (including "good" practices) "approved" by the "adaptive judgment" and involving skills of data analysis.

Therefore, in this context, the metaphorical Wisdom Equation becomes:

$$WISDOM[DA] =$$

Data contained in the physical world of their representations and interactions

+

Adaptive Judgment as a result of the cooperation between domain experts and "computer systems" (computer implementation of data analysis techniques) leading to decision, what is important? why? and what to do next?

+

Current knowledge (including principles, interaction rules, reasoning schemas, and "good" practices) learned and/or discovered and approved by Adaptive Judgment by agent's control, domain experts, and by credible (by agent's control) external sources.

To paraphrase the above equation, we may say that even the best knowledge and skills in data analysis are not sufficient to provide valid and complex solutions to practical problems. In such situations, the access to real data, practical conflict resolution procedures, and real adaptive judgment of real domain experts is required.

Unfortunately, in practice, it turns out that the educational program on data analysis and/or real projects executed by analysts are often supported by "refined" data and superficial opinions expressed by the "currently available" peaceful experts (often of the dubious quality). This generally leads to the application of too trivial models and techniques which lead to incorrect conclusions. We may also use the Wisdom Equation to better understand how to model the computation supporting the processes of acquiring, processing, and sharing various "chunks" of wisdom by a given agent, ag. To this end, we may apply the following interpretation of the metaphorical equation:

$$\begin{aligned} &WISDOM[ag] = \\ &INTERACTIONS[ag]\ + ADAPTIVE\ JUDGMENT[ag]\ +\ KNOWLEDGE[ag], \end{aligned} \tag{29.3}$$

where:

1. *INTERACTIONS*[ag] is the world of physical phenomena (resulting from the interaction of physical objects) concerning an agent, ag, as well as physical phenomena occurring in the environment relevant to her/his functioning, including, the physical implementation of an agent, in particular.
2. $ADAPTIVE\ JUDGMENT[ag]$ contains mechanisms that establish important relationships (links) between phenomena occurring in the world of physical phenomena concerning an agent, ag, and the mental phenomena of ag. This is a world of judgments of an agent concerning her/his interpretation of the physical and mental phenomena. A key aspect of adaptive judgment is the potential ability to judge a situation in accordance with the agent's hierarchy of needs and values. This can also help specify what assumptions and models are valid (with regard to the hierarchy of needs). Adaptive judgment should justify and provide "optimal" decision about "what to do next?"
3. $KNOWLEDGE[ag]$ is the world of agent's mental phenomena (*KNOWLEDGE*) expressed in a language understood by this agent and, also, with some degree of trust approved by her/his adaptive judgment. This world of mental phenomena includes images and relationships between them (including principles, interaction rules, reasoning schemas, and "good" practices), expressed in the agent's mental language and stored on external physical media using a language understood by the agent. This language also includes concepts, inference, available knowledge, agent's hierarchy of needs, as well as sets/specifications of possible agent's actions in the world of physical and mental phenomena.

Chapter 30
Some C-granule "Links" to Socratic Dialogues, Tarski Truth, and Semantic Games

SOCRATES:.. but when they came to the letters,
*"This invention, O king," said Theuth, "***will make the Egyptians wiser and will improve their memories; for it is an elixir of memory and wisdom that I have discovered."**
But Thamus replied, Most ingenious Theuth, one man has the ability to beget arts, but the ability to judge of their usefulness or harmfulness to their users belongs to another; and now you, who are the father of letters, have been led by your affection to ascribe to them a power the opposite of that which they really possess.
For this invention will produce forgetfulness in the minds of those who learn to use it, because they will not practice their memory. Their trust in writing, produced by external characters which are no part of themselves, will discourage the use of their own memory within them.
You have invented **an elixir not of memory, but of reminding; and you offer your pupils the appearance of wisdom, not true wisdom, for they will read many things without instruction and will therefore seem to know many things, when they are for the most part ignorant and hard to get along with, since they are not wise, but only appear wise.**
Plato "Phaedrus (dialogue)" 274e–275b [399]

Despite the passage of time, the above thoughts of Socrates are still valid, especially when we are talking about the thoughtless and frequent use of the **letters of contemporary media**, including the "letters of internet." Hence, in the quote above, one should replace "letters" with the "letters of contemporary media" to arrive at the following paraphrase of the Socrates's thought:

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_30

> SOCRATES: ... **letters of contemporary media** invented an elixir not of memory, but of reminding; and "letters of internet's" offer to their pupils the appearance of wisdom, not true wisdom, for they will read many things without instruction and will therefore seem to know many things, when they are for the most part ignorant and hard to get along with, since they are not wise, but only appear wise.

The WisTech's approach towards "wisdom" presented in this book refers to the motto of this chapter, which is the above-quoted message of Socrates. In simplified terms, the motto may be interpreted in a way in which **wisdom is not only the "logos of letters," but, above all, interactive**, adaptive and wise "dialogos" which contains instructions that could be helpful in the process of solving some practical problems. **In this book, we use "logos" to refer to a single logic of an agent in contrast to "dialogos," which as a term denotes numerous logics of various agents who communicate with each other**.

Socrates's thoughts may suggest that while developing, civilizations deepen their knowledge and power through the use of a wise "dialogos," at the same time submitting that "dialogos" to the "logos of letters" (not the other way round). Along with the passage of time, maintaining the advantage of the growing level of the wise "dialogos" over the growing level of "logos of letters" requires more and more effort. Particularly, performing the wise "dialogos" is more and more complex as it requires a proper selection of the most important problems to be solved, along with a proper assessment of the chances for the success. It also requires systematic improvement of education of the new participants that become involved in the process. None of the currently known civilizations was able to face this challenge in a satisfactory manner.

After some time, a civilization that was able to face the challenge is gradually reducing the advantage of "dialogos" over the "logos of letters" and starts moving towards the phase in which "logos of letters" gains a position which is dominant over the interactive, adaptive, wise "dialogos." According to Socrates, this usually leads to the distortion of "wisdom." Even such great civilizations as the ancient Egypt could not face the challenge at some point—after some time, the advantage of the "logos of letter" over the dialogos component led to the fall of the ancient Egypt.

It is worth to recall the fact that Socrates did manifest his beliefs through his actions. Particularly, he did not leave any—authorized by him—works for the future generations. He stated that they may only constitute **a representation of the "logos of letters," that could be wrongfully interpreted, instead of that of the "dialogos."** Thus, the documentation of the "logos of letters" would make it impossible for Socrates to instruct his interlocutors within the scope of a proper understanding of his message after his death. In this way, Socrates was able to avoid a potential distortion of his own, direct messages. Fortunately for the future generations, his students, particularly Plato, transcribed numerous dialogues that were reportedly created by Socrates [399].

The conflict between "dialogos" and "logos of letters" present within the process of acquiring wisdom was continued by the students of Socrates. The conflict is still visible in many areas of the human activities. From the perspective of a technological development, especially AI applications, the nature of that dispute which has been present for a long time, is well summed up by the following quote from the introduction of the book by Johan van Benthem entitled "Argumentation in Artificial Intelligence," pp. vii in [408]:

> I see two main paradigms from Antiquity that come together in the modern study of argumentation: Platos' Dialogues as the paradigm of **intelligent interaction**, and Euclid's Elements as the model of rigour. Of course, some people also think that **formal mathematical proof** is itself the ultimate ideal of reasoning—but you may want to change your mind about reasoning's 'peak experiences' when you see **top mathematicians argue interactively at a seminar**.

It means that any interactive debate could be treated as a an intelligent reasoning in terms of the argument processing game.

In the light of the considerations presented above, referring to the advantage of "dialogos" over the "logos of letters," a significant role is played by the search for a technology that would be capable of supporting the wise dialogue between a man and a system, which enables her/him to access the resources represented in a form of "letters," including the net dialogues.

The search for this type of technology refers to the Turing Test[1] carried out in the 1950s. Despite the increased efforts undertaken during the last decades and the progress achieved within this scope, until today we have had no knowledge of a technology which would be capable of performing a substantial text analysis and of carrying out a "wise dialogue" on that basis, in the line of requirements of the Turing Test. Intensive research within that scope is being carried out by numerous reputable research centers all around the world. Particularly in the recent years, numerous Turing's awards[2] were related to the search for the computational models that could represent the natural language processing capabilities, along with the capabilities of processing the knowledge represented using the natural language. Let us recall that in the Excavio projet (cf. Chap. 19), the author describes modest attempts within that scope of engineering of simple "intelligent" man-machine systems. This activity led to the research on WisTech, carried out together with professor Andrzej Skowron [244, 245, 247]. **In the case of WisTech, links of c-granule's link_suit play a key role within the interactive, adaptive dialogos**.

The research undertaken during the first half of the 20th century, concerning the understanding of the concept of "truth" and reasoning related to "true" sentences, was based, in an essence, on the formalization of the way the term of "truth" was understood, in a form proposed by Alfred Tarski [163, 497, 499].

In a matter of fact, the concept of a c-granule can be considered as an extension of the concept of satisfiability relation by Tarski in formalized languages of deductive science [497] and [26, 39, 48, 77, 163, 229, 296, 410, 474, 499].

Analogous to a c-granule, which can be treated as a function transforming certain c-granules into other c-granules, from a purely formal point of view, a satisfiability relation may be treated as the characteristic function of a satisfiability relation.

---

[1] http://en.wikipedia.org/wiki/Turing_test.

[2] http://amturing.acm.org/byyear.cfm.

Such a characteristic function has a model and formula as arguments and returns a truth value (true or false) in the case of classical bivalent logic. In the case of multi-valued logics, the values of the function can also be other logical values [245, 262, 409, 410, 513, 546]. Thus, the backbone layers, with such a metaphorically simplified understanding of c-granules, can be intuitively imagined as follows:

1. Hard_suit—these are models of a language,
2. Soft_suit—these are expressions of a language
3. Link_suit is a satisfiability relation between expressions and models of a language (more precisely, a characteristic function of this relation).

The static approach to the satisfiability relation proposed by Tarski has played a dominant role in the development of logic and foundations of mathematics in the 20th century.

At the same time, just like any other philosophical concept, this approach has also become the subject of criticism by many philosophers and logicians. An example of such a criticism are proposals by Wittgenstein [539] to the treatment of language and its meaning based on potentially changing and not always unambiguous rules used in “language games.”

These works by Wittgenstein were deemed by many logicians (especially within the Vienna Circle and Cambridge) as the attacks to the dominant, in many scientific centers, analytical philosophy of the 20th century. Moreover, Wittgenstein himself confirmed that this was his intention. Due to the dominant role played by the paradigm of analytic philosophy in the 20th century, the criticism initiated by Wittgenstein did not meet the approval criterion at once.

Nonetheless, many logicians of the 20th century used much more moderate (in comparison to Wittgenstein) ways to “animate” the highly “static” approach by Tarski to the satisfiability relation. Within this stream of research, the evolution of approaches to semantics under the name of “semantics games” [3] [46, 324, 398], have played an important role. It has become increasingly popular among professionals working on intelligent dialogue systems.

The “semantic games” approach to formal semantics treats the satisfiability relation on the basis of game theory ontology (concepts), particularly the concepts relevant to the existence of a winning strategy for a player. In a sense, the concept of **semantics games is somewhat resembling Socratic dialogues** discussed at the beginning of this chapter. Moreover, in some sense, Aristotle’s writings about syllogism are closely intertwined with his study of the aims and rules of debating. Aristotle’s viewpoint survived in the common medieval name of logic, called dialectics.

In the mid-20th century, Charles Hamblin [186, 187] established interesting links between dialogues and the rules of sound reasoning. Following this, Paul Lorenzen [492] connected dialogue with constructive foundations of logic. Some modifications of Lorenzen’s games along these lines could serve as an approach to informal logic

[3] http://plato.stanford.edu/entries/logicgames/.

and especially to the research that aims to systematize possible structures of sound informal argument and AI applications [408].

From the point of view of AI applications, models of argumentation as a dialogue process and diagrammatic views of argument structure seem especially interesting [43, 529]. The phrase "models of argumentation as a dialogue process and diagrammatic views of argument structure" is not limited to the potential AI applications of interactive processes between the human beings only or to the processes between artificial intelligent agents only. In practice, this phrase has a crucial meaning in regard to the interaction between:

$$(society\ of\ humans)\ \ and\ \ (society\ of\ agents). \tag{30.1}$$

On the other hand, these systems are supposed to serve mankind and therefore, human beings should be able to communicate with computers representing wisdoms in a lucid manner. Interactions should give the greatest synergy effects possible as a result of this type of cooperation.

The metaphor concerning the **path to knowledge through "Socratic dialogues" presented above can be treated as one of the main pillars of AI technology when it comes to learning the ways to approximate and recognize important vague complex concepts** (such as risk or chance of success) in a satisfactory way. This ability should allow an agent to decide whether a given object falls under a complex concept or not. In practical situations, the process of reaching such a decision is itself highly complex and often, does not yield a definitive outcome.

The quality of this decision depends on the balance between arguments "for" and arguments "against" the membership of a test object to a complex concept (e.g., if a given bank is in a high risk of bankruptcy). Learning the mechanisms that allow for the collection of important arguments "for" and "against" as well as the discussion and investigation aimed at resolving this type of a conflict can be considered as one of the central directions of AI technology. As an example of an interdisciplinary area, where the issues of resolving conflicts using judgment are of great importance, one can also mention here the computational social choice area [67]. Again one should have in mind the motto of this book by Pearl and Zadeh.

Considering that the purpose of learning this type of mechanisms is to elicit the satisfiability relation for highly complex concepts in a dynamic environment, one should expect that the accurate elicitation of a precise and unambiguous satisfiability relation is not possible within a reasonable period of time.

Therefore, we have to limit ourselves to eliciting approximated forms of this satisfiability relation. At the same time, we expect that this approximation will be satisfactory enough for typical applications. In other words, using the metaphor of learning how to safely drive a car to a given destination using generally abiding rules, we do not need to learn to move around with excessive precision. Moreover, given the limited knowledge about the environment, we have to accept the fact that on the way, there may occur situations which are absolutely unpredictable to a man. Therefore, arranging too many travel details beforehand is a waste of time and other resources. Moreover, when it comes to fulfilling the trip requirements, it is much more important to systematically monitor and implement the interactive processes with the environment. In this particular example, such processes include the following, concurrently implemented tasks:

- monitoring the values of attributes that describe important aspects of a trip by car;
- reviewing to what extent the criteria that we expect our trip by car to fulfill are affected;
- identifying, analyzing, evaluating and judging arguments “for” and “against” undertaking corrective actions or completing additional experiments which enhance our understanding of the situation on the road;
- performing corrective actions.

To simplify, it is not important for us to know too precisely the exact conditions that our trip by car should meet (or more generally: to know too precisely the exact conditions of implementing complex projects). It is much more important to prepare and—if possible—elicit and discover adaptive rules and habits of a skilful and interactive game with a changing and unfamiliar environment so as to optimally satisfy—often changing in time—travel conditions (or more generally—the ever-changing conditions of implementing complex projects).

In this example, the goal of the “game” with the environment is to fulfill, in the best way possible, major travel requirements through a continuous interactive analysis. Of course, this example can be extended to an unbounded number of other fields of activity. The metaphor of driving a car to a given destination is an important observation with regard to the principles of complex project engineering. This metaphorical example indicates a potentially significant role of approximate learning of the satisfiability relation to complex and changing concepts. In the 20th century, the dominant approach in this field was based on formal logical deduction, which generally requires extremely high computational complexity (in practical applications). Often, in these approaches there is a closed world, which does not work well with the practice involving the learning of complex vague concepts.

Problems arising from the high complexity and lack of tools for effective deduction of cause-and-effect relationships still constitute a crucial barrier, limiting the development of practical applications of automatic AI technology in areas such as risk and security management.

Therefore, many researchers are wondering how to overcome such difficulties. The method of IGrC (more specifically, c-granular computing) presented in this book and implemented within the framework of WisTech aims to indicate some promising directions in this field.

Part VII
Framework Postulates for Ontology of WisTech Models

Chapter 31
Complexity of Designing an Ontology for Practically Useful IGrC Models

31.1 IGrC and CSE Development Synergy Hypothesis

In Chap. 22, we have noticed that CSE projects can be treated as interactive thinking processes, implemented by the society of agents.

On the other hand, we use IGrC for thinking processes modelling. Thus, we can point out similarities between the key functioning mechanisms of the following processes:

I. IGrC—with particular emphasis on techniques of reasoning about properties of computations, which can be effectively used in practical IGrC applications, especially in the AI technology application.
II. Effective completion of CSE projects.

Both types of processes, I and II, are considered in the book as the implementations of their respective computing models. Analogies between type I and type II computing models show there is a deep relationship between IGrC and CSE.

Let us recall that at the turn of the century, these two fields, CSE and IGrC (in our approach, IGrC also includes AI), have been studied and developed mostly independently.

At the same time, the current results of the theoretical research in these fields do not enjoy a favorable opinion among practitioners. Many practitioners believe that the results are far from being applicable in complex practical tasks related to CSE and/or AI. They share the view that we still do not understand the mechanisms for the implementation of AI in complex applications and the mechanisms for the delivery of CSE projects well enough.

In this context, the statement that "CSE projects can be considered as a special case of IGrC" does not expand substantive and practical knowledge of IGrC and/or CSE. In contrast, this statement leads to an important conclusion that studying the properties of CSE projects can deepen our understanding of IGrC, as well as enhance the effectiveness of its application and, vice versa, examining IGrC can also deepen our

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_31

understanding of CSE projects, and boost the effectiveness of their implementation.

This means that a deeper analysis and understanding of the analogies between the key mechanisms of the above-mentioned types of processes (I and II) should bring significant benefits for both of these areas and result in an accelerated development of IGrC and CSE. It should also contribute to increasing the efficiency of applications. Of course, the positive effects stemming from a possible synergy between the development of IGrC and CSE may also find applications in many other fields (e.g., chemistry, biology, medicine).

In conclusion, the above mentioned synergy effects can be understood as follows:

IGrC and CSE synergy hypothesis

Conclusions drawn from the development of IGrC techniques should lead to an acceleration in the development of CSE (with particular emphasis on improving the effectiveness and efficiency of CSE project products and services delivery) and vice versa, conclusions drawn from the development of CSE should help speed up the development of IGrC (with particular emphasis on widening the scope and improving the efficiency of application of IGrC technology).

The *IGrC and CSE synergy hypothesis* is expressed at a fairly high level of abstraction and therefore, most probably, it will not encompass any deep semantic content for people without a practical experience in IGrC and CSE. Hence, given its great importance in this book, we shall continue to concisely explain the meaning of this hypothesis from the perspective of this book.

To explain the intuition of the aforesaid hypothesis, we will start with the following remark concerning TPGP (cf. Chap. 11 and Sect. 1.2.2). Let us note that a consequence of the hypothesis is the expectation that the development of IGrC should contribute to improving the efficiency of CSE project delivery, in particular, to minimizing the risk of the adverse impact of TPGP. In order to identify possible mechanisms for the implementation of this hypothesis, in the next part of this book, we shall illustrate them with the use of IGrC as a natural computing model [420]. The example presented by us is an extension of the computing model that was succinctly described in Sect. 13.2.3 of the book, [42], and a lecture by E. Ben-Jacob entitled: "Learning from Bacteria about Social Networks".[1] This model can be used to build an optimizer for solving the classical multi-criteria optimization problem. The working mechanism of this optimizer is based on a simulation of adaptive behavior of a society of bacteria.

Observing the behavior of a bacterial colony, we can see the mechanisms of collective problem solving [42]. Once we achieve a deeper understanding of these mechanisms and express them in a proper way, we can try to apply them to cope with the CSE problems in a more effective way. Simple examples of behaviors displayed by the colonies of bacteria, which can be compared with that of the participants of CSE projects, have been described in Sect. 13.2.3. More interesting examples can be found in literature [42]. Of particular interest are such examples as given below.

[1] https://www.youtube.com/watch?v=yJpi8SnFXHs.

a. Mechanisms to improve CSE project management through decentralization (e.g., using a network of relevant moderators and informants). In a typical colony of bacteria, there does not exist a single bacterium performing the classical role of a project manager!
b. Mechanisms for improving the adaptation of communication in CSE (e.g., through learning by moderators and improving channels and forms of communication throughout the colony).

These mechanisms are illustrated in more details in the cited lecture and in [42], in the context of the following observations:

a'. The high effectiveness of decentralized behavior control displayed by bacterial colonies initiating an optimized search for food, which dynamically appears in a changing environment and in the context of the Buridan's Ass problem.[2]
b'. The high effectiveness of communication between bacteria, leading to effective teamwork, especially in response to the most imminent threats to the entire colony (e.g., the mechanisms of resistance of certain strains of bacteria to antibiotics).

On the other hand, the conclusions drawn from a practical experience in the field of CSE should lead to the construction of better models for IGrC. Then, these models should pave the way for new—much more effective than those that are currently available—AI technologies. To illustrate the significance of this process, let us consider another example.

One of the key barriers that currently hinders the development of AI applications is the lack of understanding (not to mention the lack of an effective automation) of the mechanisms for **exploring and learning relevant complex concepts** effectively supporting constructions of classifiers, optimizers, and interaction plans in an uncertain environment.

To simplify, the better concepts we use, the more effectively we can construct: classifiers, optimizers, and builders for interaction plans. By a better concept, we mean primarily a concept that allows the completion of a more efficient computing scheme. In general, a better concept is typically more suited to the specifics of its application domain. Hence, while looking for "better concepts," we ought to take into account domain knowledge of an expert to the greatest extent possible.

It is worth emphasizing that the issue of "discovering and learning relevant complex concepts" applies to virtually all applications of modern AI technologies. This is directly due to the fact that typical AI applications constitute the aggregation of techniques for classification, optimization, and interaction planning.

At the same time, the implementation of CSE projects provides us with a great deal of practical knowledge in this field. Mechanisms for discovering and learning—by project teams—all the necessary concepts (in order to properly complete project tasks) constitute, in a sense, the substantive nature of a CSE project. The techniques for discovering and learning concepts that support the decision-making processes

[2] https://en.wikipedia.org/wiki/Buridan%27s_ass.

concerning the classification and/or optimization and/or action planning are of particular importance. These techniques are indispensable when dealing with unexpectedly occurring (and often unpredictable in their content and scope) problems in the implementation of complex projects. In the case of well-planned, well-managed and well-organized CSE projects, it was good (or bad) effectiveness in exploring and learning concepts, supporting the above-mentioned decision-making processes that decides the success (or failure) of a project.

Experiences related to CSE projects and developing new AI technologies (including the experience described in the case studies in Part IV of the book), it appears that many mechanisms (discovering and learning relevant complex concepts collectively) used in complex projects can be automatized and applied in the development of AI technologies and vice versa. However, unfortunately, this is not a simple task.

Arbitrary analogies are hard to spot since they are apparently related to different phenomena, with approximate and often flawed descriptions, expressed in different languages (including languages of CSE experts and AI experts). A simple translation of concepts between these languages does not solve the problem as these languages do not operate on the same concepts.

Therefore, while translating between these languages, one would need to employ approximations of concepts expressed in these languages and/or to construct new common concepts. The problem is especially visible in the case of modern AI R&D projects. They are unique in featuring "innovative" and hermetic internal languages. The result is that often, for such projects to capture the essence of their specifics and to transfer the experience from the delivery of CSE projects to the development of IGrC, individual adaptation (to the specifics of IGrC) of this experience is required.

Such a complex work—performed separately for each project—often becomes inefficient cost-wise. In this context, having a language that could express and classify experience obtained from CSE, to make it easier to adapt to different AI projects, would be vastly convenient.

In other words, such a language should help to better understand and express analogies between key operating mechanisms of the processes in IGrC and CSE (especially, we mean the analogies between mechanisms in processes described above in points I and II). This language should facilitate the transfer of knowledge, gathered from the experience in CSE, in order to enhance the automation of discovering and learning relevant complex concepts in new AI applications.

To sum up—the above considerations concerning the CSE and IGrC synergy hypothesis show that the absence of effective means for a transfer of knowledge, and experience between the IGrC and CSE may constitute a serious barrier in the development of AI, IGrC, and CSE, as well.

31.2 Communication in CSE Projects from the Perspective of IGrC and CSE Synergy Hypothesis

Having in mind the Pearl's motto of this book [392], one of the critical success factors of development of contemporary AI is the development of a man-machine communication by a "science-friendly language for articulating causal knowledge." In particular, it is also critical success factor for the CSE development. For any CSE project, we need a "science-friendly language" to specify models which illustrate important properties of a system that is being developed and techniques for the project's implementation. In other words, these properties include both the properties of a target system and those related to the tasks performed during the development of a system.

Of course, every specification requires a language of communication. In the case of complex projects, it often happens that there are several stakeholder groups, each speaking a different language. These different stakeholder groups can even share a common language, but it may understood differently by each group (e.g., financiers, developers, or mathematicians).

In such a situation, *translators/moderators* may play an important role in facilitating the process of communication between stakeholder groups. Translators/moderators attempt to translate and explain contents across all parties that are involved in a project, especially when there is any doubt regarding the proper understanding of a given content by the sender and the receiver. They also attempt to moderate the process of reaching a mutual understanding. It is worth to note that the role of translators/moderators in CSE projects is not only limited to translation between different languages.

For example, they can also be agents who support distributed project management in various domains. Typical areas of project management where translators/moderators may be employed are the management of: quality, safety, labor standards, culture, change control procedures. They can also be engaged in supporting the identification and resolution of design problems (including local conflicts).

Note that the role of translators/moderators is different from that of the local managers. The main difference lies in the fact that translators/moderators rarely make decisions. Instead they work to reach a consensus within the standards and requirements of a project or suggest changes where appropriate. Moreover, as opposed to managers, they do not control the design resources allocated to tasks other than the support for translation and moderation processes.

It should also be noted that nowadays, CSE projects may use a great variety of supporting tools, which we shall refer to as *automated agents* (or robots). Automated software tools that support the deployment of a project constitute examples of such agents. Modern computer tools employ formal languages, different from the natural language used by human beings.

Hence, in the case of such projects, there arises an additional complication as it is necessary to take into account relevant interfaces and/or specialized translators/moderators in order to enable, support, and verify the communication between human beings and machines, and across large groups of human beings and automated agents. Constructing appropriate languages for such advanced interfaces between machines and human beings is a highly non-trivial task [5, 66, 304, 319].

According to some authors (e.g., [392]), a language supporting communication and collective interactive knowledge processing by human beings and machines should cover well-defined areas of simple fragments of "science friendly" natural languages, which can be processed by an appropriately extended mathematical apparatus.

It is worth emphasizing that translators/moderators and interfaces—simple as they might appear—have to perform very difficult tasks. Moreover, in practice, it often happens that their tasks cannot be performed in a precise manner.

This somewhat surprising (at least for CSE beginners) conclusion stems from the fact that in different languages, there may be concepts with no mutual semantic counterparts. However, this phenomenon is particularly evident in the case of specialized jargons used by professionals. Additionally, the situation may get even worse, e.g., when two human beings use the same words, but understand them in a different way.

In such a situation, a translator/moderator (or a technologically advanced interface) should:

a. Quickly identify misunderstandings.
b. Evaluate possible negative consequences of these misunderstandings for a project.
c. Assign and update priorities for explaining actions on the basis of the evaluation.
d. Take appropriate interaction plans in order to explain misunderstandings (according to their priorities).
e. Verify the effects of explaining actions and undertake more actions if needed.

Should there exist any significant (for a project) differences in mutual understanding of a given content (between the sender and the receiver), a translator/moderator should take the role of explaining actions (cf. point d. above). In practice, this task requires extensive knowledge and skills of a translator/moderator (or the interface).

A translator/moderator (and/or an automated interface) should assist in reaching an optimal and acceptable approximation of concepts (in terms of concepts understandable to both the sender and the receiver of a given message). Such approximation is often needed when the language of the recipient is too poor to interpret the message (e.g., a typical European may find it difficult to select equivalents to different terms used by Eskimos to describe various shades of white). This is where a translator/moderator (and/or automated interface) should commence a dialogue with the recipient. During this dialogue, she/he should support the recipient by explaining given concepts or by providing their satisfactory approximation. Then—through a dialogue with the sender—a translator/moderator (and/or an automated interface) should ensure that her/his approximation of concepts is satisfactory to the sender's message. She/he may also be needed to negotiate with the sender some other acceptable approximations for the concepts.

Despite the difficulties outlined above, the tasks of translators/moderators (and/or automated interfaces) aimed at supporting the communication should be carried out with a satisfactory degree of accuracy (with regard to the project's needs). Without their proper implementation, it is impossible for a project to achieve success.

Unfortunately, the role of translators/moderators in CSE projects is still underestimated (especially when it comes to financial incentives) and consequently, their absence (or low quality of their performance) leads to a much larger financial loss, caused by the negative consequences of misunderstandings between project participants. The vast majority of failures in CSE projects have their source in the absence of communication (especially that between prospective users and developers). Hence, while constructing a CSE project team, one must keep in mind that someone has to assume the role of translators/moderators (perhaps after additional training).

The absence of translators/moderators translates into a lack of good communication in CSE projects. On the other hand, we must remember that translators/moderators are merely one of many other techniques supporting communication mechanisms. Therefore, even the best translator/moderator team will not warrant an effective—in terms of content and form—communication in a project.

On the basis of practical conclusions drawn from the projects outlined in Part IV of this book, it follows that communication in CSE projects should meet the following *basic criteria for effective communications*:

A. It should be based on an adaptive ontological apparatus and communication language that effectively support as quickly and cheaply as it is possible and as it is necessary:

    1. Identification, determination, aggregation, decomposition, prioritization, and change of design requirements.
    2. Identification, determination, aggregation, decomposition and prioritization of design problems.
    3. Determination of assessment criteria and assessment of solutions to design problems.
    4. Discovery and determination of the relationship between solutions to design problems.
    5. Discovery, determination, aggregation, decomposition and prioritization of different variants of the solutions to design problems.
    6. Use, construction, and development of design solution templates—in both current and future projects, and also including mechanisms for reusing the most important components of solutions to design problems.
    7. Determination of mechanisms for planning, management, organization and standardization of a project, tailored to its specifics.
    8. Determination of conditions necessary for using methods and tools applied in a project.

B. The content and form of communication should be, to a satisfactory degree (with respect to the implementation of a project and cultural traits of the project's stakeholders), clearly interpretable by stakeholders and automated agents. At the same time, the interpretation of the content and form must be culturally acceptable to stakeholders. In the course of the project's completion, it is crucial to maintain unambiguous interpretation (to a satisfactory degree of accuracy) of the fundamental aspects of the project such as:

1. Requirements concerning the final products of a project.
2. Requirements concerning the techniques for the completion of all project's tasks.

C. Communication should support, as effectively as possible and to the level of project detail which is satisfactory for individual stakeholders, the determination and transmission (among appropriate human beings and/or automated agents) of important requirements. This concerns both the target system that is being developed and all the processes related to its development and implementation. These requirements are articulated by different stakeholder groups and analyzed in terms of their relevance and feasibility. Then, in justified cases, they are approved by the project's management.
D. Communication allows for a fast adaptation of a project (including language and forms of communication) to the changing needs arising from both the specifics of a given project and the dynamics of changes to the environment in which a project is being implemented.
E. Communication should support fast conflict identification and resolution based on the current project needs.

The quality in meeting the basic criteria for an effective communication in a project (conditions A–E) significantly depends on the substantive core of communication, which constitute ontologies of languages used by the project participants.

Let us **recall that by ontology (in particular in the above point A), we mean a set of concepts and relationships between these concepts (which also constitute concepts).** For example, if some participants does not know the concept of a wheel (and the like) and the relationship between this concept and those representing the use of a wheel, **they can not discuss the possibility of using a wheel in the transport of heavy objects**.

The comments on communication presented above also apply to research projects related to the IGrC and CSE synergy hypothesis.

Deepening the analogy between the key mechanisms in computing models of processes (type I and II—Sect. 31.1) would be much easier if we had appropriate communication techniques between experts who investigate various aspects of the IGrC and CSE synergy hypothesis. Especially for this purpose, a sufficiently rich *specification language for computing models* is needed. This language should include concepts which, apart from unnecessary details, generalize key mechanisms occurring in mutually remote areas of application.

The language used by us in the previous parts of the book, which we will refer to as the *introductory language* does not meet this criterion. This is a consequence of its use. Therefore, the introductory language serves primarily to explain (mainly to specialists in fields other than IGrC and CSE) the basic concepts and results of the book, as well as the significance of key facts related to IGrC and CSE within the scope of the book's research.

Therefore, this language is much simpler than that supporting the specification of computing models. Introductory language uses intuitive and possibly well-known concepts which are generally imprecise. It should also be noted that in the case of introductory language, the fact that it comprises clear and commonly used concepts is indeed an advantage.

On the other hand, it lacks specialized concepts describing common mechanisms and properties of computing (for I and II in Sect. 31.1). This, in turn, is a disadvantage from the point of view of a specification language used in computing models as these concepts should be included in the *specifications of computing models*.

Moreover, a specification language of computing models should allow for the expression of useful specifications of computing models, which present the key mechanisms of the processes described earlier in the book (type I and II). By "useful specifications of computing models"—within the formation and implementation of a specified computing model—we mean a specification that meets such criteria as:

1. <u>Communicativity.</u> A specification language of computing models and specifications, to a great extent, support and facilitate the fulfillment of basic criteria for the effectiveness of communication in a project (i.e., points A–D above).
   Any team assuming a given role in a project ought to understand an adequate fragment of specifications, needed to ensure that the team members will fulfill their role in a project. In addition, a common holistic outline of the project's basic objectives should be clear to all project participants.
2. <u>Transferability of practical knowledge between IGrC and CSE.</u> Specification allows for the determination of the properties of models, which facilitate research and the flow of practical knowledge (including templates of design solutions) between IGrC and CSE.
   This knowledge primarily concern the conclusions of the research, involving the observation and analysis of mechanisms in examining, designing, building, experimenting, completing and/or maintaining systems that implement these specified models.
   When it comes to the transfer of practical knowledge, the priority is given to the ability to transfer the knowledge about the mechanisms of type I and II processes (Sect. 31.1), with a particular emphasis on mechanisms that can boost the effectiveness of cooperation between teams of agents (in IGrC) with regard to the fulfillment of their needs (e.g., agents, people, societies).
3. <u>Prospective implementablity.</u> Specification provides hints to seasoned engineers employed in the implementation of a specified computing model. By hints, we mean mainly architectural solution proposals, especially concerning the IGrC ontology concepts.

The design and development of useful specification of computing models is a highly difficult task. One of the main difficulties in completing this task is the selection of an appropriate ontological apparatus which constitutes key concepts of the specification language of the computing model. In the following section, we shall illustrate the level of difficulty towards approaching such tasks.

31.3 Level of Difficulty with Basic Ontological Task for IGrC

Following the above considerations, a systematic study on the IGrC and CSE synergy hypothesis should begin with finding an appropriate ontological apparatus. Such an ontology should facilitate the design and development of useful computing model specifications to expand our knowledge about the hypothesis.

Computing models that have been specified and implemented should be able to advance our knowledge on the key mechanisms of type I and type II processes (Sect. 31.1). In this context, it is important to start from the implementation of the following **Basic Ontological Task for the IGrC**, abbreviated as **BOT4IGC**:

The development of ontology and related rules, facilitating the design and development of the most useful specifications of computing models and their consequent implementations, in order to promote research and application of the key type I and type II processes (Sect. 31.1), *with particular emphasis on:*

I'. *Efficiency management mechanisms in satisfying the needs of an agent, using the agent's control of IGrC processes in a dynamically changing and partially known environment. By an agent (or a meta-agent), we also mean a team of cooperating and/or competing agents.*

II'. *Efficiency management mechanisms in successful CSE project completion.*

Constructing ontologies and rules for their development, which satisfactorily solve BOT4IGC, is in itself a very complex enterprise. In a sense, it is directly linked to the problems rated as the most important challenges of the humanity in the 21st century. For example, task I' is directly linked to the problem with understanding the concept of intelligence and computation modeling, which supports the simulation of intelligent human behavior.

In addition, the difficulty level of BOT4IGC from the perspective of task II' is related with the considerations included in the chapter on the key reasons of TPGP (Sect. 12.1), which de facto states that in the case of CSE projects, establishing an optimal granulation for such a project (including, among others, the construction of an appropriate ontology for the expression of global models) is one of the fundamental components of the fundamental precause of TPGP (cf. Sect. 1.2.2). Therefore, it can be treated as a different formulation of the BOT4IGC problem.

To better understand the important phenomena, concepts, and relationships between TPGP and BOT4IGC, we need to pay attention to the nature of the projects, which are particularly strongly exposed to the risk of failure. It is commonly believed that such projects include all complex projects that aim to design and produce an innovative new technology.

In such a situation, it is very easy to make an error and omit some important concepts that decide for the success or failure of such a project. Ontology learning and/or discovery is one of the biggest challenges for any innovative CSE project. Metaphorically speaking, practically it is impossible to apply the concept of a metaphorical "wheel" until one do not know the meaning of concept (in particular how to use it). Since projects aimed at developing new technologies are subject to such a high risk of failure, they are particularly interesting in the context of a study on fundamental TPGP precause (i.e., FP3C, cf. Sect. 1.2.2). Consequently, they are also very interesting to analyze when it comes to the ontology for modeling the control over interactive computing. We should begin our discussion on this subject from the following observation:

When planning the design and implementation of any complex new technology, at the very beginning, particular attention should be paid to the appropriate mutual iterative synchronization of classical PDCA cycles which play significant role in the construction, reconciliation, and improvement of the project's ontology (together with stakeholders relevant to a given project). It is designed to provide project participants with a conceptual apparatus to boost the efficiency of their work in terms of such aspects as:

* designing models for the new technology,
* working on the design and implementation of the new technology,
* conducting parallel analysis and assessment of compliance with requirements that a new technology is expected to satisfy,
* introducing adjustments to requirements imposed on new technologies and consequent changes in production.

It should be emphasized that for any new complex technology, very often it turns out that the pre-agreed successive versions of an ontology (and the versions of technology models based on this ontology) require a systematic adjustment and constant development. Hence, in this case, principles for the development and adaptation of an ontology become particularly important. These mutually synchronized iterative tasks (implemented in the corresponding PDCA cycles) apply to all types of CSE projects. However, in the case of projects involving new technologies, an incompetent management of these iterative tasks in practice will cause the failure of a project. The above-mentioned synchronization of the iterative tasks includes the following areas of activity:

- Analysis and revision of the basic principles that govern the design and development of ontologies, facilitating the iterative discovery of "optimal" core concepts used for specifying properties of technologies and techniques to produce new technologies. These principles should be systematically verified and updated in terms of the ability of ontologies to a more useful (for the respective participants project) expression of core architectural decisions concerning such aspects of technology production as:

  – analysis and improvement of requirements to be satisfied by the technology production process and by the new technology itself,
  – effectiveness of design, construction and technology use,
  – quality management of the technology production process and quality management of the target technology.

- Sample iterative discovery and approximation of an ontology for important (possibly small) parts of domain of applications. Then a graphical illustration of these "sample attempts," including mock-ups to illustrate and verify the proper way of understanding (by respective project stakeholders) of specifically important concepts and phenomena associated with this technology.
- Interactive construction of increasingly better prototypes of the new technology in order to verify and adjust the current basic principles governing the design and development of an ontology, as well as to verify the feasibility of key requirements expressed in this ontology with regard to the technology production process and to the target version of the technology.
- Possibly optimal (in the sense of the above-mentioned criteria, A–E, cf. Sect. 31.2) development of language and communication mechanisms, compatible with the ontology that is being developed.

Experienced practitioners in software engineering are well aware of the fact that for complex IT projects aimed at creating new technologies, implementing the very initial iterations of the aforesaid PDCA cycles for the identification and reconciliation of an optimal ontology expressing and communicating the requirements to be met by the target system (based on an technology unknown at the beginning of the project) is in itself a highly non-trivial task. Moreover, those seasoned practitioners are also well aware of the fact that constructing and obtaining an optimal ontology for the specification of requirements constitutes a solution to a very small portion of the BOT4IGC problem in relation to complex software systems engineering.

The arguments discussed above indicate a very high level of difficulty in BOT4IGC and are presented primarily from the perspective of TPGP and CSE issues. Serious research in this area started recently.

It should be noted, however, that the issue of ontology discovery, selection, construction, reconciliation, and communication aimed at improving the efficiency of human activities is very old. It directly refers to such an important idea as the "wisdom" in human actions.

Of course, the concept of "wisdom" has been known for thousands of years, before the emergence of the contemporary research on TPGP, CSE, or AI.

Particularly Aristotle devoted much of his attention to this concept. Some authors even believe that his work can be considered a pioneering study on the mechanism of practical wisdom. For example, in [504], we have already presented a very brief summary of Aristotle's views in this regard (cf. Quotation 28.1.4).

Since Aristotle, research has been conducted towards exploring a conceptual apparatus that supports the expression, explanation, and analysis of the phenomena associated with thought-processing processes and with reasoning related to this subject. Undoubtedly, significant progress in this field was made by:

- L. Zadeh, who introduced the paradigms of fuzzy sets, linguistic variables, and CWW [546, 548, 549, 551],
- Z. Pawlak, who invented rough sets and introduced tools based on rough sets for the approximation of complex (vague) concepts and reasoning about these concepts [384, 386, 389–391],

- J. Pearl, who draw our attention to the limitations of statistical inference and the need to build computer interfaces, including well-defined areas of simple fragments of natural language [392], which are processed by a sufficiently developed mathematical machinery. (cf. the Pearl motto of the book).

Obviously, this is in no way an exhaustive list of all important achievements in the field of research aimed at the exploration of a conceptual apparatus that supports the expression, explanation, and analysis of the phenomena associated with "wise processing of thought."

Let us recall that in this research, an important role is played by the broadly understood concept of judgment, which is nowhere near the typical tools of reasoning, such as deduction or induction. Our understanding of the concept of judgment refers, in particular, to the intuition presented in [504]:

> Practical judgment is not algebraic calculation. Prior to any deductive or inductive reckoning, the judge is involved in selecting objects and relationships for attention and assessing their interactions. Identifying things of importance from a potentially endless pool of candidates, assessing their relative significance, and evaluating their relationships is well beyond the jurisdiction of reason.

It is worth noting that the difficulties outlined above, associated with the construction and development of an ontology for projects devoted to the production of new technologies, also apply to the very development and production of new technologies, such as the implementations of WisTech or other technologies that implement models of IGrC, and models of reasoning about the properties of such a computing.

These difficulties mean that there may appear many possible variants of approaches to the establishment and development of a conceptual apparatus for this type of an ontology. Due to this diversity, it is advisable to draw our attention particularly to the basic principles governing the construction and development of an ontology for models of IGrC, as well as for models of reasoning about the properties of this computing, which together support the advanced implementations of WisTech. In conclusion it is t be noted that IGrC is a dynamically developing scientific discipline.

Therefore, when designing such computing models, one should bear in mind the flexibility of solutions that have been adopted in order to ensure that they are able to quickly adapt to the progress of knowledge and experiments. In addition, it is advisable to pay attention to the practical value of the established constructions and to the empirical verifiability of repetitive model properties. Especially, one must also understand and take into account the basic properties which distinguish these computing models from classical computing models (e.g., based on the Turing machines). In Sect. 36.1 of the book, a summary of such distinguishing properties of IGrC models and the Turing machine computing models is presented.

Given the above arguments, at the present stage of our research, it is not our ambition to find a comprehensive solution to BOT4IGC. In this book, we presented only some sample approaches, indicating a possible direction for further, more detailed research.

Taking into account the above considerations and previous experiments with different approaches to the implementation of WisTech (e.g., algorithmic trading described in Chap. 17), as a starting point, we will lay out some fundamental assumptions concerning the potential solutions to BOT4TGC from the perspective of different approaches to the implementation of WisTech. The assumptions are related to the properties of the WisTech ontology and its basic concepts. More precisely, in Chap. 32, we will propose some initial "approximations" of the WisTech ontological foundations, referred to as the FPW.

31.4 Logical Structures of Contemporary Mathematics and C-granules for IGrC Models

31.4.1 An Introduction to Logical Structures

Let us remind that the exemplary style of the contemporary approach to representation of logical structures of mathematical knowledge can be already found in "Elements" (about 300 BC) by Euclid of Alexandria.[3]

The formalization of mathematical knowledge, as presented in Euclid's "Elements", has absolutely dominated the way in which mathematics has been taught and developed over 23 centuries. A contemporary version of this style of formalization is a result of **evolution of David Hilbert's program**.[4] Roughly speaking, Hilbert proposed (in 1921) an approach to foundations of mathematics based on the following two main assumptions:

1. classical mathematics should be formalized in axiomatic systems;
2. using only restricted, "finitary" means, one should be able to provide proofs of the consistency of these axiomatic systems.

The fundamental idea behind the formalization of a domain knowledge in Euclid's style (and in Hilbert's style as well) may be roughly illustrated by a concept of *logical structure* presented in Sect. 31.4.1.1. In the Sect. 31.4.2 we explain some fundamental differences between the concept of c-granule and the concept of logical structure.

Contemporary logical structures of entire classical mathematics are expressed by the language of set theory (e.g., with the use of the concepts and axioms proposed by Ernest Zermelo and Abraham Fraenkel,[5] [292]).

One of the key concepts related to logical structures is the concept of *closure space* [351]. This concept can be treated as some kind of a generalization of concepts of *syntactic consequence* operator (i.e. it is based on formal rules for transformations

[3] http://aleph0.clarku.edu/~djoyce/java/elements/toc.html, https://en.wikipedia.org/wiki/Euclid%27s_Elements

[4] https://en.wikipedia.org/wiki/David_Hilbert.

[5] http://en.wikipedia.org/wiki/Set_theory.

of linguistic expressions and it does not depend on any semantic interpretation of the linguistic expressions) and *semantic consequence* operator of logical structure. This approach to consequence operator is related to ideas introduced by Tarski [163, 497]. Some results of initial research on the concept of closure operator one may find in [351].

31.4.1.1 The Problem of Characterization of Logical Structures of Contemporary Classical Mathematics

The formal concept of *logical structure* is a generalization of ideas introduced by Alfred Tarski [163, 497], Jan Łukasiewicz [511, 513], and Évariste Galois[6] [51, 374] (i.e., concept of *Galois connections* treated as a generalization of the correspondence between subgroups and subfields investigated in Galois theory [155, 317]). Presented approach to the concept of *logical structure* is based on formal definition introduced in [225], and developed in [229, 236].

In some papers logicians use the name *universal logics* for structures similar to *logical structures*. Many interesting results on *universal logics* and/or abstract *consequence operators* one can find in the anthology [49]. All the papers of the anthology are very valuable. However, from the point of view of development of logical structures, especially the following papers of [49] are interesting: [55, 96, 370, 428, 496, 520].

The studies of logical structures, in particular, have the following main objectives:

1. Deepening our knowledge and understanding in the scope of answers for the following questions:

   a. What do we mean by *contemporary logic*? What are conceptual relationships of *contemporary logic* to such concepts as: reasoning, semiotics, pragmatics, and ontology?
   b. What type of logical structures could be more effective and/or more efficient than logical structures of classical logic (especially for practical problem solving by humans and/or artificial agents in particular domain of applications)?
   c. Why the logical structures of the first order logic of classical predicate calculus are so useful and so widely used in contemporary science and/or engineering?
   d. Are there any hidden reasons for the special role of logical structures of classical predicate logic or is it just a coincidence?

2. Identification of any important *relationships* among logical structures corresponding to non-classical logics and/or logical structures of classical logic.

3. A characterization of logical structures corresponding to classical logic.

The results of author's research on logical structures and closure spaces are included in particular, in the following papers: [225–239, 242, 254]. This approach

[6] https://en.wikipedia.org/wiki/Galois_connection.

to the concept of logical structure has been the subject of research presented in particular, in [226, 228, 229, 236].

The basic components of the concept of *logical structure* are the following:

LS1. *Models* of domain knowledge, which are important for activities in the domain of knowledge. In particular, the models can be related to hunks perceived by agent and/or society of agents. These models may include also models of Euclidean geometry.[7] Another example of models can be based on objects described by attributes of a computer, relational data base, and/or objects of information systems [382, 384, 463].

LS2. *The indiscernibility relation* between models of a particular domain knowledge; the relation is based on some interactions which enable the identification and/or differentiation of *objects* that constitute models of domain knowledge. For example, it could be the relation of indiscernibility in the rough set approach [384, 463].

LS3. *Grammar of linguistic expressions*. Linguistic expressions are used to represent the properties of models that can be recognized by agents. Generally, the grammar of linguistic expressions includes rules for the construction, aggregation, decomposition, and transformation of linguistic expressions. Typically, such a grammar, defined in formalized languages of contemporary mathematical domains, includes:

  i. Specification of linguistic alphabet of the language and, in particular, specification of *primitive relations* and *primitive functions* characterizing of a given domain.
  ii. Specification of *atomic expressions*.
  iii. Specification of *rules for the construction* and/or for aggregation of *complex linguistic expressions*, using admissible logical connectives, symbols for relations (functions), and/or grammatical expressions. As examples of the grammar for language of expressions, one can consider the grammar for the language of Euclidean geometry or the grammar for the SQL-language, used to retrieve information from relational databases.[8]

LS4. *Satisfiability relation* between models and the language of expressions, which enables one to determine logical values of language of expressions in the particular models of domain knowledge. Satisfiability relation may be a current representation of results of interactive computation by implementation of interaction plans leading to current approximation of satisfiability relation.

LS5. *Approved domain knowledge (including schemes of reasoning)*, which can be represented by accepted statements describing the facts, phenomena, and *schemes of reasoning* related to a domain knowledge. The schemes describe some observed and/or confirmed facts and methods for construction of rules

[7] http://en.wikipedia.org/wiki/Euclidean_geometry.

[8] http://en.wikipedia.org/wiki/SQL.

for a proper reasoning about properties of models and properties of reasoning (i.e., meta-reasoning). For example, schemes of reasoning could include such rules as:

i. *Approved statements in the form of axioms* of a domain knowledge, which express basic and commonly accepted facts and/or principles of domain knowledge. For example, they may include axioms of Euclidean geometry.
ii. *Rules of deduction and/or induction* concerning properties of a knowledge domain. For example, these may include the modus ponens rule and/or some rules of generalization of empirical observations.
iii. *Rules for aggregation, decomposition, and transformation of reasoning.* For example, rules for combining reasoning into more complex forms or rules of reasoning *by "analogies between analogies"* in mathematics [515].

From the time of the emergence of non-classical logics (e.g., intuitionistic logic, many-valued logic logic, modal logics), logicians began to explore the possibilities of mutual interpretations of deductive systems of these logics. In the first half of the twentieth century, these studies focused on **propositional calculus**. In particular, interesting results in this area were reached by Glivenko [165], Gödel [167], Tarski [498], and Łukasiewicz [512].

However, let us remind that in ancient times Aristotle emphasized importance of the concept of quantification.[9] He developed a theory of quantification in the form of his well-known theory of syllogisms. In some sense, essential parts of the theory are present in modern classical first-order logic of predicate calculus.

Thus, the research about *relationships* among logical propositional calculus moved to the research among **first order predicate calculus** of logical structures corresponding to non-classical logics and/or logical structures of classical logic. Many results related to relationships among this kind of predicate calculus (e.g., intuitionistic, modal, many-valued logic, and classical predicate calculus) one may found in the book of Rasiowa and Sikorski [410], and also in the book of Rasiowa [409].

According to Daniel Bonevac,[10] it is possible to combine and to develop the strengths of the Aristotelian and modern approaches (to quantifiers and to predicate calculus):

> Modern quantification theory emerged from mathematical insights in the middle and late nineteenth century, displacing Aristotelian logic as the dominant theory of quantifiers for roughly a century. It has become common to see the history of logic as little more than a prelude to what we now call classical first-order logic, the logic of Frege, Peirce, and their successors. Aristotle's theory of quantification is nevertheless in some respects more powerful than its modern replacement. Aristotle's theory combines a relational conception of quantifiers with a monadic conception of terms. The modern theory combines a monadic conception of quantifiers with a relational theory of terms.**Only recently logicians combined**

[9] https://plato.stanford.edu/entries/quantification/.

[10] http://bonevac.info/papers/HistoryofQuantification.pdf.

relational conceptions of quantifiers and terms to devise a theory of generalized quantifiers capable of combining the strengths of the Aristotelian and modern approaches.

Study of modern approach to generalized quantifiers[11] were initiated by Mostowski [353]. In particular, he pointed out that there are many mathematically interesting quantifiers that are not definable in terms of the classical first-order quantifiers. Moreover, he characterized first order logic among extensions of the logic obtained by adding one so called simple unary generalized quantifier.

In other words, the research about logical predicate calculus moved to the research among **extensions of predicate calculus** (especially extensions of classical predicate logic by many approaches to generalized quantifiers),

At the beginning the research about generalized quantifiers was oriented to so-called *cardinality quantifiers*, *topological quantifiers*, and *links between logic and games*.[12] This research was not particularly relevant to natural language. However, later the natural language become important part of this research. Initial survey of this approach is presented in the paper of Barwise and Cooper [25].

Per Lindström in [309, 310] extended Mostowski's characterization on unary generalized quantifiers. Roughly speaking, the Lindström Theorem [309, 520] characterizes classical predicate calculus using two properties: countable compactness property and the downward Löwenheim-Skolem property. J. Barwise [24] extended Lindström's methods to the the infinitary model theory, combining them with ideas emerging from generalized recursion theory. His main accomplishment was a characterization of infinite languages of the form of $L_{\infty,\omega}$.

There are many other approaches to characterizes classical predicate calculus. For example, in [229, 236] the following result is presented: If L is a logical structure with a countable set of formulas, then L is embeddable in any structure logical of classical predicate calculus if and only if L is a logical structure which satisfies the compactness theorem.

31.4.1.2 A Galois Structure Approach to Information Flow Through Communication Channels in Distributed Systems

In the book by Barwise and Seligman [26] there is a very important next step toward better understanding of relationships among such important concepts like: Galois structures, computer science, and information flow through communication channels in distributed systems. In this book, one may find many illustrations of the Barwise and Seligman theory [26] applied to a wide range of phenomena, from file transfer to DNA, from quantum mechanics to speech act theory.

It is worthwhile mentioning that during the 1970s, Barwise and Seligman used the situation-theoretic approach to information. However, next they eliminated much of the technical machinery related to handling situations, relations and types, to obtain an abstract mathematical background of information flow. The starting point of their

[11] https://plato.stanford.edu/entries/generalized-quantifiers/.

[12] https://plato.stanford.edu/entries/logic-games/.

theory is the notion of a *classification* (in fact *classification* is exactly the concept of *Galois structure*, i.e., the concept of classification is the core of the concept of logical structure).

Their approach is based on the idea that information flow is realized by changes of properties of the system expressible by "types" and "tokens", where "tokens" represent observable phenomena or objects of the classification system. In other words they mainly develop machinery for analyzing of channels of communications related to those properties of systems, instead of handling situations in the situation theoretic approach to information.

31.4.1.3 Reasoning Schemes

Important component of any logical structure is *approved domain knowledge (including schemes of reasoning)* (mentioned in point LS5 above). There are many examples of very sophisticated schemes of mathematical reasoning (mentioned in point LS5 above). Classical mathematics mainly operates using such reasoning schemes as:

r1. *Deductive reasoning schemes* (i.e., **truth-preserving reasoning schemes**)—if assumptions are true, then conclusions are true.

r2. *Reasoning schemes based on the closed-world assumption*,[13] i.e., as a whole universe of contemporary mathematics, one can treat the von Neumann universe.[14]

r3. *Incremental character of mathematical knowledge,* i.e., knowledge is developed *incrementally*, that is any new knowledge is consistent with any piece of previously collected knowledge.

r4. *Formal character of the criterion of truth,* i.e., the truth value of mathematical sentences depends on the possibility to provide a *formal proof*, constructed using a widely accepted deductive system. In some sense, it means, that the truth of a sentence depends more on the accepted deductive system than on empirical experiments.

There are examples of very complicated formal proofs, which are far beyond the comprehension of any professor of advanced mathematics. For example, the proof of the Classification of Finite Simple Groups[15] [172, 173]. The proof is estimated [172] to be approximately ten thousand pages in length.

Moreover, even very "short" formal proofs are also far beyond the long life capabilities of almost all people (working outside a given field of mathematics), e.g. For example try (during your life) to understand the solution to Exercise 10 on page 331 in [259] (i.e., Freyd's proof of independence of AC) and/or try to understand the proof of one of the most widely known mathematical theorems, i.e., the Fermat's Last Theorem.[16]

---

[13] http://en.wikipedia.org/wiki/Closed-world_assumption.

[14] https://en.wikipedia.org/wiki/Von_Neumann_universe.

[15] https://en.wikipedia.org/wiki/List_of_finite_simple_groups, http://brauer.maths.qmul.ac.uk/Atlas/v3/.

[16] https://en.wikipedia.org/wiki/Fermat%27s_Last_Theorem.

r5. *Non-constructive character of the criterion of truth of the "existence" of mathematical objects.*; it means that numerous achievements of classical mathematics (e.g., related to object properties) were made by proving that their negation leads to contradiction (the use of the excluded middle law[17] or by applying the selection of one element from any set from a family of nonempty sets.[18]
Such proofs of existence do not have a constructive character (at least from the perspectives of engineering practice). Therefore, they do not provide the same level of truth value as constructively proven theorems, e.g., the Banach-Tarski's Theorem (cf. Sect. 12.3).[19]
From the point of view of practical applications, only such *objects* and their properties are important whose existence can be verified and confirmed by different people, using constructive techniques.

For sure, the aforesaid reasoning schemes constitute the core of the proof techniques known in the contemporary mathematics. However, from the practical point of view, other types of reasoning schemes are also very important. The following examples of reasoning schemes are key to any practical applications (especially in CSE projects and/or AI techniques):

R1. *Schemes of induction*[20]: These pertain to schemes of reasoning which **preserve the falsity of a sentence** (if assumptions are false, then conclusions are false). Intuitively, they include schemes of generalization, based on observations and verification by experiments.
R2. *Assumption of an open world*[21] instead of assumption of a closed world.
R3. *Knowledge adaptation*: As a consequence of iteratively repeated experiments based on often randomly collected information through interaction, knowledge adaptation becomes inevitable. New knowledge can be inconsistent with some pieces of previously collected knowledge.
There are many types of reasoning schemes for knowledge adaptation. For example, Aristotle spoke of the so-called [11] *defeasible reasoning* schemes.[22]
Intuitively speaking, *defeasible reasoning* is the process of accepting an argument because of what normally happens. It means that an argument does not need to be deductively valid.
R4. *Empirical character of the criterion of truth.* It means that the truth value of sentences regarding the physical reality as perceived by an agent is based on verifiable and repeatable results of experiments and/or repeatable "conceivable" constructions/experiments carried out in the agent's environment and/or in the agent's mind.[23]

[17] https://en.wikipedia.org/wiki/Law_of_excluded_middle.
[18] http://en.wikipedia.org/wiki/Axiom_of_choice.
[19] https://en.wikipedia.org/wiki/Banach-Tarski_paradox.
[20] https://en.wikipedia.org/wiki/Inductive_reasoning.
[21] https://en.wikipedia.org/wiki/Open-world_assumption.
[22] http://plato.stanford.edu/entries/reasoning-defeasible/.
[23] http://en.wikipedia.org/wiki/Thought_experiment.

R5. *Constructive character of the criterion of truth.* In practical applications, the **existence of any object property** typically means the ability to perceive this property by means of some measurement instruments (sensors). Thus, any "non-measurable" object property (like in contemporary mathematics—any properties of non-measurable sets[24] are often treated by engineers as a *leap of faith* and not as a component of *verifiable technical knowledge.* This means that engineers reject technical possibilities to construct billions of balls from one ball (according to the Banach-Tarski theorem).

Many forms of schemes mentioned in R2 and R3 has been intensively studied in the second part of the 20th century, especially in the context of potential AI applications. For example, during that time, there were developed many forms of *defeasible reasoning*. The key points of this development are summarized in the article.[25]

It says that:

> McCarthy and Hayes (McCarthy and Hayes 1969) developed a formal language they called the "situation calculus," for use by expert systems attempting to model changes and interactions among a domain of objects and actors. McCarthy and Hayes encountered what they called the frame problem: the problem of deciding which conditions will not change in the wake of an event. They required a defeasible **principle of inertia: the presumption that any given condition will not change, unless required to do so by actual events and dynamic laws. In addition, they encountered the qualification problem: the need for a presumption that an action can be successfully performed, once a short list of essential prerequisites have been met**. McCarthy (McCarthy 1977, 1038–1044) suggested that the solution lay in a **logical principle of circumscription**:the presumption that the actual situation is as unencumbered with abnormalities and oddities (including unexplained changes and unexpected interferences) as is consistent with our knowledge of it. (McCarthy 1982; McCarthy 1986) In effect, McCarthy suggests that it is warranted to believe whatever is true in all the minimal (or otherwise preferred) models of one's initial information set.
>
> **In the early 1980's, several systems of defeasible reasoning were proposed by others in the field of artificial intelligence:** Ray Reiter's default logic (Reiter 1980; Etherington and Reiter 1983, 104–108), McDermott and Doyle's Non-Monotonic Logic I (McDermott and Doyle 1982), Robert C. Moore's Autoepistemic Logic (Moore 1985), and Hector Levesque's formalization of the "all I know" operator (Levesque 1990).

In the second half of the 20th century, the research in the area of defensible reasoning (especially in terms of AI applications) continued to be the subject of interest. In particular, the direction of "adaptive logic" become quite popular.[26]

The list of achievements made as part of the research on *defeasible reasoning* is much longer and even more impressive. However, in the engineering practice related to the design and development of complex technologies for non-trivial applications of AI, **formal methods of defeasible reasoning are not efficient** and hence they are not very popular.

---

[24]https://en.wikipedia.org/wiki/Non-measurable_set.

[25]http://plato.stanford.edu/entries/reasoning-defeasible/.

[26]http://logica.ugent.be/adlog/al.html,
www.academia.edu/613799/Adaptive_Logic_Characterizations_of_Defeasible_Reasoning_With_Applications_in_Argumentation_and_Default_Reasoning.

Practical engineering is mainly based on practice and verifiable technical adaptive knowledge. The knowledge is acquired and developed on the basis of iterative and interactive development and through the integration of relevant granules of knowledge. In these processes "inductive" types of reasoning are especially important. Instead of principles of formal calculus for reasoning schemes, in engineering practice, "constructive" and "empirical" techniques (including "thought experiments"[27]), aimed at verifying the truth, constitute the basic criterion of the truth value and judgment regarding the existence of an object property.

Let us remind that from the perspective of mathematical foundations, at the turn of the 19th and 20th century, the **reasoning schemes mentioned above, in point R5, bring us closer to the foundation of intuitionism** (cf. Chaps. 24 and 25). We mainly refer to the version of intuitionism proposed, more than one hundred years ago, by Luitzen Browuer[28] [524, 525]. He referred to, among others, studies by Henri Poincaré, Leopold Kronecker, and many other mathematicians, logicians, and philosophers as part of a movement which is called preintuitionism.[29]

Mathematics based on Brouwer's intuitionism is not an inherent part of the presently dominating approach to the foundations of contemporary mathematics. The framework of mathematics, at the beginning of the 21st century is inspired by the development of the Hilbert's Program[30] from the beginning of the 20th century.

Moreover, it should be noted that the above considerations **touch upon the methodological foundations of the whole contemporary science, which is based on contemporary mathematics**.

These considerations serve as an illustration of **only some difficulties concerning the complexity of fulfilling main FPW objectives**—as specified in Sect. 32.1 (cf. WT1,WT2, and WT3).

Referring to point WT3 (cf. Sect. 32.1), the development of formalization requires a very deep exploration, discovery, and comprehension of "empirical" and "theoretical" principles, governing the domain of represented knowledge. For example, such process of **"empirical" and "theoretical" exploration, discovery, and comprehension of Euclidean geometry lasted for millenniums**.

31.4.2 Fundamental Differences Between Concept of C-granule and Concept of Logical Structure

Notice that there is are conceptual relationships between the concept of c-granule and the concept of logical structure. Namely, the logical structure may be interpreted as a c-granule using the following rules:

[27] http://en.wikipedia.org/wiki/Thought_experiment.

[28] http://en.wikipedia.org/wiki/L._E._J._Brouwer.

[29] http://en.wikipedia.org/wiki/Preintuitionism.

[30] https://en.wikipedia.org/wiki/Hilbert%27s_program.

a. LS2, LS3, and LS5 may be interpreted as a components of a soft_suit.
b. LS4 may be interpreted as a components of a link_suit.
c. LS1 may be interpreted as a components of a hard_suit.

However, it should be emphasized that **there are many fundamental differences between the presented concept of c-granule and the concept of logical structure**.

First of all, the c-granule should satisfy the principles described in the framework postulates for WisTech (FPW) (cf. Chap. 32) or at least it should satisfy a simplified version of FPW described by principles igc-1-igc-18 in the Sect. 36.1.

Let us present the following examples of arguments for an illustration of the fundamental differences between the concept of c-granule and the concept of logical structure:

1. $\underline{C - granules}$ are used to generate **computational building blocks for perception** [522] **(used e.g., as patterns for approximation of complex vague concepts used for control of agent's interactions)**. Such building blocks are complex and they require hierarchical modeling leading toward such relevant blocks.
   The *computational building blocks for perception* may represent a family of interaction plans constructed from adaptive interrelated *interaction rules*.
   On each level of modeling different logical structures may be relevant. These logical structures should be discovered and continuously adapted (based on results of interactions). They are not given and/or defined as in the case of the current mathematical logic. The discovery process includes discovery of semantic structures (e.g., relevant relational structures) as well as discovery of language of formulas for expressing relevant properties of the structures defining mentioned above computational building blocks. Moreover, the discovered logical structures are different from the used in mathematics so far.
   $\underline{Logical\ structures}$ **are not directly relevant to perception processes**.
2. $\underline{C - granules}$ **are grounded on physical objects continuously interacting** with and open environment.
   $\underline{Logical\ structures}$ **are formal models existing in static and closed world** of von Neuman universe[31] of sets. Logical structures are not involved in any real interactions with the physical environment.
3. $\underline{C - granules}$ **are used by agent's control and/or by other c-granules for reasoning** about interactions in dynamically changing open environment in which the agent is involved. Agent has only incomplete and uncertain knowledge about the environment. The environment is a subject of continuous changes. Very often the changes may be unpredictable.
   $\underline{Logical\ structures}$ **are embedded in static and closed environment**. They are not physically directly linked to any particular agent. Theoretically they can by used by any agent.
4. $\underline{C - granules}$ are not applicable for all situations in an environment, they are pointing out to the distinguished fragments of the spatiotemporal space. They **have to be continuously discovered, constructed, developed, assessed, and**

[31] https://en.wikipedia.org/wiki/Von_Neumann_universe.

improved by agent's control. This is due to the fact that they should control commutations toward satisfying the current needs of the agent. The control is learning how to construct and use possibly optimal c-granules related to given domain of applications. Typically, the construction of a c-granule is result of interactions and reasoning mainly controlled by agent's control. Agent's control has to discover optimal concepts, language (in particular new attributes), and reasoning schemes for applications in important domains of his/her activity. They are represented by c-granules.

$\underline{Logical\ structures}$ **are "given" form other logicians and/or they have to be discovered by a mathematician and/or logician**. Typically the construction of a logical structure is the result of an abstract reasoning by mathematicians using other mathematical constructions.

5. $\underline{C-granules}$ **cooperate in large, dynamically changing, and interacting network** of linked c-granules. Typically many parts of the network have hierarchical structure. For complex applications the network structure should be discovered and constructed by agent's control.

   $\underline{Logical\ structures}$ in contemporary applications typically **are used alone**.
6. $\underline{C-granules}$ **are used to represent an agent's hierarchy of needs**[32] **and agent's reasoning schemes** for construction of interaction plans for satisfying the needs. Based on interactions of some c-granules of control with environment agent's control is judging the importance of the top need in given situation. Agent's judgmental is subject of continuous adaptation based on the assessment of used versions of judgment.

   $\underline{Logical\ structures}$ in general case **are not related to interactive perception of phenomena in physical world**. Thus the concept is not related to *adaptive judgment* in the sense of c-granules.
7. $\underline{C-granules}$ **can not be precisely expressed using formal language**. An agent (or other c-granule) can try to perceive and/or understand (to a degree) the meaning of behavior of a c-granule by:

   a. Observation of results of interactions relevant to the c-granule and/or interpretation of the results.
   b. Learning about the c-granule features from other agents and/or other sources of knowledge (e.g., c-granules).
   c. Taking into account that typically in the learning process are involved complex and/or vague concepts (e.g., concepts of a natural language).
   d. Accepting that meaning of words (especially words representing of complex and/or vague concepts) depends on interpretations of words by "sender" and "receiver". Following Wittgenstein [539] approach to the concept of "language game", we treat meaning as a "social event", i.e., *meaning happens between language users*. In other words the question: "what is a word really?" is similar to question "what is a piece of chess"? In consequence, in

[32] https://en.wikipedia.org/wiki/Maslow's_hierarchy_of_needs.

particular, limits of language communication depend on "language games rules" among "language users."

e. Taking into account that typically, the perception and/or understanding of the meaning of behavior of a c-granule can be only partial. However, it can be interactively improved using discovery and approximation of corresponding concepts (e.g., concepts for expressions of *pre-conditions, expected post-conditions, and actions* of interaction rules). The essential part of this process consists of approximation with satisfactory quality of perception and/or understanding of the behavior. The approximations together with the accumulated knowledge by the agent should allow her/him to induce new interaction rules [521, 522] describing features of interactions in the scope of the c-granules.

f. Following consequences of the fact that the essential feature of very complex c-granule is an ability to perform adaptive judgment and causal reasoning for construction of "optimal" interaction plans. According to J. Pearl classical tools (e.g., based on causal inference in statistics) are not sufficient solution (cf. Pearl's motto of the book) . He has following advice[33] for data scientist and/or for c-granule engineers: *the paradigmatic shifts that must be undertaken in moving from traditional statistical analysis to causal analysis of multivariate data. Special emphasis is placed on the assumptions that underly all causal inferences, the languages used in formulating those assumptions, the conditional nature of all causal and counterfactual claims, and the methods that have been developed for the assessment of such claims. Causal inference requires two additional ingredients: a science-friendly language for articulating causal knowledge, and a mathematical machinery for processing that knowledge, combining it with data and drawing new causal conclusions about a phenomenon.*

Logical structures **are expressed in formal mathematical language which exists in statics and closed universe** of mathematical sets.

[33] http://ftp.cs.ucla.edu/pub/stat_ser/r350.pdf.

Chapter 32
Framework Postulates for WisTech (FPW)

Good mathematicians see analogies between theorems or theories, the very best ones see ***analogies between analogies****.*
Stefan Banach [515]

32.1 The Main Objectives of FPW

One of the **main objectives of WisTech** is

to provide some conceptual tools supporting the construction of models for CSE projects implementation and/or for the development of AI technologies as complex adaptive interactive systems, which aggregate many local models interacting in an open and changing environment, along with the techniques for reasoning about important properties of the behavior of such models.

According to what has been implied in Sect. 1.2.2 and Chap. 13, this objective is directly linked to the FP3C and to TPGP, as defined in Chap. 11.

Moreover, as we have discussed in Chap. 31, even the very simple designing of an ontology for practically useful IGrC models is a highly complex task. Based on the IGrC and CSE development synergy hypothesis discussed in Sect. 31.1, finding any appropriate solution to this problem could be very beneficial for the development of CSE and AI technology.

Thus, the basic ontological task for IGrC (BOT4IGC), defined in Sect. 31.3, is very important. Moreover, we should underline that *the complexities are the essence* of BOT4IGC. We should be aware of warning by Brooks [68] (and others, e.g., [158]) that contemporary mathematics (and thus, the whole science based on mathematics), in general *does not work when the complexities are the essence*. Unfortunately, as this is exactly the our concern case, we have to be very careful and do not expect to find any simple "fast solutions."

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_32

In this chapter, we propose a conceptual framework for WisTech in the form of *Framework Postulates for WisTech* (FPW). At the current stage of the WisTech research, the main objectives of FPW are as follows:

WT1. Constructing the ontological base for WisTech. This particularly concerns **the concepts used in FPW and the properties of mutual relationships among them, which create the initial "approximations" of the ontological foundations of WisTech**.

WT2. Identifying possible **directions of interactive development for the architectural framework of the WisTech implementation models**, based on the models of IGrC.

It should be underlined that FPW postulates require further clarification and more detailed development before they are applied to potential WisTech implementations. In general, a significant part of this book (especially within the scope of WisTech) indicates potential directions for interpretation, development, and explanation of FPW.

WT3. Identifying **potential directions and related possibilities or difficulties stemming from the development of WisTech "formalization" and "axiomatisation,"** especially from the perspective of the experiences gained during the development of mathematics over years.

The objectives presented above, in points WT1 and WT2 are quite clear. However, in order to better explain the intuitions included in the third point, WT3 (concerning the WisTech "formalization" and "axiomatisation"), let us remind that the exemplary style of the contemporary approach to formalization of the mathematical knowledge can be already found in "Elements" by Euclid of Alexandria[1] (cf. Sect. 31.4).

32.2 General Structure of Packages of FPW Postulates

Referring to FPW, it is necessary to underline that at the present stage, we have only some initial proposals regarding the computing models for WisTech based on IGrC. Once these models are prepared in detail, **they should enable us to explore, discover, and comprehend their principles and consequently, gain "empirical" and "theoretical" knowledge of WisTech based on IGrC.**

WisTech Framework Postulates (FPW) are presented in the form of packages of postulates, FPW-01–FPW-16. All postulates have a framework-like character and can be interpreted, elaborated on, and implemented in numerous ways. These packages are divided into the following two groups of postulate packages:

[1] http://aleph0.clarku.edu/~djoyce/java/elements/toc.html, https://en.wikipedia.org/wiki/Euclid%27s_Elements.

A. **Physical character of an agent, c-granules, and interaction models**. This group includes packages of postulates from FPW-01 to FPW-09.
B. **Managing agent's need satisfiability, effectiveness, and efficiency**. This group includes packages of postulates from FPW-10 to FPW-16.

The formal difference between these two groups of postulates consists of the fact that the description of each postulate from group B is further divided into two parts:

I. **Abstract** that presents the main message of a given package of postulates.
II. **Exemplary refinement** that introduces refinements to FPW in the scope of a particular package of postulates.

Such differentiation results from the fact that group B is presented on a lower level of generality and refers, to a larger extent, to potential techniques of implementation.

Moreover, in the exemplary refinement of postulates from group B, we refer to metaphors from the fields of economy and finance. However, it has to be remembered that the "economical and financial" metaphor used to explain FPW-10–FPW-16 postulates is only one of many possibilities to illustrate the concept. Such an approach was adopted mainly given its importance for the central theme of this book, i.e., the combination of risk management in CSE and the development of new AI technologies.

The aforesaid packages of postulates, FPW-01–FPW-16, are discussed in the following sections.

The FPW, in particular describes an approach to long lasting research subject on relationships between mind and body. There are many publications and research portals presenting discussions about this issue (cf. e.g., the popular portal "Portal: Mind_and_brain"[2] dedicated to Wikipedia's resources on philosophy of mind). For beginners an introductory book [429] can be quite interesting.

The discussions about relationships between mind and body were quite intensive in ancient Greece. As a consequence, it can be a good idea to consider the FPW in the context of the Aristotle Tetrahedron presented in Fig. 3.7 and related Figs. 3.5 and 3.6 in Chap. 3 (cf. [122, 245, 294, 371]).

Having in mind Fig. 3.7 one can treat a person, reading and/or developing the FPW, as the Interpreter (soul, psyche). She/he uses some Symbols or signs "THOUGHT OR REFERENCE"(e.g., "objects" ("pragmata") or "things" of which our experiences are images ("symbola") and/or "mental images" of perception (e.g., affections or impressions ("pathemata") of the soul ("psyche")). The FPW is expressed in a language which should be known (to a satisfactory degree) to the current Interpreter. In general, the Interpreter can be any Reader and/or Agent who understands (to a satisfactory degree) the language of the author. An analogous situation can be considered, for any author involved in development of FPW. In general, the author can be any person and/or agent who uses an appropriate language of FPW. Assuming the Wittgenstein [539] approach to the concept of language one can treat the meaning of the language as the result of a "language game" [166, 539] of society of

[2] https://en.wikipedia.org/wiki/Portal:Mind_and_brain.

communicating agents. The rules of the game constitute an interactively developing family of rules (not a static class/set of rules). Thus, the meaning of expression of any language (in particular the language of FPW) is not static. This happens between language users who use current version of rules of the language game. Typically at the beginning of any dialogue one version of rules is used by emitter and often different version of rules is used by receivers. However, after interactive exchange of "language expressions" they establish better approximation of commonly used rules of the language game. Let us underline that according to this approach following any rule is a social activity and usually it is to be a subject of change.

32.3 Physical Character of Agent, C-granule, and Agent's Perception

32.3.1 FPW-01: Physical World

1. *The physical world* consists of **spatiotemporal** *physical beings*.
2. Physical beings may *interact* with each other.
3. *Interactions* satisfy some of the *cause-effect relationships* from the physical world, i.e., they are according to the *natural laws*.
   i. Cause-effect relationships may constitute *interaction invariants*.
   ii. Cause-effect relationships can be *non-deterministic*.
4. Any physical being can consists of some other physical beings, referred to as the *parts of a physical being.* Any part of a physical being is also a physical being.

32.3.2 FPW-02: Agent's Networks of Interactions

1. An agent is a *physical being* that is able to **initiate and modify some interactions**, using other physical beings called the *agent's controllable assets*.
2. *Agent's controllable assets* can be either physical beings in the *agent's external environment* and/or they can be *agent's parts*.
3. *Agent's controllable assets* consists of:
   i. *Agent's memory*, including:
      a. *agent's fast access memory*,
      b. *agent's slow access memory*.
   ii. *Agent's spatiotemporal maps being parts of agent's memory*.
   iii. *Agent's networks of interactions*. The following types of *agent's networks of interactions* are especially important to any agent:
      a. *Agent's control*.

  b. *Agent's private clocks*.
  c. *Agent's sensors*.
  d. *Agent's actuators*.
  e. *Agent's supply chains* (e.g., energy supply, chemical compounds supply).

4. The interactions of *agent's controllable assets* satisfy *agent's cause-effect relationships* specific to a given agent. The interactions, consistent with the relationships, proceed according to the *natural laws*. They are based on the agent's conditions (i.e., components of causes of the agent's cause-effect relationships). For example, it is impossible to initiate interactions by an agent who has no energy resources to do this.
5. An *agent's spatiotemporal map* is a partial representation of some components of the *agent's environment*, which may include a partial representation of some interactions, recorded by the *agent's memory*.
6. An *agent's network of interactions* consists of the following parts:

  a. *Vertices of interactions*.
  b. *Transmission channels*.
  c. *Vertex interfaces* connecting *vertices of interactions* to *transmission channels*.

7. *Transmission channels* are used as the carriers of interaction effects among *vertices of interactions*.
8. *An agent's control* consists of *agent's controllable assets* which support interactions in the *agent's control*. The assets are called *agent's control components*. The interactions among the components of the agent's control preserve laws of nature, determined by the *agent's cause-effect relationships*. Among them, there are relationships specific to the *agent's control* (e.g., principles of behavior of neural nets in a human brain).
9. *The agent's control* can use *transmission channels* and *vertices of interactions*, connected to transmission channels, in order to meet the following purposes:

  i. Access, verify, and modify other *agent's parts* and/or physical beings in the physical world.
  ii. Record, read, and/or modify parts of the *agent's memory*, by transmitting some effects of interactions.
  iii. Verify the *satisfiability* (to a *satisfactory degree*) of:
    a. some *causes* of cause-effect relationships that led to interactions,
    b. some *effects* of cause-effect relationships produced by interactions.
  iv. Initiate interactions after identifying some effects of interactions among physical beings.

10. The *agent's control* can use the agent's memory in numerous ways. One can assume that:

  i. The agent's fast access memory constitutes a part of the agent's control.
  ii. An agent can use her/his transmission channels to extend the agent's memory.

iii. The agent's *spatiotemporal map*, or shortly the *map*, has the following properties:
   a. The map is a part of the *agent's memory*. The map is continuously updated over the course of the *agent's private time*.
   b. The *map* has parts related to specific types of the *agent's time*. These types are used to distinguish the time or the duration of effects opf interaction that were recorded or planned. We can differentiate between the following types of *agent's spatiotemporal maps: past map, present map*, and/or *planned map*.
   c. *Past map* results from recording (over the *agent's time scale*, measured by the agent's private clock) properties of inter-actions that were transmitted in some parts of the agent's memory.
   d. *Planned map* can be used by the agent's control for the *what-if analysis* [264, 533, 534].
   e. The agent's control uses the agent's spatiotemporal map for the *agent's navigation* in accessible parts of the agent's environment and/or navigation in the *agent's memory*.

11. When it comes to agent's *transmission channels*, the *agent's control* has the ability to do the following:
   i. Activate a transmission channel, provided the channel is not active and it is ready for activation by the agent control.
   ii. Deactivate a transmission channel, provided that the channel was activated by the agent's control.
   iii. Construct and/or reconstruct a transmission channel, provided that a representation of the channel is stored in the agent's memory and/or in the agent's external memory.

12. *Vertices of interactions* connected to the *transmission channel* C, can be classified as follows:
   i. *Sending vertices* of C, i.e., sources of interaction effects transmitted by the channel.
   ii. *Receiving vertices* of C, i.e., receivers of interaction effects transmitted by the channel.
   iii. *Mixed vertices* of C, i.e., vertices which can be both *sending vertices* and *receiving vertices*.
   iv. *Out-of-order vertices* of C, vertices which cannot work properly as *sending vertices*, *receiving vertices*, and *mixed vertices*.

13. The *agent's control* may use the *transmission channel* C as the *agent's pointing-arrow* (cf. also the concept of *Semantic Pointer Architecture* in [126]). In order to do this, the *agent's control* has to distinguish the following two types of *vertices of interactions*, connected to the *transmission channel* C (cf. Fig. 32.1):

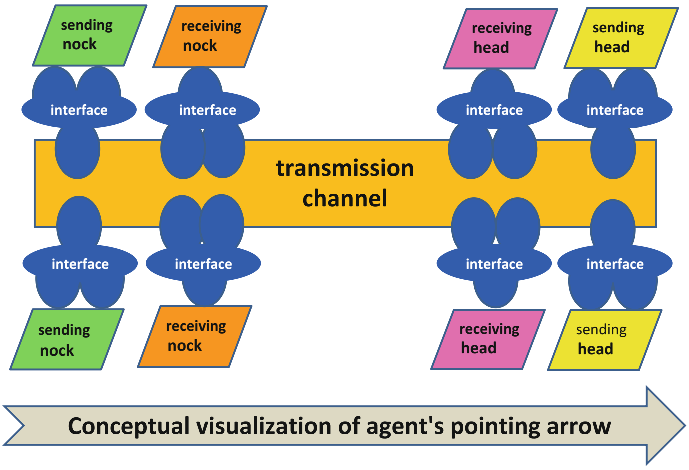


Fig. 32.1 Agent's pointing arrow

i. The *heads* of the channel C are vertices at which the transmission channel C ends (according to the *agent's control*).
ii. The *nocks* of C are vertices at which the transmission channel C begins (according to the *agent's control*).

In such a situation, by the *agent's pointing-arrow*, we mean an aggregation of the following components:

a. The *transmission channel* C.
b. All interfaces to the *heads* of C.
c. All interfaces to the *nocks* of C.

14. Any *head* of C and any *nock* of C, can be at once a *sending vertex* of C and/or a *receiving vertex* of C at the same time. In other words, we may have: *sending nock, receiving nock*, *sending head*, and *receiving head*. The conceptual visualization of such a situation for an agent's pointing arrow is presented in the Fig. 32.1.
15. An agent's *pointing-arrow* is called an *agent's base link*, referred to, in short, as a *base link*, provided that all the *nocks* of a given channel are parts of the agent's memory and/or they are parts of the *agent's control component*.
16. Vertices, heads, nocks, and interfaces of an agent's base link are called respectively: *base link vertices*, *base link heads*, *base link nocks*, and *base link interfaces*.
17. Any *base link*, connecting some parts of the agent's spatiotemporal map and parts of the spatiotemporal space, is called the *base link to spatiotemporal location*. Any base link head of the *base link to spatiotemporal location* is called the *base spatiotemporal location*.

32.3.3 FPW-03: Agent's C-granules

1. Let us assume that:

   i. L is an *agent's base link*.
   ii. $\mathscr{N}$ is the family of all *nocks* connected to L.
   iii. $\mathscr{H}$ is the family of all *heads* connected to L.

   A *base c-granule* G is an aggregation, by the *agent's control*, of the following three components: L, $\mathscr{N}$, and $\mathscr{H}$. In such a situation, we use the following terminological conventions:

   i'. The family $\mathscr{N}$ is called the *soft_suit of* G, or, shortly, *soft_suit*.
   ii'. The *base link L* is called the *link_suit of* G, or, shortly, *link_suit*.
   iii'. The family $\mathscr{H}$ is called the *hard_suit of* G, or, shortly, *hard_suit*.

   In general, the concepts of *c-granule* and *links* can be introduced in an inductive manner, that is:

   i". Any *base c-granule* is also a *c-granule*.
   ii". Any *base link* is also a *link*. Any link is also a *pointing arrow*.
   iii". The agent's control may select c-granules and/or agent's links, and can aggregate or reconfigure them to get the following:
      a. *Higher order* and/or *base agent's links*, connected to *c-granules* and/or other physical beings.
      b. *Higher order* and/or *base agent's c-granule* G.

   Notice that the concepts of *c-granule* and *link* can be considered in relation to the *agent's private time*, i.e., it is measured by the *agent's private clock*.
2. Let $\mathscr{G}_G$ be a family of all c-granules which are the components of a c-granule G and let $\mathscr{L}_G$ be a family of all links which are the components of the aggregated c-granule G.
   The aggregated *c-granule G* consists of the following three *architectural layers* of G:

   i. *Soft_suit of G* or, shortly, *soft_suit* consists of a *network of interactions* of all c-granules from the family $\mathscr{G}_G$, which are also the *link nocks* of links from the family $\mathscr{L}_G$.
      The c-granules of the soft_suit are linked to one another by the links from the c-granules (which are the components of the soft_suit).
   ii. *Link_suit of G* or, shortly, *link_suit* consists of all links from the family $\mathscr{L}_G$
   iii. *Hard_suit of G* or, shortly, *hard_suit* consists of the *network of interactions* between all link heads of the links from the family $\mathscr{L}_G$ (which can be also some c-granules from the family $\mathscr{G}_G$).
      The c-granules of the hard_suit are linked to one another by the links from the c-granules (which are the components of the hard_suit).

3. By the **rank of a c-granule** (respectively the **rank of a link**), we mean a measure of the aggregation level, i.e., *base c-granules* and *base links* have order 0, next, their aggregations have order 1, and next, their aggregations are defined recursively.
4. Any c-granule G may use its *architectural layers* for the implementation of the following functions:

   i. *Soft_suit*:

      a. In the soft_suit, there are encoded procedures for:
         - *Recording* certain effects of interactions (using agent's sensors over the selected fragments of the spatiotemporal space), which are transmitted by the links from $\mathscr{L}_G$.
         - *Representing* recorded effects of interactions, as (partial) information (available to the links from $\mathscr{L}_G$) about the environment and the history of effects of interactions in the environment and, in particular, in the agent's memory.
         - *Classifying* the effects of interactions (transmitted by links from $\mathscr{L}_G$). Then, the effects are recorded in appropriate parts of the agent's memory.
         - *Initiating interactions* (using agent's actuators over the selected fragments of the spatiotemporal space) by links from $\mathscr{L}_G$, according to the specification available in the agent's memory and indicated by the agent's control.
      b. In the soft_suit, a representation of the data, subject to the procedures encoded in the soft_suit, can be stored.
      c. In the soft_suit, a representation of the *expected effects* of interactions, monitored and/or initiated by the links from $\mathscr{L}_G$, is recorded.
         It may happen that the interactions initiated by the links of the c-granule (in particular, the interactions in the hard_suit) may be disrupted by unpredictable interactions of the hunks with the environment. This may cause that the *expected effects* of these interactions, carried out by the agent's control, will be different from the actual ones.

   ii. *Link_suit*:
      a. We assume that for any link in the link_suit, it is possible to localize the *link head* in the agent's time space.
      b. The transmission of the effects of interactions by any link depends on the physical components of the link and the environment.

   iii. *Hard_suit*:

      a. It consists of all physical beings (link heads) linked to the c-granule G by the links from $\mathscr{L}_G$. Intuitively, it is the whole physical world of the c-granule, available to the link nodes of the links from $\mathscr{L}_G$.

b. The destination of a given link can be at the same time a part of the soft_suit and/or a part of the link_suit. Thus, some parts of the hard_suit can be also parts of the soft_suit or link_suit.

5. The c-granules, in turn, can be composed of other c-granules, which provide representations of: *configurations of physical beings*, *properties of interactions*, *perception attributes*, and/or other *c-granules*, in the agent's memory.
6. i. For each agent's sensor, there exists at least one agent's link whose link head is a sending vertex, being a part of the sensor.
   ii. For any agent's actuator, there exists at least one agent's link whose link head is a sending vertex, being a part of the actuator.
7. Some of the agent's c-granules may be active during the time of the *agent's life*.
8. Some of the agent's c-granules can be activated (or deactivated) by the agent's control and/or by another c-granule. An activated (c-granule is called an *active c-granule* (respectively, a deactivated c-granules is called an *inactive c-granule*). An active c-granule may process, modify, and/or produce other c-granules. An agent can store representations of some inactive c-granules in the agent's memory.
9. Any c-granule can be a part of another, more complex c-granule, e.g., in a form of a hierarchical network of c-granules, interconnected by the agent's links.
10. Any c-granule belongs to at least one agent, who is the *owner of the c-granule*.
11. A society of interacting agents may produce *meta-social c-granules*. In such a case, each agent from the society forms her/his own *interpretation of the meta-social c-granule* on the basis of other c-granules.
12. The *meta-social c-granules* can belong to a *meta-social agent* (or *meta-agent*, in short), composed of the agents that form a given society. The control of a meta-agent, obtained in this way, treats the meta-social c-granules as c-granules of a meta-agent. Each agent from the society of agents, used in the construction of a meta-agent, may be treated by such agent from this society as her/his c-granule. Typically, the meta-social c-granule can be accessible only partially by a meta-agent.

32.3.4 FPW-04: Perceivability and Hunk Configurations

1. The *agent's control* has the ability to use some c-granules to initiate interactions encoded in the soft_suit. The interactions are initiated following an identification of some features of interactions among the physical beings and/or c-granules. This ability is called the *agent's perceivability*.
   The *agent's perceivability* is implemented, using the *agent's private time* and relevant interactions among the agent's control, agent's memory, and/or other physical beings.

2. The *perceivable beings* and/or *perceivable interactions* among them are accessible by the agent's control by means of the *agent's spatiotemporal windows* or, shortly, *windows*, which are specified in the agent's memory.
   The *agent's window* restricts the agent's potential perceiving of physical beings and/or interactions. Fragments of the spatiotemporal space are selected by the links of c-granules.
   Intuitively speaking, the *agent's window* can be treated as an aggregation of interfaces of link heads.
3. Agent uses the *agent's windows* for perceiving, recording, simulating, initiating interactions, predicting relationships between some elements of the physical world and as an interface with interactions occurring in the environment. The result of the agent's activity of perceiving the physical world through a given window at the agent's time point t is called the *state (of the physical world restricted by the window)* at t.
4. On the basis of the *agent's perceivability*, c-granules enables an agent to *perceive* (by the agent's control, i.e., also by the agent) the spatiotemporal physical objects through the agent's spatiotemporal windows.
   Any physical being (spatiotemporal portion of the matter) perceived by an agent's c-granule is called the *agent's hunk* [201] (or *hunk*, in short). Typically, each hunk is linked to a specific spatiotemporal subspace.
5. An agent perceives the physical world, using interactions among:

   i. the c-granules composed of hunks,
   ii. hunks and/or interactions in the agent's environment, and
   iii. interactions activated and/or monitored by c-granules.

6. *Imperfectness of perceivability*. Due to the partial ability of perceiving physical beings and/or interactions and given the uncertainty when identifying the sources of interactions, **the perceivability can be imperfect or incorrect**.
7. All beings and interactions perceivable by an agent, as well as all interactions within the scope of potential activities of an agent, create the *agent's environment*. Thus, hunks which constitute a part of the agent's environment can be perceived by an agent.
8. An agent may perceive hunks by recording in her/his memory the results of aggregated consequences of interactions between its sensors/actuators and the environment, in which hunks are located.
   The results of such an aggregation, as perceived by an agent, are known as the *properties of hunks*.
   Hunks have a spatiotemporal character. Thus, the properties of hunks also have spatiotemporal character and may be very complex. For example, these may include various dynamic complex models of such spatiotemporal hunks.

9. An agent may interpret various properties of hunks. One can distinguish two types of such properties of hunks:

   i. *Dynamic properties*. These are the properties of hunks perceived by agents which cannot be perceived within the smallest slots (intervals) of the agent's time, treated as the moments (units, slots) of her/his time. *Dynamic properties* are formed through the aggregation of properties over numerous moments of the agent's time, within some period of time or some dimensions of spatiotemporal space. Intuitively, these are the properties of hunks which an agent cannot perceive in a photograph. However, they can be perceived in a movie.
   ii. *Static properties*. These are the properties of hunks perceived by agents which are not dynamic. This means that these properties can be perceived over the above-mentioned smallest slots of time, i.e., the properties which within such short moment of time an agent can distinguish. Intuitively, these are the properties of hunks which an agent can perceive in a photograph.

10. Agent can group sets of specified hunks (using some properties) over different moments of their private time.
    Such groups of hunks and some of their properties, ocurring during specific moments of time, are called *hunk configurations*.
    Notice that **the same set of hunks may have different configurations relative to different moments of time**.
11. For the agent's control, a c-granule represents particular *hunk configurations*, along with certain results of interactions between these hunks and parts of the environment.
    Hunks and the hunk configuration may concern different *agent's time types*, e.g., past, present, or future (relative to the agent's clock) of interactions in the physical world (related to the internal agent's private clock). Hence, the interactions may be classified by the agent's control, using the *agent's time types*, as *past, actual*, and *planned/forecasted*.
12. Configurations of hunks consisting of a c-granule are called *configurations of a c-granule*.
13. Agent's interactions with the environment are conducted, using *configurations of c-granules*. The agent's control use the procedures encoded in the soft_suit of a c-granule, and sometimes procedures encoded in other c-granules, to control the interaction, using links of a c-granule and sometimes links of other c-granules. The configurations of c-granules encode any important information available to the agent's control and c-granules.
14. The *configuration of a c-granules* evolves due to interactions with their environment, including the agent's control. This leads to changes in the existing configuration of c-granules by following means:

    i. Extending them with new c-granules, selected by the agent's control responsible for perceiving new interactions (also stimulated by c-granules).

   ii. Extending the configuration with new c-granules, encoding the results of the perceived interactions.
   iii. Deleting some c-granules from the current configuration.
   iv. Introducing other modifications to some parts of the c-granule configuration.

15. The c-granules can be used to monitor and/or activate interactions. Through the initiation of the interactions, an *agent aims to achieve* relevant effects of interactions, consistent with the cause-effect relationships perceived by her/him. However, due to some unpredictable interactions, the effects may not be achieved. Some of the effects may be very complex, for example, the interactions may lead to the construction of new c-granules and/or other complex hunks. For these activities, the agent's control may use:

   i. processes in environments, leading to the self-organization for aggregating hunk configuration which interact with each other, and
   ii. other complex networks of interconnected cause-effect relationships.

32.3.5 FPW-05: Predictability and M-hunks

1. The *agent's control* may perceive (to a *satisfactory degree*) satisfiability of some causes of the cause-effect relationships in interactions perceived by it. Next, an agent may try to predict some effects of these interactions. This ability is called the *agent's predictability*.
2. An agent can only partially predict some effects of interactions. More detailed effects of interactions can be perceived by an agent only when they are completed.
3. Some configurations of hunks from the agent's memory and/or configurations of hunks from the agent's control can be perceived by the agent's control.
   These configurations represent physical beings, and/or the results of interactions between them, in the agent's memory. The physical beings are called the *agent's m-hunks* (or *m-hunks*, in short).
4. An agent can use m-hunks to represent some effects of interactions and use them in the process of perceiving. In particular, m-hunks can represent the effects of perceiving the following:

   i. Hunks.
   ii. Interactions.
   iii. Configuration of hunks and interactions, which are called *situations*.
   iv. *Contexts of interactions*.
   v. *Constraints* related to all *perceivable phenomena* (e.g., physical beings, interactions, situations, and contexts).

32.3.6 FPW-06: Interactive Computations, Perceiving Attributes, and Hunk Properties

1. *Interactive computations* are performed due to changes in the configurations of hunks. These changes progress through the *interactions* on a given set of hunks.
2. If some special constraints (called *cause-constraints*), related to the *context of interactions* that support computations, are satisfied (to a satisfactory degree), then the agent's control may expect that the computations satisfy some other constraints (called *effect-constraints*).
3. The effect-constraints can be used to specify the *halting conditions of computations* and final effects of computations, which are called the *results of computations*. Some distinguished results of computations are called the *values of computations*. Note that the actual effects may be different from what was expected due to unpredictable interactions with the environment.
4. An *agent's perceiving attribute* [382, 384, 463, 466, 469] (or an *attribute*, in short) of hunks and/or their configuration (and/or relations between hunks) **is a c-granule**, **implementing** the ability of the agent's control to initiate/perceive/modify the following interactions:

   i. Apply the *attribute control parameters* to the *interactive computations* of the *attribute values* over the *attribute object* (indicated by the attribute control parameters).
   ii. Specify an *attribute object* and its further verification/acceptance by the agent's control, using the attribute control parameters. The attribute object can be: a relevant hunk, interaction, and/or configuration (of hunks, interactions, and/or other configuration).
   The attribute object is connected, via special links, with other hunks that encode interactions between the object and other hunks, as well as other interactions relevant to the states of the attribute object. These states are called the *values of the attribute object*.
   iii. Localize, recognize, and/or acquire of the attribute objects and/or other hunks necessary to perform a computation of the attribute value.
   iv. Compute, update, store, and/or retrieve the current *value of the attribute*, called also the *hunk property relative to the attribute* or, shortly, *hunk property*.

5. **Any perceiving attribute is implemented by a c-granule, using parts of an agent and/or other tools**.
6. For any attribute A_G and any two different hunks H_1 and H_2 it is possible that the corresponding *values of the attribute* (*hunk properties relative to the attribute*) *are equal* for the agent's control. Certainly, in a more formal setting one should consider that the computation of attribute values is performed in the agent time.

For any attribute A_G and its value v one may consider the family of all hunks on which the perception leads to this value. One may treat this family as the semantics of the descriptor $A_G = v$. Usually, this set is known only partially, for a sample.

7. Let $\mathscr{I}$ be a set of perceiving attributes. We may say that two hunks, H_1 and H_2, are *I-indiscernible by an agent*, provided that for any attribute from the set $\mathscr{I}$, the agent's control is not able to distinguish the two beings by the computed values of the perceiving attributes [382, 384, 463]. If H_1 and H_2 are *I-indiscernible by an agent*, then we may also say that H_1 and H_2 have the same hunk properties relative to $\mathscr{I}$.
8. The value of a perceiving attribute can be unknown to the agent's control. What is more, it can be unavailable to the agent's control.
9. Due to unexpected interactions, the computation of the value for the perceiving attribute can be disrupted. Sometimes, it is impossible to correctly compute the value of the perceiving attribute.
10. Due to the *imperfectness of perceivability* (point 6 in FPW-4 Sect. 32.3.4), **only some perceiving attributes can be available** to an agent and perceived by the agent's control.
11. In order to illustrate the concept of a c-granule, let us consider an example of a c-granule, representing a complex sensory perceiving attribute [466, 469].

    This attribute may be implemented in the physical world by a device, e.g., a TV set, cameras, and/or other measurement tools which are in relevant interactions with the physical world.

    As a result, one may understand a c-granule as a complex perceiving attribute, e.g., implemented in the physical world, using as a sensor a complex device such as a TV set (which is "observed" and interactively controlled by the agent's control) connected to a camera. A TV set is linked to cameras and/or other measurement tools.

    The example provided above is used to explain a very important functionality of c-granules, which enables them to interact with the reality which is out of the agent's reach.

    This example is also an illustration of the concept of the *agent's window* (cf. point 2 of FPW-04 Sect. 32.3.4). For example, with the use of a screen and a remote control of a TV set, an agent is able to perceive and register the effects of interactions in the physical world.

    In more general cases, an agent, by using her/his actuators, can intervene in the ongoing inter-actions perceived through the agent's window. Usually, the agent's control can interactively initiate complex interactions in the agent's environment (e.g., using a remotely controlled drone).

32.3.7 FPW-07: Interaction Properties and Interaction Scene

1. The agent's control can perceive some relations between hunks. For example, it can recognize that one hunk H_1 is a *part of* another hunk H_2 (based on the process of perceiving by an agent at given moment of agent's time). In such a situation, we may write that $H_1 < H_2$.
   If $\mathscr{F}_H$ is a family of parts of the hunk H, then $\langle \mathscr{F}_H, < \rangle$, along with the relation of being a *part* "$<$", is called a *mereological structure* of H.
2. Let us assume that $\langle \mathscr{F}_H, < \rangle$ is a mereological structure of a hunk H. We may say that P is an *interaction attribute*, provided that an agent perceives $\langle \mathscr{F}_H, < \rangle$ by P, based on the application of the *interaction attribute control parameters* of P to the *interactive computations* of the *interaction attribute values* of P over $\langle \mathscr{F}_H, < \rangle$ (indicated by the attribute control parameters of P), through the aggregation of the following attributes:

   IP-a. *Attributes inherent to the mereological structure* $\langle \mathscr{F}_H, < \rangle$ of H (i.e., an agent's perception of being a part of H components, or part of the parts, etc.).
   IP-b. *Attributes inherent to the parts* of H at the level of its mereological structure (including the properties of parts that are parts of its components, etc.).
   IP-c. *Attributes of relations between the sets of H parts.*
   IP-d. *Attributes of H being the results of aggregating the properties of H parts.*

3. Analogous to the *agent's perceiving attribute*, the *interaction attribute* also satisfies the following conditions:

   i. The *attribute control parameters* are applied to the *interactive computations* of the *attribute values* over the *attribute object* (indicated by the attribute control parameters).
   ii. The agent's control, using the control parameters of the interaction-attribute specifies the *mereological structure* and its further verification/acceptance. The mereological structure encodes interactions among the objects through a special linking to some other hunks and by relevant states of the attribute object. These states are called the *values of the interaction*.
   iii. Localization, recognition, and/or acquisition of the *mereological structure* and/or other hunks, necessary to perform computation of the interaction attribute value.
   iv. Compute, update, store, and/or retrieve the current *value of the interaction attribute*, called also the *interaction property relative to the interaction attribute* or, shortly, the *interaction property*.

4. To put it more simply, the *interaction property* P of a mereological structure of a hunk H **are formed in the process of aggregating the properties of parts of a given hunk and the relations between these parts. The interactions themselves, however, constitute all the processes of the physical objects, leading to the creation of interaction properties**.
   Such an interpretation of an interaction property (or the interaction alone) may be of a twofold nature: *static* (e.g., a ship staying afloat) and dynamic (e.g., a rock falling to the Earth).
5. If P is the *interaction property* of a mereological structure of the hunk H, then the hunk H is commonly referred to as the *interaction scene* (or *scene*, in short), whereas the hunks which constitute its parts, appearing under the aforesaid conditions (IP-a–IP-d), are known as the *participants of interaction* or, shortly, the *interaction hunks* (i.e., hunks which are in an interaction).
6. Some examples of interaction hunks include: planetary motion around the Sun, electron motion around the nucleus of an atom, car traffic, objects lying on a table, building constructions, as well as theatrical performances.
7. When H is an interaction scene, at various moments of time, its parts that are in an interaction in the interaction scene, determined by this hunk, may be found in states that distinguishable by an agent, based on its perceiving capability. In such a situation, we may say that an agent perceives various *hunk configurations* of H.
8. *Interaction scenes* are open for other interactions in the physical world. The interactions establish some relationships between interactions scenes. For example, if H_1 and H_2 are two interactions scenes, then we can consider the following very intuitive relationships between H_1 and H_2:

   i. H_1 *is a part* of H_2.
   ii. H_1 *follows* H_2, and respectively *followed, will follow, has followed, have followed, is following, was following, will be following, etc.*.
   iii. H_1 constitutes a *context of* H_2 provided that some interactions on H_1 may have an impact on interactions on H_2.

9. For any important (for an agent) interaction scene H, she/he can assign a c-granule G_H, which is an aggregation of all c-granules, relevant to the scene H (to a certain degree of importance of the relationships). The aggregated c-granule G_H is called the *agent's interpretation of the scene H* or, shortly, the *scene interpretation.*
10. For any attribute P_G and any two different interaction scenes H_1 and H_2, it is possible that the corresponding *values of the interaction attribute* (*hunk properties relative to the interaction attribute*) *are equal* for the agent's control.
11. Let $\mathscr{I}$ be a set of perceiving attributes and let $\mathscr{S}$ be a family of interaction properties for two interaction scenes based on two hunks, H_1 and H_2. We say that the two interaction scenes, H_1 and H_2, are $\langle \mathscr{I}, \mathscr{S} \rangle$*-indiscernible by an agent*, provided that:

i. The two hunks, H_1 and H_2, have the same hunk properties relative to $\mathscr{I}$, that is the hunks H_1 and H_2 have the same values of the attributes from $\mathscr{I}$.
ii. The two hunks, H_1 and H_2, have the same interaction properties relative to S, that is the hunks H_1 and H_2 have the same values of the interactive attributes from $\mathscr{S}$.

12. By an $\langle \mathscr{I}, \mathscr{S} \rangle$-*specification* of interactions scenes, we understand any selection of the values of attributes from $\mathscr{I}$ and any selection of the values of interactive attributes from $\mathscr{I}$. Any $\langle \mathscr{I}, \mathscr{S} \rangle$-*specification* leads to a class of interaction scenes, which satisfy (to a certain degree of satisfiability—established by an agent) the values from the specification.

32.3.8 FPW-08: Interaction Plans

1. The *agent's interaction model* is a c-granule, obtained by an aggregation of c-granules, which represents:

   i. *Configurations of hunks and/or interactions* with the physical world, as well as cause-effect relationships among them, as observed, simulated, and predicted by c-granules.
   ii. Some constraints over interactions perceived by an agent, using c-granules. C-granules point to the structure of the different parts of the physical world, perceived by an agent, and enable her/him to (partially) perceive interactions between these parts. The interactions may be activated by the agent's activity in a given environment and/or by the agent's control in the environment of the agent's activity.

   The agent's interaction model can be constructed and/or imposed on by an $\langle \mathscr{I}, \mathscr{S} \rangle$-*specification* (Sect. 32.3.7 FPW-07 point 12).
2. An *agent's interaction plan* (or *interaction plan*, in short) is a c-granule which is an aggregation of c-granules representing: interaction models, conditions for initializing or completing interactions, constraints to the achievement and specification of the expected results. The agent's interaction plan covers the following specifications:

   i. Agent's strategies of exploring the agent's environment by the agent's activity, carried out in accordance with the rules for the selection and initiation of interactions in different (potentiality expected) parts of the physical world.
   ii. *Requirements for the implementation of an interaction plan* being a group of conditions that belong to one of the following families of conditions:
      a. Expected constraints on the final configuration of hunks after a correct implementation of an *interaction plan*.

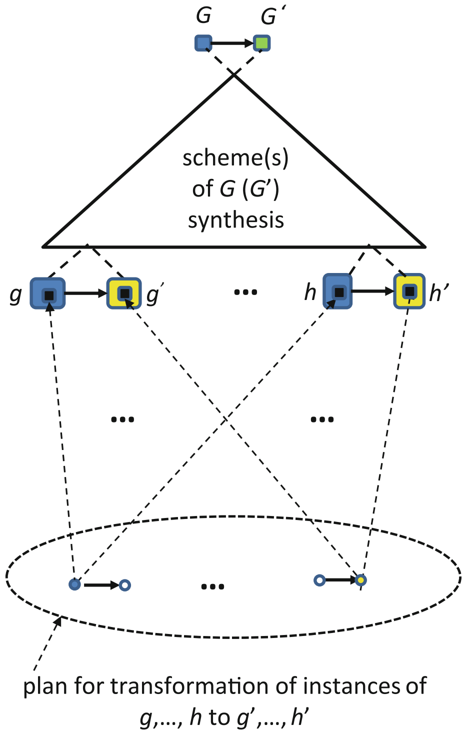


Fig. 32.2 A plan for the transformation of hunk configuration $g, \ldots, h$ to hunk configuration $h', \ldots, g'$ and the concept of its implementation

b. Constraints on the whole trajectory of a configuration, resulting from the implementation of an *interaction plan* in the physical world.
c. Conditions specific to a given *interaction plan*, indicating the necessity to initiate some actions during specified phases of its implementation.

iii. Rules for the generation, activation, adaptation, and deactivation of appropriate actions, applied according to the *requirements for the implementation of an interaction plan* (cf. above ii.).

3. In Fig. 32.2, we present an illustration of an "interaction plan" at the current state of the world, represented by the c-granule G (e.g., representing a complex vague concept and a degree to which this concept is currently satisfied) and the requirement that have to be fulfilled in order to change the current situation to a situation represented by the c-granule G' (e.g., representing the same concept as in G and the desired degree of satisfiability). The c-granules G and G' illustrate the c-granule synthesis, achieved by transmitting the effects of interactions (by sensors and actuators).
The degree of satisfiability of the concept in G is obtained as a result of perceiving the hunk configuration $g, \ldots, h$. The aim of the plan is to perform a sequence of actions to transform the hunk configuration $g, \ldots, h$ into the

hunk configuration $g', \ldots, h'$, **in order to achieve the desired degree of satisfiability of the concept represented in** G'.
During the implementation of an interaction plan, the c-granule G may recognize some new constraints and/or changes. Hence, it has to produce a new c-granule G', which is adapted to the new constraints.
4. An agent uses interaction plans for the construction of interaction plans and for their implementation, adaptation, and judgment, as well as to learn how to effectively discover the constructions of new c-granules.
5. An agent can construct interaction plans, using the attributes of:

   i. agent's control configuration,
   ii. relations between the agent's control configuration, and
   iii. other interactions causing changes in the agent's control configuration.

32.3.9 FPW-09: Agent's Perception

1. An *agent's perception* of the physical world is the agent's interactive ability to comprehend the perceived parts of the physical world, accessible by interactions through the agent's windows, implemented by c-granules based on the following tasks:

   i. Construction, verification, development, adaptation, and/or verification of the *agent's spatiotemporal maps* and/or *agent's interaction models* which is achieved by the following actions:
      a. Comparing the actual effects of the agent's interactions and the expected effects, on the basis of the *agent's spatiotemporal maps* and/or *agent's interaction models*.
      b. Recording and representing the actual effects of the agent's interactions in her/his memory.
      c. Constructing and interpreting the expected effects, on the basis of the *agent's spatiotemporal maps* and/or agent's interaction models.
   ii. Relevant construction, implementation, verification, and adaptation of the *agent's interaction plans*.

2. During the process of perception, the agent's control uses c-granules in a number of different ways. Especially, the following forms of using c-granules are important for the *agent's perception*:

   i. Initialization, modification, and/or termination of interactions in the agent's environment.
   ii. Computations, storage (registering), and/or retrieval of perceiving attributes.
   iii. *Granulations*, by which we mean the interactions which represent some type of c-granule processing. We may differentiate between the following types of granulations:

a. Construction, aggregation, decomposition, and destruction of c-granules
b. Storage, retrieval, and reconstruction of c-granules.

effectiveness and efficiency

32.4 Need Satisfiability, Effectiveness, and Efficiency of Interaction Plans

32.4.1 FPW-10: Adaptive Judgment Relative to the Agent's Hierarchy of Needs

FPW-10: Abstract

A. The agent's control can use some special c-granules for expressing her/his *hierarchy of values*. Such a c-granule is called a *judgment scale*.
For example, as a judgment scale, the agent's control may use partially ordered sets (*posets* [441]) over particular subsets of c-granules, which are called the *degrees of the judgment scale*.
Any finite poset can be visualized through its Hasse diagram [441]. Figure 32.3 illustrates an example of the Hasse diagram of the judgment scale and particular subsets of degrees.
The *judgment scale* can be used by the agent's control in many contexts. For example, it can be used to represent such concepts as: judgment results, hierarchy of concepts, or satisfiability degrees of a statement expressing some conditions, properties, and/or constraints.
In general, a *degree of judgment scale* can be a more complex structure than just a mere element of the judgment scale represented by a poset. Note that elements of the judgment scale are c-granules.
One may consider some structures of elements in the judgment scale as degrees of the judgment results. They can express some interpretations of the degree.
The simplest case of such a structure is the representation of the current knowledge about the approximation of an unknown value of the judgment result by intervals in posets. For example, the degree of the judgment result can by simply an ordered pair $< d_1, d_2 >$, where d_1 is a lower approximation of the result of the current judgment and d_2 is a current, upper approximation of the judgment result. Depending on some interactions (experiments), the judgment result can be approximated in a more (or less) precise manner.
The selection of particular implementation structures (in the form of c-granules) for such a representation of degrees depends on the specificity of particular applications. However, to simplify, in further examples, we will use the elements of the judgment scale as the degrees of the judgment scale.

B. The judgment scale depends on the context of interactions. For example, some unacceptable actions of an agent may be treated as acceptable in some special contexts of interactions.

C. An agent requires such resources as a "healthy physical body," energy, materials, and/or tools in order to function in a given environment and carry out interaction plans.

   By using c-granules, the agent's control can perceive some aggregations of attribute values related to the physical agent's control and the agent's environment.

D. By an *agent's need* (or *need*, in short), we mean the aggregations of the following c-granule collections:

   i. A collection of c-granules called the *need's criteria*. Any *need criterion* is used to compute the following values (which are c-granules themselves):

      a. *Arguments of the need's criterion*, on the basis of which the agent's control makes a decision, whether the need criterion is satisfied.

         The *arguments of the need's criterion* are classified by the agent's control (or a c-granule designated by the agent's control) as one of the following types of arguments:

         [+] —*for*, if the satisfiability degree of an argument is positive,
         [–] —*against*, if the satisfiability degree of an argument is negative,
         [0] —*neutral*, if the satisfiability degree of an argument is neutral.

         Any changes in the agent's environment may lead to changes in the results of the evaluation and aggregation of the above-mentioned arguments: *for, against*, and *neutral*.

         Any *argument of need's criterion* is evaluated in *terms of its satisfiability degree*, which is an element included in the *judgment scale of the need's criterion* argument (cf. Fig. 32.3).

      b. The *satisfiability degree of the need's criterion*, which results from aggregating all *arguments of the need's criterion*, is an element included in the judgment scale of the need's criterion (cf. Fig. 32.3).

   ii. *Satisfiability degree of a need*, which results from aggregating all need's criteria. The *satisfiability degree of a need* is an element included in the *judgment scale of a need* (cf. Fig. 32.3).

   iii. A collection of c-granules, which provides to the agent's control with some methods for the construction of interactions plans, leads to an increase in the satisfiability degree of a need. The methods, in particular, depend on:

      a. the specificity of a need,
      b. the current situation in the environment, especially of its components, specified by the *characteristics of the environment*, which help in increasing the satisfiability degree of a given need, and
      c. resources available to the agent's control.

E. As examples of an agent's need, one may consider the following needs:

   i. *Need for an acquisition of necessary resources for the agent's activities*.
   ii. *Need for a construction and/or evaluation of possible solutions to problems* that have been identified (e.g., problems related to the construction of an interaction plan, which helps to satisfy another need).
   Typical examples of such problems, are:

      a. *Classification problems*, e.g., to which decision class, a given object belongs.
      b. *Optimization problems*, e.g., a construction of an optimal interaction plan that would fulfill the given constraints.

   The need for the construction and/or evaluation of a problem's solution can be used to generate, actualize, and aggregate the constructions of various alternative solutions to a given problem. In the construction of a solution, the *agent's context of interactions* is taken into account.
   iii. *Need for a verification of conditions, necessary for the initialization of an interaction plan*.
   iv. *Need for a verification of the results, expected by an agent after the implementation of a given interaction plan* to see whether they are satisfactory enough. Otherwise (that is when the results after the implementation of an interaction plan do not match the expectations), a judgment is made that leads to a decision on which interaction plan should be implemented next to develop and initiate actions responsible for the process of adaptation.

F. The agent's control has an ability to perceive a c-granule called the *agent's hierarchy of needs*. This c-granule is an aggregation of the c-granules of needs (representing the properties of needs perceived by the agent's control), c- granules that represent the (adaptive) relations between them, and adaptive judgments over the actualization of the components in the hierarchy of the agent's needs.
The *agent's hierarchy of needs* consists, in particular, of the following c-granules:

   i. A judgment scale, called the *judgment scale of the hierarchy of needs*.
   ii. A *need's importance judgment*, which is a c-granule that computes the importance of the agent's need for any moment of the agent's time t (especially important is the judgment of the need's importance of the agent, relative to the importance of other agent's needs at a given t).
   The current result of computations of *need's importance judgment* is an element included in the *judgment scale of the hierarchy of needs*.
   iii. The agent's control has skills for identifying which of the needs from the agent's hierarchy of needs are currently especially important. Such needs are aggregated into a c-granule, called the *current agent's prioritized needs*.

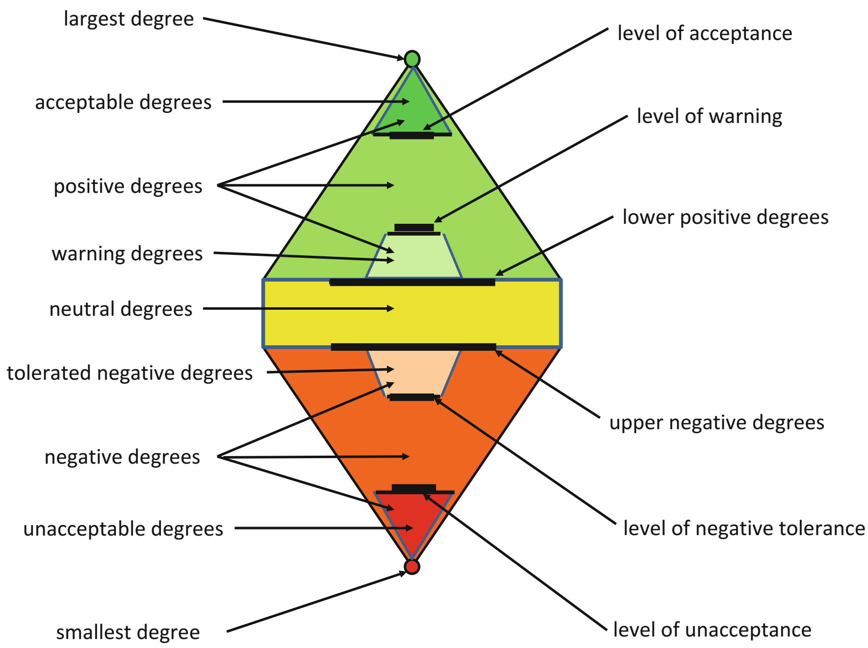


Fig. 32.3 Judgment scale

The agent's hierarchy of needs is a metaphor of the well-known concept of the *Maslov's hierarchy of needs*.[3] In this hierarchy, there are needs for assuring good social relations, good health, material resources, or broadly-understood security issues.

FPW-10: Exemplary Refinement

1. Let us assume that S and S' are judgment scales based on posets (cf. Hasse diagram in Fig. 32.3) and let g_Δ be a c-granule computing a function $\Delta : S \times S \longrightarrow S'$.
   We say that Δ is a *difference operator over the judgment scale* S (or *difference*, in short), provided that for any $d_1, d'_1, d_2, d'_2 \in S$, the following conditions are satisfied:
   i. if $d_1 \leqslant d_2$, then $\Delta(d_1, d_2)$ is neutral or positive in S',
   ii. if $d_2 \leqslant d_1$, then $\Delta(d_1, d_2)$ is neutral or negative in S',
   iii. if d_1 is a neutral degree in S, then $\Delta(d_1, d_2)$ is:
      a. positive in S', if d_2 is positive in S,
      b. neutral in S', if d_2 is neutral in S,
      c. negative in S', if d_2 is negative in S,

[3] http://en.wikipedia.org/wiki/Abraham_Maslow.

iv. if d_1 is a negative degree in S and d_2 is a neutral or positive degree in S, then $\Delta(d_1, d_2)$ is a positive degree in S',
v. if d_1 is a positive degree in S, and d_2 is a neutral or negative degree in S, then $\Delta(d_1, d_2)$ is a negative degree in S',
vi. if $d_1' \leqslant d_1$ and $d_2 \leqslant d_2'$, then $\Delta(d_1, d_2) \leqslant \Delta(d_1', d_2')$.

2. i. Let S be the judgment scale of a need N and let Δ be the difference operator over the judgment scale S.
ii. Moreover, let us assume that for any moment of the agent's time t in a given time interval T and for any need N, the agent ag has an ability to perceive or predict (if t is a moment of time in the future) the satisfiability degree of the need N at t, denoted by $d_t(N)$ (or d_t, for short).
iii. Let T be the agent's time interval and let $t, t' \in T$.
iv. Then, by $\delta_{t,t'}(N)$, we denote the value $\Delta(d_t(N), d_{t'}(N))$ (or, in short, $\Delta(d_t, d_{t'})$).
v. The satisfiability degree $d_t(N)$ is called the *initial degree of the change* $\delta_{t,t'}(N)$, and analogously the satisfiability degree $d_{t'}(N)$ is called the *resulting degree of the change* $\delta_{t,t'}(N)$.
vi. If S is a judgment scale over a poset presented in Fig. 32.3 and $\mathscr{A}$ is a property of an element included in the poset S, then for any family of degrees $\{d_l\}_{l \in L}$ of the judgment scale, we can define the properties of the family, using the properties of the degrees of—the family. For example:
   a. We say that the family $\{d_l\}_{l \in L}$ has a *property* $\mathscr{A}$ *locally*, provided that there exists at least one $l \in L$ such that d_l has the property $\mathscr{A}$.
   b. We may say that the family $\{d_l\}_{l \in L}$ has a *property* $\mathscr{A}$ *globally*, provided that for any $l \in L$, d_l has the property $\mathscr{A}$.
   c. Typical examples of a property $\mathscr{A}$ for the judgment scale in Fig. 32.3 include: *positive change*, *negative change*, *acceptable change*, *unacceptable change*, *warning change*, *tolerated negative change*, *neutral change*.

vii. If S and S' are judgment scales over posets presented in Fig. 32.3, then we can define a *type* of the change degree $\delta_{t,t'}(N)$, using the types of degrees of the judgment scales S and/or S'. As an example let us consider the following instance.
The degree $\delta_{t,t'}(N)$ has one of types of degrees of the judgment scale, listed in Fig. 32.3. Depending on the type, we may obtain the following types of the degree change $\delta_{t,t'}(N)$: *positive change*, *negative change*, *acceptable change*, *unacceptable change*, *warning change*, *tolerated negative change*, *neutral change*.

3. $\mathscr{C}_T$ is a set of *interaction constraints over the agent's time interval* T (or, in short, *constraints* for T), provided that it is a family of conditions which an agent aims to satisfy for all interactions perceived by her/him over the interval T.

i. $\delta_{t,t'}(N)$ (where $t, t' \in T$) is an *inadmissible change* provided that the change satisfies at least one of the following conditions:
   a. it is an unacceptable change,
   b. the resulting degree of the change is none of 'tolerated negative degree', 'positive degree', 'neutral degree', and
   c. the change causes the unsatisfiability of some interaction constraints from $\mathscr{C}_T$ over the interval $[t, t']$.

ii. $\delta_{t,t'}(N)$ is a *desired change*, provided that the change satisfies all of the following conditions:
   a. it is an acceptable change,
   b. the change does not cause the unsatisfiability of some interaction constraints from $\mathscr{C}_T$ over the interval $[t, t']$.

32.4.2 FPW-11: Cost/Benefit Analysis (CBA) for Interaction Plans

FPW-11: Abstract

A. An agent develops, implements, and adopts interaction plans (represented by c-granules) aimed at increasing the satisfiability degrees of her/his needs. This especially concerns the most important needs, according to the current hierarchy of prioritized needs.
B. In general, the implementation of interaction plans may generate *costs*. The costs are related with decreasing the satisfiability degrees of the needs. An agent expects some *benefits* after the implementation of an interaction plan. The benefits are related with increasing the satisfiability degrees of the needs.
C. An agent has some skills to judge (evaluate) the costs and benefits of interaction plans. The process of judgment enables an agent to construct those interaction plans that are mostly needed.
D. An agent has also some skills for judging the process of judgment related with the selection and/or construction of interaction plans, called *meta-judgments*. For an adaptation of plans, these skills may use, e.g., the results of the comparison between the expected and actual results of actions.
E. We assume that in order to achieve these goals, an agent has c-granules for computing the estimated costs and benefits, as well as the aggregation of costs and benefits (for their comparison).
F. *Benefit Cost Ratio* (BCR) *of an interaction plan*, which is the aggregation of all the benefits and costs of an interaction plan. An agent has a need to increase BCR, using her/his interaction plans. For BCR maximization need, we assume that the judgment scales of satisfiability degrees of needs are linear.
G. We may say that *interaction plans are BCR-comparable* as long as their need for the BCR maximization is characterized by the same judgment scale. In this situation, they can easily become the subject of a cost benefit analysis.

H. The implementation of this approach should make it possible for agents to compare different variants of interaction plans in a framework related to the well-known approach adopted from economy, called the Cost/Benefit Analysis (CBA).[4]

FPW-11: Exemplary Refinement

Let us assume the following notations:

1. N is used to denote a need of an agent ag.
2. For any moment of the agent's time t, in a given time interval T, an agent ag has a potential ability to perceive, predict, and/or estimate an approximation of the value d_t, which is the satisfiability degree of the need N at the time t. The value d_t can be precisely computed at the time t, only when t is a moment of time in the past. Otherwise, the value d_t can be only predicted or estimated.
3. $\mathscr{C}$ is a set of *interaction constraints for the agent's time interval* T.
4. An agent uses some succeeding moments of the agent's time: $t_1, t_2, \ldots, t_n \in T = [t_1, t_n]$, to measure the satisfiability degree d_i of the need N, at a given moment of time t_i.
   The collection of intervals $\{[t_i, t_{i+1}]\}_{i=1,2,\ldots,n-1}$ or, in short, $\{T_i\}$ (where T is the interval $[t_i, t_{i+1}]$) is called a *perceivable partition of the need N* (or a *perceivable partition*, in short) over the agent's time interval $T = [t_1, t_n]$.
5. Let us assume that S is the judgment scale of the need N and let Δ be the difference operator over the judgment scale S, as it is defined in Sect. 32.3.5.
6. The family of values $\{d_i(N)\}_{i=1,\ldots,n}$ is called the *family of satisfiability degrees of need over the perceivable partition* or, in short, the *family of satisfiability degrees of need*. Let us recall that $d_i(N)$ is the satisfiability degree of the need N at a given moment of the agent's time t_i.
   Notice that any element of the family $\{d_i(N)\}$, belongs to the judgment scale S. Thus, any element of the family $\{d_i(N)\}$ can be *positive*, *neutral*, *negative*, or *constitute* another type of elements from S (cf. Fig. 32.3).
7. For any $i \in \{1, \ldots, n-1\}$, let $\delta_i(N) = \delta_{t_i,t_{i+1}}(N)$. The family of values $\{\delta_i(N)\}_{i\in\{1,\ldots,n-1\}}$ is called the *family of need's changes over the perceivable partition* $\{T_i\}$ or, in short, the *family of need's changes* $\{\delta_i\}$.
   Notice that any element of the family $\{\delta_i(N)\}$ is an element of the judgment scale S'. Thus, any element of $\{\delta_i(N)\}$ can be *positive*, *neutral*, *negative*, or can *constitute* another type of elements from S' (cf. Fig. 32.3).
8. If $\{\delta_i\}_{i\in I}$ is a family of changes of need, with elements from the judgment scale S' presented in Fig. 32.3, and $\mathscr{A}$ is a property of elements of the poset S', then one can define the properties of the family $\{\delta_i\}$, using the properties of its degrees. For example:

[4] http://en.wikipedia.org/wiki/Cost-benefit_analysis.

a. We may say that the family $\{\delta_i\}$ has *possibly the property* $\mathscr{A}$, provided that there exists at least one $i \in I$ for which d_i has the property $\mathscr{A}$.
b. We may say that the family $\{\delta_i\}$ *has certainly the property* $\mathscr{A}$, provided that for any $i \in I$, d_i has the property $\mathscr{A}$.

9. Let IP be the agent's interaction plan, implemented in the scope of the agent's time interval T. If an agent implements the interaction plan at a given moment of time $t \in T$, then the following can happen:

   a. An agent can perceive situations (scenes) related to the moments of time before t, with a possibility of including t.
   b. An agent can try to predict some properties of the future interactions. However, this is only a prediction, which may not be true. For example, an agent cannot be sure even about the duration of the implementation of the interaction plan.
   c. Hence, the time interval T and any of its partition $\{T_i\}$ can be subject to change (i.e., the duration of the interaction plan can be longer or shorter).

10. We may assume that $\{N^k\}_{k \in K}$ is a family of all *important needs of an agent over the interval* T, *relative to an agent's interaction plan* IP. For any $k \in K$ and the need N^k, let us denote by $\{T_i^k\}_{i \in I_k}$ a perceivable time partition for the need N^k, such that for any $i \in I_k$, $T_i^k \subseteq T$.

    a. We may assume that S^k is the judgment scale of the need N^k, and $\Delta^k : S^k \times S^k \longrightarrow S'^k$ is the difference operator over the judgment scale S'^k, as it is defined in Sect. 32.3.5.
    b. For any $k \in K$, the family of changes of needs over the need N^k is denoted by $\{\delta_i^k(N^k)\}_{i \in I_k}$. By $\{\delta_i^k(N^k)\}$, we denote the family $\{\delta_i^k(N^k)\}_{k \in K, i \in I_k}$ of all changes.

Using the above notations, one can propose definitions of such basic concepts of agents as cost, benefit, risk, co-risk, efficiency, and many others.

1. **The need of cost cutting in the fulfillment of the need** N.

   i. The *cost of the need* N is measured using the judgment scale S', defined for values of the difference operator related to N and a given time interval.
   ii. By a *cost of the need* N over a perceivable partition $\{T_i\}$, we mean an aggregation of values of all negative need's changes over this perceivable partition.
   iii. The cost of the need N over the perceivable partition is an *inadmissible cost* if in the family $\{\delta_i(N)\}$, there exists at least one inadmissible change (i.e., the family $\{\delta_i(N)\}$ has possibly an inadmissible change property).
   iv. An agent ag may have another potential need N_{CC}, associated to the need N and related to a given perceivable partition $\{T_i\}$, which is called the *need of costs' cutting for the need* N.
   v. The aim of N_{CC} is to achieve the following *Min-Min objectives* of N_{CC}:

$1a_{Min}$. To eliminate or, if it is impossible, at least minimize the value of the aggregation of all current and/or predicted inadmissible changes of the need N in the family $\{\delta_i(N)\}$ (in terms of the judgment scale S').

$1b_{Min}$. If the minimization described above (point $1a_{Min}$) is achieved, then the aim of the need N_{CC} is to minimize the aggregation of the negative changes in the family $\{\delta_i(N)\}$ (provided that the minimal value obtained in point 1_{Min} is preserved).

2. **The need of total cost cutting in the implementation of the interaction plan IP**.

   i. The *total cost of the interaction plan* IP can be measured over the judgment scale, obtained by aggregating the family of judgment scales $\{S'^k\}$, where for each $k \in K$, the judgment scale S'^k is the judgment scale for the values of the difference operator of N^k over a given time interval.
   ii. By a *total cost of the interaction plan* IP, we mean an aggregation of all costs for the family $\{N^k\}$ of needs, consisting of all needs that are important for IP.
   iii. An agent ag may associate another potential need IP_{CC} with the interaction plan IP, called the *need of total costs' cutting for the interaction plan.*
   iv. The aim of IP_{CC} is to achieve the following *Min-Min objectives* of IP_{CC}:

$1a_{Min}$. To eliminate or, if it is impossible, at least minimize the value of the aggregation of all current and/or predicted inadmissible changes in the family $\{\delta_i^k(N^k)\}$ (in terms of the judgment scales S'^k) for all needs from the family $\{N^k\}$.

$1b_{Min}$. If the minimization described above (point 1_{Min}) is achieved, then the aim of the need IP_{BI} is to minimize the aggregation of the negative changes in the family $\{\delta_i^k(N^k)\}$, (provided that the minimal value obtained in point 1_{Min} is preserved).

3. **The need of a benefit increase in the fulfillment of the need** N.

   i. The *benefit of the need* N is measured using the judgment scale S' for the values of the difference operator related to N over a given time interval.
   ii. By a *benefit of the need* N over a perceivable partition $\{T_i\}$, we mean an aggregation of values of all positive need's changes over this perceivable partition.
   iii. The benefit of the need N over the perceivable partition is a *desired benefit*, if any element of the family $\{\delta_i(N)\}$ is a desired change (i.e., the family $\{\delta_i(N)\}$ has certainly the desired change property).
   iv. An agent ag may associate another need N_{BI} with N and a given perceivable partition $\{T_i\}$, which is called the *need of benefit increasing for the need* N.
   v. The aim of N_{BI} is to achieve the following *Min-Max objectives* of N_{BI}:

1_{MM}. The *first Min-Max objectives* of N_{BI}.

1_{Min}. To eliminate or, if it is impossible, at least minimize the value of the aggregation of all current and/or predicted inadmissible changes of the need N in the family $\{\delta_i(N)\}$ (in terms of the judgment scale S').

1_{Max}. If the minimization described above (point 1_{Min}) is achieved, then the aim of the need N_{BI} is to maximize the aggregation of the positive changes in the family $\{\delta_i(N)\}$ (provided that the minimal value obtained in the point 1_{Min} is preserved).

2_{MM}. If the *first Min-Max objectives* of N_{BI} described above (points 1_{Min} and 1_{Max}) are achieved, then the next objective of the need IP_{BI} is to achieve the *second Min-Max objectives* of N_{BI}.

2_{Min}. The min-objective of the second Min-Max condition of N_{BI} is to minimize the value of the aggregation of any non-desired changes from the family $\{\delta_i(N)\}$ and next to fulfil the N_{CC} need (provided that the Min-Max values obtained in 1_{Min}, and 1_{Max} are preserved).

2_{Max}. If the objectives 1_{Min}, 1_{Max}, and 2_{Min} are fulfilled, then the next max-objective is to maximize the aggregation of the desired changes in the family $\{\delta_i(N)\}$ (provided that the Min-Max values obtained in 1_{Min}, 1_{Max}, and 2_{Min} are preserved).

4. **The need of a total benefit increase in the implementation of the interaction plan IP**.

   i. The *total benefit of the interaction plan* IP can be measured using the judgment scale, obtained by aggregating the family of judgment scales $\{S'^k\}$, where for each $k \in K$, the judgment scale S'^k is the judgment scale for the difference operator of N^k over a given time interval.

   ii. By a *total benefit of the interaction plan* IP, we mean an aggregation of all benefits for the family $\{N^k\}$ of needs, consisting of all needs that are important for IP.

   iii. An agent ag may have another need IP_{BI}, associated with the interaction plan IP, called the *need of of total benefit increase for the interaction plan* IP.

   v. The aim of IP_{BI} is to achieve the following *Min-Max objectives* of IP_{BI}:

   1_{MM}. The *first Min-Max conditions* of IP_{BI}.

   1_{Min}. To eliminate or, if it is impossible, at least minimize the value of the aggregation of all current and/or predicted inadmissible changes in the family $\{\delta_i^k(N^k)\}$ (in terms of the judgment scales S'^k) for all needs from the family $\{N^k\}$ consisting of all needs that are important for IP.

   1_{Max}. If the minimization described above (point 1_{Min}) is achieved, then the aim of the need IP_{BI} is to maximize the aggregation of the positive changes in the family $\{\delta_i^k(N^k)\}$ (provided that the minimal value obtained in the point 1_{Min} is preserved).

2_{MM}. If the *first Min-Max objectives* of IP_{BI} described above (points 1_{Min} and 1_{Max}) are achieved, then the next aim of the need IP_{BI} is to achieve the *second Min-Max objectives* of IP_{BI}.

2_{Min}. The min-objective of the second Min-Max condition is to eliminate any non-desired changes from the family $\{\delta_i^k(N^k)\}$ (provided that the Min-Max values obtained in 1_{Min}, and 1_{Max} are preserved).

2_{Max}. If the objectives 1_{Min}, 1_{Max}, and 2_{Min} are fulfilled, then the next max-objective is to maximize the aggregation of the desired changes in the family $\{\delta_i^k(N^k)\}$ and next to fulfil the above IP_{CC} need (provided that the Min-Max values obtained in 1_{Min}, 1_{Max}, and 2_{Min} are preserved).

5. We assume that for any moment of the agent's time, the judgment scales of IP_{CC} and IP_{BI} are the same.

32.4.3 FWP-12: SWOT Analysis and Risks of Interaction Plan

FPW-12: Abstract

A. The *implementation of interaction plans may be disrupted*. As a consequence of an interaction plan disruption, the actual properties of implementation of an interaction plan may be different from the expected ones. Disruptions may also occur in the future. Such disruptions are called *potential disruptions*.
B. A *disruption is negative* if at least one of the following conditions is met:

   i. The benefit-cost ratio (BCR cf. FPW-11) for the implementation of a given interaction plan (under this disruption) substantially decreases or
   ii. The total cost of the implementation (under this disruption) does not exceed the level of the negative tolerance (cf. Fig. 32.3).

   If one of the above conditions (i.e., i. or ii.) is met, then we call such a situation a negative consequence of the interaction plan disruption or, in short, a negative consequence.
   In other words, by definition, any *negative disruption* has some *negative consequences* for the interaction plan.
C. A *disruption of an interaction plan* is *positive* if it is not negative and, at the same time, if it substantially increases the benefit-cost ratio for the implementation of a given interaction plan (under this disruption). The substantial increase in the benefit-cost ratio is called a *positive consequence of the interaction plan* disruption or, in short, a *positive consequence*.
D. A positive and/or negative disruption of an interaction plan is an interaction causing agent's perceptible changes in costs, benefits, or BCR of this plan.

E. Any implementation of an interaction plan includes:

   i. *identification of potential positive and/or negative disruptions*,
   ii. *description of potential positive and/or negative disruptions*,
   iii. *evaluation of criteria for the recognition of potential positive and/or negative consequences of disruptions*,
   iv. *evaluation of the likelihood of positive disruptions*,
   v. *evaluation of the likelihood of negative disruptions*,
   vi. *evaluation of the likelihood of positive consequences of a disruption*,
   vii. *evaluation of the likelihood of negative consequences of a disruption*.

F. The *risk of a negative disruption* for a given implementation of an interaction plan is evaluated by aggregating:

   i. the likelihood of a negative disruption,
   ii. all negative consequences of a disruption,
   iii. all likelihoods of the negative consequences of a disruption.

   The larger the likelihood is and/or the greater the negative consequences are, the higher the risk becomes.

G. The *risk of an interaction plan* is an aggregation of all risks of the negative disruptions over the interaction plan.
   The implementation of any interaction plan is related to *risk management*, encompassing the *risk assessment*, *risk treatment, risk communication*, and *improvement of risk management*.

H. In order to comprehend and analyze agent's (or a team of agents) current situation, agents (or a team of agents) use the SWOT[5] analysis [190, 191, 376, 377]. In the explanation of the SWOT analysis we may use the concept of scene as it is defined in FPW-07. However, the SWOT analysis can be performed relative to particular situation of an agent (or groups of agents), and/or relative to any interaction plan. Roughly speaking, by the SWOT analysis of agent (or the team of agents) in a scene (or a situation) or SWOT analysis of an interaction plan, we mean an analysis providing at least the following properties of agents, environment, and/or interaction plan:

   i. Strengths: a property of the agent (or the team of agents and/or interaction plan) causing an advantage or some benefits (i.e., agent could initiate positive disruptions) for the agent (or a team of agents); in particular it may lead to advantage over other agents.
   ii. Weaknesses: a property of the agent (or the team of agents and/or interaction plan)—in the scene (or situation)—causing a disadvantage or some costs (i.e., environment could initiate negative) for the agent (or a team of agents), in particular, it may lead to disadvantage relative to other agents.

[5] https://en.wikipedia.org/wiki/SWOT_analysis.

iii. Opportunities: a property of the environment that the agent (or the team of agents) could exploit (in the current scene (or situation) and/or in the future) to its advantage (i.e., agent could initiate positive disruptions).
iv. Threats: a property of the environment that could cause troubles (i.e., environment could initiate negative disruptions) for the agent (or the team of agents) in the scene (or situation).

FPW-12: Exemplary Refinement

1. An agent is able to construct and actualize a c-granule responsible for the SWOT analysis of an interaction plan that is being developed or implemented.
2. One of the main goals of the SWOT analysis is the analysis and assessment of phenomena, related to positive and/or negative disruptions that influence the implementation of a given interaction plan in a particular project. During the analysis, an agent especially takes into account the following issues:

   i. Negative disruptions of an interaction plan are caused by interactions in which threats make use of the weakness of the interaction plan.
   ii. Positive disruptions of an interaction plan are caused by interactions in which the strengths of the interaction plan make use of the emerging opportunities.

3. Any agent has at least two needs related to the negative disruption of her/his interaction plans:

   i. *To avoid negative consequences and, if that is impossible, to minimize the negative consequences of the negative disruption* during the implementation of an interaction plan.
   ii. *To maintain the likelihood* of negative consequences of negative disruptions as small as possible.

4. Any agent has a need to minimize the risk of a negative disruption during the implementation of an interaction plan, initiated by that agent.
5. Any agent has the need to minimize the risk of an interaction plan, initiated by that agent.

32.4.4 FPW-13: Co-risks and Efficiency of Interaction Plans

FPW-13 Abstract

A. By analogy to how an agent needs to avoid the occurrence of a negative disruption in the implementation of her/his interaction plans, she/he may also use her/his strengths to identify opportunities (especially those that were not predicted before the initiation of an interaction plan) and take advantage of them. In this way, *positive disruptions* of interaction plans can occur. Such a reasoning leads to the concept of a *co-risk*.

B. However, an **agent avoids any positive disruption which could cause a negative disruption, creating a risk and/or cost below the level of a negative tolerance (cf. Fig. 32.3)**. **Having in mind this assumption, by a *co-risk* of a positive disruption, we mean an aggregation of**:

   i. likelihood of a positive disruption,
   ii. all positive consequences of a disruption,
   iii. all likelihoods of positive consequences of a disruption,
   iv. possible significant side effects, in particular, risks and costs caused by a positive disruption.

C. *The co-risk of an interaction plan* is an aggregation of all co-risks of positive disruptions in the implementation of an interaction plan. An agent has a need to maximize the co-risk of a given interaction plan, bearing in mind that she/he should avoid any positive disruption, which could cause a negative disruption, creates a risk and/or cost above the level of the negative tolerance.
D. An agent, by **judging the interaction plans developed and implemented by her/him**, may take into account the *efficiency of an interaction plan*, which is understood as an aggregation of: total cost, total benefit, risk, co-risk (of this interaction plan IP), and the need IP_{BI} (i.e., the need of total benefit increase for the interaction plan IP, cf. FPW-11 point 4.iii).
E. For any interaction plan, an agent has a **need to maximize the efficiency of an interaction plan** implemented by her/him, in accordance with the priorities of her/his needs. An interaction plan is called an *efficient interactive plan*, provided that the need to maximize the efficiency of this interaction plan is satisfied to an acceptable satisfiability degree of the need.
F. For any family Φ of interaction plans perceivable by an agent (e.g., interaction plans that are currently implemented, interaction plans implemented in the past, and/or interaction plans planned for the future), an agent may evaluate the *Φ-efficiency of the family of interaction plan Φ*, referred to as *Φ-efficiency*, in short, which is understood as the aggregation of total cost, total benefit, risk, co-risk, and the need IP_{BI} of all interaction plans from the family Φ.
G. For any family Φ of interaction plans perceivable by an agent, an agent has a **need to maximize the Φ-efficiency, in accordance with the priorities of her/his needs**. The family Φ of interaction plans is called the *Φ-efficient family*, provided that the need to maximize the efficiency of an interaction plan is satisfied to an acceptable degree.
H. **An agent has a need to maximize the efficiency of the family Ω, including all interaction plans that are currently perceivable by her/him**, taking into account the agent's priority needs at a given moment. In such a situation, by the *agent's efficiency*, we mean the *Ω-efficiency for the family of all perceivable interaction plans by an agent*.

FPW-13: Exemplary Refinement

1. An agent has at least the following needs regarding a positive disruption of an interaction plan:

    i. The *likelihood of a positive disruption* is as high as reasonably possible.
    ii. The *consequences of a positive disruption* are relatively large.
    iii. The *likelihoods of positive consequences* are relatively large.
    iv. An agent avoids any *positive disruption* which could cause some negative side effects, such as risks and costs below the level of the negative tolerance (cf. Fig. 32.3).

2. An agent has a need to maximize each co-risk of a positive disruption during the implementation of an interaction plan.
3. The *efficiency* and *Φ-efficiency* of interaction plans **directly depend on the agent's hierarchy of needs and on the judgment scales of these needs**.
   Thus, this type of efficiency is called the *subjective efficiency*. To improve the *objective efficiency*, an agent should learn (particularly from other agents, especially those with a better efficiency) and improve the hierarchy of needs and the judgment scales.
4. **The hierarchy of needs, judgment scales, and thus the efficiency should depend on the properties of interaction scene and on the context of a given scene. For example, an agent may have different hierarchies of needs and judgment scales for** *times of peace* **and** *times of war*.

32.4.5 FPW-14: Agent's Most Important Tasks and Their Execution Based on the Process of Perception of a Perceived Situation and Reasoning About It

FPW-14: Abstract

The most important tasks of an agent include:

A. Identification of all agent's needs, which currently fulfill one of the following conditions:

    1. The need has an unacceptable degree of satisfiability.
    2. The need which can soon have an unacceptable degree of satisfability.

B. Construction, aggregation (integration), and implementation of a reasonably efficient family of interaction plans to improve the satisfiability degree of needs, identified above in point A.
C. Once the two aforesaid conditions, A and B, are matched at a satisfactory degree, the most important tasks of an agent include:

1. Achievement of a *satisfactory degree of comprehension of a perceived situation*.
   By the *comprehension of a perceived situation*, we mean the c-granule of an agent, responsible for controlling the whole process, which leads to the comprehension of a given perceived situation by an agent, especially in the *contexts of interactions*, in the contexts of the current *agent's needs*.
   The c-granule responsible for the *comprehension of a perceived situation* is an aggregation of other c-granules, resulting mainly from the classification of interactions relative to the perceived situation. These c-granules may be related to different constructions by an agent, encompassing such aspects of the perceived situation as contexts of interactions, SWOT, risks, co-risks, prioritized needs, and the priorities regarding the initiation of interactions in a given situation.
   The *satisfactory degree of comprehension of a perceived situation* means, in particular, that the comprehension is sufficient for the *agent's control* to perform:
   i. Proper identification of important interactions in a *perceived situation*.
   ii. Adaptive judgment of a *perceived situation*.
   iii. Identification of top priorities for the *attention of the agent's control*.
   iv. Construction, selection and implementation of an appropriate family of interaction plans to increase the *agent's efficiency* (cf. FPW-13 point H).
2. Maximization of the *agent's efficiency*. It concerns, in particular, the maximization of the agent's efficiency in terms of currently implemented interaction plans in order to satisfy the current priority of needs of an agent.
3. **Development and adaptation of the hierarchy of needs and efficiency of judgment processing**. An agent learns methods (represented as c-granules) for supporting the development and/or improvement of her/his hierarchy of needs and c-granules for adaptive judgment performed by the agent.
4. Development of techniques and skills for:
   i. Improving the perception and comprehension of a perceived situation. **An agent uses these skills to effectively update, improve, and meet her/his needs, as well as to change her/his hierarchy of needs through an effective development, implementation, verification, and improvement of interaction plans, which turn out to be more effective, less risky, and more co-risky**.
   ii. Learning, discovering and/or constructing new c-granules (in particular, new concepts) to improve the efficiency of the adaptive judgment as well as enhance the discovery of knowledge, description and identification of the agent's needs, for efficient fulfillment and/or development of the agent's priority needs.
   iii. Establishing efficient cooperation and/or competition in the society of agents.
   iv. Judging and improving the *agent's efficiency*.

FPW-14: Exemplary Refinement

1. In order to improve the execution of the most important tasks, an agent constructs and uses properly specialized c-granules. The c-granules which enable an agent to interpret the properties of a c-granule reflecting the effects of interactions, taking place in the environment of the agent's activity, are particularly important.
2. An agent has a set of c-granules cooperating with one another; we refer to them as the *adaptive judgment's c-granules*. The c-granules from this set cooperate with c-granules that support the SWOT analysis (cf. Sect. 32.3.7). The results of the analysis are aggregated with the agent's knowledge, particularly when it comes to the context of the agent's present interactions and the environment of her/his activity. Based on that, the c-granule representing the current hierarchy of the agent's needs is updated (along with the agent's priority needs).
3. An agent optimizes, in an adaptive manner, the current degree of the agent's global efficiency maximization in accordance with the conditions, resulting from the adaptive hierarchy of needs.
4. An agent does its best to satisfy her/his needs by discovering and learning, in a reasonably efficient manner, formulas that help to improve the efficiency of matching the agent's priority and adaptively changing needs.
   Therefore, an agent aims at increasing the efficiency of construction, judgment (also satisfiability), and adaptation when it comes to changing (generally in the PDCA cycles) appropriate interaction plans that are intended to satisfy the agent's priority of needs.

32.4.6 FPW-15: Communication Among C-granules and Among Agents

FPW-15 Abstract

A. The c-granules constitute the basic tools for modelling interactions among an agent, agent's c-granules, and the environment of her/his activity (c-granule links can also be treated as c-granules).
   It is important to remember that interactions also constitute a mechanism of communication among c-granules and other agents.
B. It is worth emphasizing that such an interpretation (including meaning) of the properties of a given c-granule G depends on the agent's perception of the context of interactions relevant to G. The context is based on how an agent registers and interprets the effects of interactions observed by her/him. The registration and interpretation can be the activities of a special c-granule G_I. We will refer to such a c-granule G_I (which interprets the properties of c-granule G in different applicable contexts of interactions) as the *context interpretation c-granule* for G.

C. *The context interpretation c-granules* can play an important role in the comprehension of the perceived hunks (including m-hunks) as well as in interactions. What is more, they may be significant for the cooperation/communication among agents.
   They establish aspects of the contexts and priorities that are currently important for an agent and her/his comprehension and indicate how to potentially use (according to the agent's knowledge) any currently communicated and/or perceived: hunks, m-hunks, and/or interactions.
D. We say that a *context interpretation c-granule* is a *semiotic interpretation c-granule* of a c-granule G, provided that it is obtained by the aggregation of the following packages of c-granules:

   1. Package of *pragmatic c-granules* relevant to G in specific contexts of its usage, i.e., for any context in which G is used, we may distinguish a family of links to the following:

      i. Agent's prioritized needs relevant to G in specific contexts, in particular, in solving problems and preserving constraints.
      ii. Parts of the agent's "windows," representing the focus of the *agent's attention* relevant to G in specific contexts of its usage.
      iii. Agent's scenes, their attributes, and relationships between them, e.g., mereology (part of), taxonomy, and cause-effect relationships (especially those, representing the negative and/or positive consequences).
      iv. Current and/or predicted contexts of, e.g., interactions, agent's needs, points of hierarchy of the current agent's interaction plans and next steps that are to be undertaken.
      v. SWOT analysis of the currently important (according to the agent's judgment) scenes.
      vi. *Cost-Benefit Analysis* (CBA) of the currently implemented and important (according to the agent's judgment) interactions plans.
      vii. Judgment regarding the current state of an agent and environment, resulting in the aggregation of all current judgments performed by an agent.

   2. Package of *semantic c-granules* relevant to G in specific contexts of its usage and for any context of its usage, i.e., for any contexts in which G is used, we may distinguish a family of links to hunks, interactions, m-hunks, c-granules, and relationships among them. The family of links indicates the following groups of entities:

      i. Currently perceived hunks, m-hunks, and/or interactions that have been currently recognized, classified, and represented by relevant *c-granules* and their interpretations, which fall under the *meaning of a semiotic c-granule*.
      ii. Examples of the agent's *c-granules* representing hunks, m-hunks, and/or interactions that fall under the *meaning of a semiotic c-granule*.

iii. Rules and examples of constructions of interaction plans, indicating how to satisfy some needs of an agent using hunks, m-hunks, and/or interactions that fall under the *meaning of a semiotic c-granule*.
iv. Explanation of how to use (rules of usage) the key concepts relevant to the c-granule G.

3. For a package of *syntactic c-granules* relevant to G in specific contexts of its usage, i.e., for any context in which G is used, we may distinguish a family of links among c-granules and/or hunks representing possible representations and communication of the semiotic c-granule by hunks, m-hunks, and interactions.
4. For a package of c-granules supporting the *modifications and development of semiotic interpretation*, relevant to G in specific contexts of its usage, i.e., for any context in which G is used, we may distinguish a family of links to c-granules for the development of a semiotic interpretation of a c-granule. **Semiotic interpretations of c-granules are subject to continuous changes. Usually, they are not perfect and they represent only temporary approximation (made by an agent) of their vague meaning**.
   Moreover, an agent must continuously learn and improve the most important (and most useful) semiotic interpretations of c-granules.

E. If for a c-granule G_1 there exists a c-granule G_2, which is a *semiotic interpretation of a c-granule* G_1, then by a *semiotic c-granule*, we mean an aggregation of c-granules G_1 and G_2.
F. Semiotic c-granules support the agent's control in improving the approximate reasoning in order to identify, specify, and understand the *current context of the agent's interactions*, as well as develop and implement the optimal interaction plans (in a given context). **A collection of semiotic c-granules used by an agent, together with the rules for their aggregation into new semiotic c-granules, is referred to as the** *agent's private language*.
G. Roughly speaking, a private language is comprehensible only to a given agent because the things which define its vocabulary are—in fact—inaccessible to others.
   In general, other agents have only access to some presentations (which can be intentionally modified) of the meaning, provided by the originator of a given concept from the private language. The concept of a private language was studied by Wittgenstein. In his book [539], we can find the following explanation:

   > The words of this language are to refer to what can be known only to the speaker; to his immediate, private, sensations. So another cannot understand the language.

H. Societies of agents can acquire, use, and develop an ability to communicate among themselves. For that purpose, they construct, use, and develop a *language of communication* that is adequate to their mutual needs. In the process of communication, two roles *sender* and *receiver* of a message, played by agents, are fundamental.

I. For example, a *sender*, in certain contexts of her/his activity, initiates interaction plans adjusted to particular situations, which results in the creation and identification of certain hunks, referred to as the *sender's artifacts*. A *receiver* perceives, judges and, possibly, records in her/his memory the coexisting situations, including the patterns of the sender's behavior along with the artifacts that have been created and indicated. Once a *receiver* considers such a coexistence of situations and artifacts sufficiently significant, she/he records the relationships and assigns them to her/his own c-granules (including names and types).

J. Agents can switch their roles of *senders* and *receivers*. It leads to a *dialogue* among agents. Such a dialogue can be treated as an interaction process which is some kind of a *language game* [539] played by agents. During the game, agents use some artifacts (like chess figures in a chess game): words, phrases of expressions, and the agent's body language.

K. As an example of a language game, one can consider the following situation presented in [539]:

> The language is meant to serve for communication between a builder A and an assistant B. A is building with building-stones: there are blocks, pillars, slabs and beams. B has to pass the stones, in the order in which A needs them. For this purpose they use a language consisting of the words "block," "pillar" "slab", "beam." A calls them out;—B brings the stone which he has learnt to bring at such-and-such a call. Conceive this as a complete primitive language.

L. As a results of a dialogue, agents can learn from one another and/or agree upon particular properties of interaction plans. This enables them to mutually arrange structures and properties of common *meta-social c-granules*, which include jointly undertaken interaction plans (cf. also FPW-03 points 11 and 12).

M. Every agent in a given society may create her/his own semiotic interpretation for any meta-social c-granule perceived by her/him. It may happen that every agent interprets a "meta-social c-granule" in a slightly different way. In such situations, agents may, in the form of a dialogue, arrive at a common c-granule or even—in justified cases—modify the c-granule.

N. Local societies of agents increase the efficiency of their communication by improving their meta-social c-granules.

O. The meta-social c-granules and the way they are interpreted using agents' own c-granules may be considered as a metaphor for a scheme of in-formation flow, as presented by Barwise and Seligman [26].

FPW-15: Exemplary Refinement

1. The agent's control aims to classify, describe, identify, and learn interaction plans in different contexts of the agent's interactions in the best manner possible. The contexts of the agent's activity are represented by the agent's c-granules. Generally, these c-granules representing contexts constitute an aggregation of other c-granules which represent the following:

i. Important (according to the agent's judgment) effects of the agent's perception in her/his environment of activity, characteristic of a given context of the agent's activity.
ii. Important (according to the agent's judgment) effects of perception (from the past), associated and stored by the agent's control, which are related to the agent's environment of activity and match a specific context of the agent's activity.
iii. Recommended priority of the agent's needs, characterizing a specific context of her/his interactions.
iv. Recommended interaction plans to be initiated in a specific context of the agent's interactions.
v. Reference to the current hierarchy of the agent's needs.

2. By the *judgment of the agent's context of interactions* of a given situation in the agent's environment of activity, we understand a c-granule that represents the following components:

   i. Prioritized variants of the agent's contexts of interactions, arranged in accordance with the degree to which the present situation in the agent's environment of activity is adjusted to the context of the interactions.
   ii. SWOT analysis results for an agent in the present situation.
   iii. Possibilities of undertaking interaction plans in a given situation.
   iv. Revised hierarchy of the agent's needs.
   v. Construction of a recommended plan for the upcoming interactions.
   vi. Judgment recording the recommended plan for the upcoming interactions in the context of the earlier interaction plans, implemented by an agent.

3. For a given c-granule G, by the *context interpretation of the c-granule* G, we mean another c-granule G'. The c-granule G' **assigns, to the input c-granule** G **and the interaction context judgment, prioritized links. This is done according to the needs resulting from the current judgment regarding the context of the agent's activity. These links point to examples, demonstrating to the agent's control such aspects of meaning of the c-granule** G **as**:

   i. Patterns of its occurrence in particular situations (compatible with the context in which the interpreted c-granule is being used) and development scenarios of these situations, together with the determination of potentially important effects of these scenarios (both increasing as well as decreasing the agent's satisfaction).
   ii. Patterns of interaction plans that determine potentially important methods of using c-granules (or their parts) and other resources available to an agent in the present context, which enable her/him to achieve her/his objectives.
   iii. Alternative names together with the contexts and methods of their use and archivization.

iv. Interaction plans leading to the development and adaptation of a c-granule in its present form, along with its present or currently developed context interpretation, expressed by means of suitable interactions.

4. An ability of an agent *ag* to communicate is based on the following aspects:

   i. Her/his perception of behavioral patterns (agent's expressions), represented by c-granules and linked with changes in the environment caused by another agent (or hunks), with whom the agent *ag* interacts.
   ii. Interpretation of behavioral patterns (agent's expressions) by an agent who perceives them.
   iii. Use of properties of such patterns of behavior (caused by another agent or hunks) in the decision-making process of the agent *ag* and in the actions undertaken by her/him.

5. Communication among agents is possible—to a large extent—owing to the following skills of agents:

   i. Agent's abilities to encode their internal c-granules in the form of communicated behavioral patterns (agent's expressions).
   ii. Perception of behavioral patterns (agent's expressions).
   iii. Interpretation of behavioral patterns (agent's expressions) in the form of c-granules by the receiving agent, who interprets the c-granules of the sending agent.
   iv. Processing of internal c-granules with the aim of undertaking actions, that result from the conclusions drawn from the interpretations of such expressions.

32.4.7 *FPW-16: Cooperation Among Agents for Problem Solving*

<u>FPW-16 Abstract</u>

If a society of agents meets a set of relevant conditions for cooperation (resulting from the specificity of a given class of problems that are to be solved) it may be capable of undertaking and engaging in a constructive cooperation when solving such problems. In this process, a particular role is attached to such aspects as given below.

A. The quality of commonly used and **adaptive ontology** of key concepts for initiating and implementing a project aimed at solving a specified class of problems.
B. Properly selected (especially in terms of the efficiency of initiation and implementation) **adaptive decomposition** of a problem to tasks that are to be completed by the properly prepared subgroups of agents.

C. **Efficiency of concept granulation and communication among agents**—particularly in the scope of prioritized issues that are to be solved at the present moment. A society of agents may improve the efficiency of problem solving by identifying and improving the optimal meta-social c-granules relative to the concepts (and relations among them) that efficiently support all aspects of solving current and prospective problems (cf. also FPW-03 points 11, 12, and FPW 15 points L, M, N, and O).

FPW-16: Exemplary Refinement

1. An important condition necessary for the initiation and implementation of a constructive cooperation in a given society of agents is the following: the society of agents which constitutes a part of a project (aimed at solving a class of problems), has the following properties:
   i. Ability to perceive:
      a. Emergence of rules describing agent's proper behavior as part of the cooperation with other agents when solving a given problem. For example, agents should particularly obey the adaptive rules of conflict resolution. A set of mandatory rules (including obligations and competencies) regarding the agent's behavior during the process of problem solving is referred to as the *agent's role* in a project.
      b. The rules for adapting the agent's role in a project to the changing conditions in the agent's environment of activity and to the phase and status of activities aimed at solving a given problem.
      c. Mutual communication and interaction with the environment—within the scope of a given domain, including the agent's role in problem solving and the rule for the adaptation of this role.
   ii. Placing the agent's role in a project at the relevant level of the hierarchy of the agent's needs, in relation to the adaptively changing role of a given agent in the process of problem solving.
   iii. Using agents with sufficient competencies and resources to ensure the proper satisfiability of their roles in a given project.
   iv. Assuring that the agents learn from one another to develop and master the abilities associated with the role they play in a given project and improve: (i) the interaction among them, (ii) the process of acquiring new knowledge, and (iii) mechanisms of adaptive judgment related to different issues of problem solving (with particular focus on their role in a project).
   v. Providing agents with the abilities to express arguments *for*, *against*, and *neutral* while discussing proposals for alternative solutions to problems and resolving conflicts (in a relevant time).
   vi. In a society of agents, at least four roles of agents can be distinguished:
      a. *Analysts*—agents who cooperate in decomposition of a problem to such a degree of complexity, that is easy enough for the agents from the group of *Constructors* to propose some solutions.

b. *Constructors*—agents who develop and propose the constructions of partial solutions to problems (based on a selection from different variants).
c. *Integrators*—agents who verify partial solutions proposed by "Constructors" as well as create and integrate proposals of solutions on higher degrees of integration.
d. *Supervisors*—agents who verify the correctness and quality of the entire process of problem solving, recommend rules for the adaptation and improvement of cooperation rules, and modify the definitions of roles for individual groups of agents.

2. Constructive cooperation of a society of agents when solving a problem is executed through the following steps:

   i. Systematic verification and adaptation of the quality that ensures the best possible conditions required to initiate and implement a constructive cooperation.
   ii. Cooperation of agents over the "optimal" problem decomposition, using an adaptive hierarchy of needs (in the form of meta-social c-granules in the societies of cooperating agents), that is the decomposition of a subproblem to *partial subproblems*.
   iii. The best possible constructions of solutions to partial subproblems. The development of partial solutions is managed and carried out in the PDCA cycles—based on efficiency management by local teams of agents and with particular attention paid to a possibility of using mechanisms such as:
      a. cause and effect relationships leading to the bottom-up self-organization, especially along the schemes of problem decomposition, and
      b. re-use of recurring (or analogous) schemes and components of partial solutions (both created in other projects and created for the sake of future projects).

3. A society of agents aims to ensure the most efficient cooperative problem solving in all aspects of this process. In particular, agents aim to ensure that during the decomposition of problems, partial problem solving, and integration of partial solutions to problems, a possibility to increase the efficiency of problem solving is taken into account by means of making use of factors as given below.

   i. Selection and skillful use of adequate knowledge and technology of problem solving in the most coherent way in many areas of the project implementation. In this selection, it is important to pay attention to the possibilities of using cause and effect relationships, which take place and/or may take place in the environment of the project implementation. The relationships which support the self-organizations of local teams of agents and results of their work are particularly important in order to help obtaining, on the basis of interactive iteration, the best possible solutions in the shortest time.

ii. Systematic identification and improvement of the optimal meta-social c-granules (especially, their adequate efficiency attributes that facilitate the process of reasoning leading to solutions) relative to the concepts which efficiently support various aspects of a constructed set of solutions and potential problems that might have to be solved.

4. Good cooperation within a society of agents during solving recurring problems may lead to the formation of a new agent. We refer to such an agent as a *meta-agent* (cf. also +).

Part VIII
WisTech Introduction to Efficient Acting, Learning, and CSE Project Implementation

Chapter 33
WisTech Approach to Agent's Efficiency Management

33.1 "Efficiency" in the Most Important Agent's Tasks

By *WisTech agents*, *agents* in short, we mean agents, who comply with the WisTech postulates (i.e., FPW-01-FPW-16). For example, WisTech agents may represent human beings participating in CSE projects, IT system components, or simply agents, as understood by the AI systems.

In this book, we focus on agents who perform certain tasks in CSE and in AI environments.

It should be emphasized that according to the FPW-04 postulate, included in Sect. 32.3.4, an agent may perceive and register only some properties of the results of interactions in her/his environment of activity. In light of the FPW postulates, the properties displayed by the states of the agent's control (Sect. 32.4.1), representing the satisfiability degree of the agent's needs and determining, on the basis of her/his current hierarchy of needs, top-priority actions that are to be initiated, are particularly important (according to the agent's judgment).

The WisTech postulates, described in Chap. 32, are rather general in nature. Analogously to each consistent mathematical theory, which may have an infinite number of interpretations (models), the FPW postulates may also have an infinite number of computational models, on the basis of which they can be interpreted. Therefore, in the next parts of the book, we will present only some possible interpretations of such computational models, based on the key points of WisTech approach. In these models, much importance is attached to the way in which the most important tasks of an agent, recommended as part of the FPW-14 postulate (Sect. 32.4.5), are interpreted.

According to the FPW-14 postulate, the adaptive hierarchies of the agent's needs, aiming to satisfy her/his needs in accordance with the changing hierarchy of needs, help to identify the priorities within the implementation of interaction plans in the most possible efficient way. To put it more simply, according to the postulate, the most important task of an agent is to construct, integrate, and implement a relevant

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_33

and efficient family of interaction plans, in order to increase the satisfiability degree of her/his top priority needs to an acceptable level.

It should be noted that the aims of the tasks (including tasks mentioned in the FPW-14 postulate) and the way, in which an agent executes them, are expressed by means of concepts; these concepts are represented by c-granules, and used by an agent as part of her/his private language. In general, the knowledge possessed by an agent and its representation in her/his private language are far from perfect. Consequently, the way in which an agent understands her/his most important needs and the way in which an agent executes such a task may significantly differ from the actual objective needs of an agent and the optimal fulfilment of these needs. However, an agent may recognize her/his mistakes (including her/his misunderstanding of aims and improper way of executing the most important task) through an interaction with the environment. Next, on the basis of her/his knowledge and adaptability of her/his judgment, an agent may undertake optimal (according to the agent's judgment) corrective actions. In this context, particular significance should be attached to the agent's ability to: estimate the potential scope of hardly predictable changes, identify any deviations from the expected properties of the environment, and recognize, in the quickest way possible, upcoming changes important for an agent. The FPW-12 postulate, regarding the SWOT analysis and risk of interaction plan (Sect. 32.4.3), requires that an agent possess such abilities.

For example, if an agent does not have adequate c-granules to express and recognize situations that pose a great risk to her/him, then, in case such a risky situation occurs, an agent is not aware of the fact that she/he should counteract it (e.g., undiagnosed "disease" that leads to her/his eventual destruction). Moreover, even if an agent has adequate c-granules to express and recognize risky situations, it may turn out that she/he has no c-granules which enable her/him to develop and undertake a proper interaction plan for counteracting the risks, that have already been identified.

In other words, **the quality of c-granules, used by an agent, has a direct influence on the quality of her/his performance**. Therefore, computational models which enable an agent to "compute"—in an adaptive manner—optimal approximations of c-granules used for efficient supporting an agent is her/his tasks, are so important (Sect. 32.4.5).

The situation presented above resembles a game (with incomplete information) between an agent and the environment, in which the agent's victory is measured, using attributes that evaluate the degree of completion of her/his most important tasks (Sect. 32.4.5). During the game, an agent, through interactions, increases her/his knowledge about the rules of the game and approximates these rules in her/his own private language. Unfortunately, the rules may unexpectedly change. Intuitions of this type were discussed in Sect. 28.3 and illustrated in Fig. 28.17.

In the figure, the arrows stand for actions, undertaken by an agent once she/he makes sure that all the conditions (symbolically represented as colorful ellipses), necessary for initiating a given action, have been met to a satisfactory degree. An agent, being able to judge whether the concepts that express her/his needs are fulfilled, initiates interaction plans that are, according to her/his knowledge, the best.

Of course, in general, an agent does not have a full knowledge about the concepts that represent her/his actual needs (in terms of the most important agent's task—Sect. 32.4.5). Neither has she/he got a full knowledge, that could enable her/him to implement the best interaction plans. In the commonly accepted model, an agent is "doomed to" the continuous process of learning and adapting the ever-changing perception of the environment, in which she/he acts.

Agents may learn and perform their tasks more effectively through communication and cooperation with other agents, as described in FPW-15 postulates (Sect. 32.4.6) and FPW-16 postulates (Sect. 32.4.7).

It is worth emphasizing that when it comes to practical application of WisTech agents in CSE and AI environments, the critical condition, deciding for the success of a given project, is a broadly-understood efficiency of interaction plans, implemented by agents. The meaning of the concept—the "efficiency" of WisTech agents—is determined under the FPW-13 postulate (Sect. 32.4.4, point H).

33.2 The Role of "Efficiency" and Its Interpretation in AI and CSE Projects

One of the key R&D objectives of engineers engaged in AI, is to equip the AI systems with the ability to effectively recognize the environment, in order to plan, execute, and adapt the actions that lead to success in terms of her/his hierarchy of needs. In general, the success of a given project is understood as the maximization of benefits and minimization of costs. Such an approach is based on AI paradigms that dominate in today's AI. As an illustration, let us quote the following description of AI included in [422]:

> the study, R&D and engineering of intelligent agents, where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.

In our approach, maximizing the agent's chances of success in a fast-changing, unknown environment is called the co-risk maximization (FPW-11). This requires a quick identification of emerging opportunities and threats, in relations to the strengths and weaknesses of an agent. In management theory and economics, analysis and planning techniques such as SWOT analysis [190, 191, 376, 377] are used. The SWOT analysis helps effectively identify key, alternative actions that can be initiated. Then, on the basis of a Cost/Benefit Analysis (CBA), optimal variants of actions for implementation are chosen. Usually, an efficiency management system is monitored and improved as part of the PDCA cycle (Plan Do Check Act) [219, 476].

Depending on the technical possibilities and available resources, the activities described above are carried out simultaneously. For example, if someone plans a wonderful holiday trip and then, on the day of the scheduled flight, learns that a war has just broken out in the destination place, she/he cancels the trip rather than waits till the planning and decision-making processes, related to the PDCA cycle, are done.

During each phase of the decision-making process, it is important to look at both sides of a given decision: its benefits and costs. Therefore, in this context, efficiency management also plays some role. The definition of the concept of "efficiency" can be found in almost every dictionary. For example, in the Merriam-Webster online Dictionary and The saurus (www.merriam-webster.com/dictionary/efficiency), the word is defined in the following manner:

> Efficiency is the ability to do something or produce something without wasting materials, time, or energy.

The concept of efficiency is used not only to refer to AI and/or CSE projects; it can be applied to describe all areas of the human activity. Given the pace in which our day-to-day life changes, the ability to perceive and analyze important changes in the environment, in order to undertake actions that will maximize the chances of success while minimizing costs, becomes an important aspect of the concept of efficiency. For any complex project, this process should be a part of the change control procedure. To simplify, efficiency management should be precisely synchronized with properly adapted change management. This is where the principle expressed by Heraclitus, known for about 2,500 years, comes into play: "change is the only constant in life."

The concepts of agent's "needs," "success," and "costs" represent typical complex vague concepts. As a consequence, defining them on the basis of non-complex and easily computable attributes is not that simple. The situation is even more complicated as these three concepts, "needs," "success", and "costs," often change over time. At any arbitrarily fixed point of time, they are in mutual relations with one another. By intuition—in general—the greater the success, the bigger the cost. However, this is not always the case. The way we understand "efficiency" today implies that we should do our best to depart from this rule in favor of another rule, that is "increase success and reduce cost." Therefore, some approaches to the concept of "efficiency" define it as the ratio of success to the total cost. However, in practice, such a definition is metaphorical and difficult to apply in real life.

The main difficulty lies in the fact that it is hard to measure needs, success, and cost by means of simple attributes.

What is more, these three concepts, used in the same context, cannot always be measured and aggregated by means of a value, expressed in a given currency.

For example, the cost of a project implementation often entails consequences, which are not measurable by means of financial ratios. These include: human health, human effort, unnecessary risks, security, time, energy, and other appropriate means.

Moreover, the very concept of "success" can vary in meaning for each agent. If we assume that agents have their own individual hierarchies of needs, the concept may be considered in relation to this hierarchy. In this context, the success may be defined as certain benefits or results, which satisfy particular needs of an agent. In short, success can be understood as gaining and/or creating certain benefits and/or results, that an agent needs. On the basis of the agent's hierarchy of needs, this approach allows us to compare various types of successes of a given agent. For example, we may say the more important needs an agent fulfills, the greater her/his success is.

In order to arrive at a more precise definition of the concept of "efficiency," it is worth taking into account the conclusions drawn from the considerations above. As a result, the traditional definition of "efficiency" provided by Merriam-Webster online Dictionary and Thesaurus (http://www.merriam-webster.com/dictionary/efficiency) can be defined more precisely as follows:

Efficiency is the ability to do successfully something or produce something in line with the needs and without wasting any costs (like money, time, human effort, human health, security, unnecessary risks, energy and other appropriate means). Maximizing the efficiency means ability to perceive important phenomena and changes in environment to take such actions that maximize chances of the success while minimizing all costs above acceptable level of costs.

A more precise definition of the concept of "efficiency" presented above may become the basis for further iterative clarifications of the term, that will help construct computational models for the implementation of a system supporting the process of efficiency management, based on WisTech.

The essence of another iterative clarification of the aforesaid definition is expressed as part of the FPW postulates (Chap. 32), especially the FPW-13 postulate (Sect. 32.4.4), which directly relates to the concept of efficiency. According to the FPW-13 postulate, the "efficiency" of an interaction plan is understood as the aggregation of its total cost, benefits, risk, and co-risk. An agent has a need to maximize the efficiency of interaction plans that she/he implements, according to her/his current priority of needs.

In our proposal, a great emphasis is put on the mechanisms for the formation, self-organization, and adaptation of hierarchical networks, composed of the societies of co-operating c-granules (FP2-13—Sect. 32.4.4) and the societies of cooperating c-granules that constitute agents (FPW-16—Sect. 32.4.7).

The approach to computational models based on interactions between co-operating societies of WisTech agents may be treated as a reference to the concept of CPSs[1] and Cyber-Physical Systems of Systems (CPSS) (cf. Sect. 6.3).[2] WisTech agents may be distributed among the physical components of a complex CPS (or CPSS) system.

The intricacies of these systems are described in more details in such works as [1, 223, 297]. In these works, we may find numerous comments on the importance of physical computations. To illustrate the idea, let us quote some examples from the works [223, 297]:

> [...] CPS are essential for the future of the system industry worldwide and collaboration at all levels, from practicing engineers to product architects, from tool makers to technology providers, from service to research, is necessary.
>
> [...] Embedded systems—some visible, others integrated into every day equipment and devices—are becoming increasingly pervasive, and increasingly responsible for ensuring our comfort, health, services, safety and security. In combination and close interaction with the unpredictable real-world environment and humans, they become "Cyber-Physical Systems" (CPSs), which act independently, co-operatively or as "systems-of-systems"

[1] https://en.wikipedia.org/wiki/Cyber-physical_system.

[2] http://www.cpsos.eu/.

composed of interconnected autonomous systems originally independently developed to fulfill dedicated tasks.

[…] From the German Agenda CPSs, Intermediate Results, Acatech, 2010: "Cyber-physical systems" typically comprise embedded systems (as parts of devices, buildings, vehicles, routes, production plants, logistics and management processes etc.) that

- use sensors and actuators to gather physical data directly and to directly affect physical processes
- are connected to digital networks (wireless, wired, local, global)
- use globally available data and services
- possess a range of multi-modal human-machine interfaces (dedicated interfaces in devices, or unspecific interfaces accessed through browsers, etc.).

[…] Applications with enormous societal impact and economic benefit will be created. Cyber-Physical Systems will transform how we interact with the physical world just as the Internet transformed how we interact with one another.

33.3 Efficiency Management as an Extension to Risk Management

The approach proposed above is based on the concepts of "risk" and "co-risk," introduced as part of the FPW postulates (Chap. 32), as well as on the generalizations and adaptations of these concepts, presented in the ISO 31k:2009 (or shortly ISO 31K:2009) risk management standard.

It should be noted that such an approach to the concept of risk does not fully meet the ISO 31k:2009 risk management standard. In the standard, the risk is understood as the "effect of uncertainty on objectives," which may be positive and/or negative. Therefore, our approach is closer to the older versions of the standard, in which the risk was related to the negative consequences and their influence on our plans and in which the process of risk management consisted, above all, of minimizing these negative consequences.

Nevertheless, it does not mean that in our approach, positive consequences that affect the implementation of a given project are ignored. However, they are addressed in the process of co-risk management, which has been introduced as part of the FPW-13 postulate.

Therefore, going back to the idea of risk, in our approach, risk management performed by an agent is understood as comprehensive measures, which should be implemented in order to minimize the most serious risks.

Of course, in practice, an agent who manages risk cannot counteract all possible events, during which she/he is exposed to some danger. When the risk is small (that is when the consequences are negligible and the chances of these consequences are low), an agent may undertake no actions. Such risks are sometimes referred to as *assumed acceptable risks*.

An agent initiates adequate forms of risk management, whenever she/he recognizes a great risk to the extent that is compatible with the constraints, in which she/he finds herself/himself. Therefore, the cost of executing an action by an agent may be misestimated due to some unpredictable risks that may appear amidst the assumed acceptable risks. At this point, it is worth recalling that in practice, the level of acceptable costs of actions, or assumed acceptable risks, executed by an agent does not have to be expressed in absolute values. The values may be expressed by the satisfiability to a degree of vague concepts and the degrees do not have to be expressed, using an absolute scale. In practice, these values are often relative and they are analyzed in relation to the current number of available values and/or an individual decision of an agent, that takes into account some additional, unique constraints. For example, financial investors may often say that the asset management cost constitutes approx. 2% of the initial asset value, and is paid on the annual basis. However, for numerous investors, the acceptable risk means low chances of loss, not exceeding 5% of the current asset value.

The aforesaid relativity of concepts related with costs can be addressed in a different way. When agents compete for the same limited resources, what constitutes costs for one agent may, in practice, it may constitute benefits to another. Therefore, if we shift the perspective from one agent to another, her/his competitor, we receive, in a sense, dual concepts. For example, if we were to dualize the concept of "costs"—understood according to the FPW-11 postulate as a decrease in the satisfiability degree of some of the agent's needs—we would receive a dual concept of "benefits." In other words, agent's benefits are understood as increasing the satisfiability degree of some of the agent's needs. In this sense, all positive results of changes, which increase the range and/or values of hunks/c-granules protected by an agent, are considered "benefits."

Analogous to costs, when it comes to benefits, an agent may also expect that during the course of a given interaction plan, she/he may gain certain benefits. Moreover, in this case, there may also appear some unexpected events, which can provide some additional opportunities. These opportunities, if skillfully used by the agent's strengths, may bring even more benefits. For example, a radical decrease in costs and/or greater chances for expanding the range and increasing the values of benefits, along with reducing some dangerous risks, may all constitute such an opportunity. In a nutshell, this means that during the implementation of an interaction plan, there may appear some chances for gaining additional benefits by an agent.

Acting in line with the scheme presented above, we can dualize the concept of "risk," just as we dualized the concept of "cost" (cf. FPW-13). From a practical point of view, however, in order to ensure a greater usability of the concept, its definition should take into account the fact that in practice, an agent does not possess unlimited resources and should not, if she/he is sensible enough, exceed a certain level of risk, related to the activities she/he performs. At the same time, agents, trying to take advantage of every opportunity, should ensure that their activities do not decrease the scope of benefits expected to be received, but instead, if possible, reduce the costs. Actually, an opportunity may consist of solely reducing the assumed level of acceptable costs and risks. Therefore, in practice, the dualized concept of risk should be considered in relation to the assumed levels of acceptable costs, risks, and expected benefits.

The above practical observation is a starting point for the concept of co-risk, introduced as part of the FPW-13 postulate. To simplify, let us assume that an agent has defined acceptable costs and risks, as well as expected benefits, for a given interaction plan. Therefore, once the plan is implemented, the actual costs and risks should not exceed the acceptable level, while the benefits should not be lower than expected. Then, the intuitive co-risk of an agent consists of two pieces of information:

1. about expected consequences, stemming from the event in which an agent, using her/his strengths, takes advantage of an opportunity and, as a result, receives additional benefits, while maintaining risks and costs at an acceptable level,
2. about the degree of chance, associated with the occurrence of the aforesaid consequences.

To be more precise, a given interaction plan and its assumed levels of acceptable costs and risks, along with expected benefits and potential opportunities that may be used by the agent's strength, co-risk is understood as a function, which aggregates the following:

- expected additional benefits, resulting from an opportunity of which the agent's strength took advantage (alongside maintaining costs and risks at an acceptable level),
- chances associated with the occurrence of:
  - opportunity and possibility of gaining additional benefits, by taking advantage of an opportunity, using the agent's strength (alongside maintaining costs and risks at an acceptable level),
  - additional benefits, received by taking advantage of an opportunity, using the agent's strength (alongside maintaining costs and risks at an acceptable level).

If we remember to maintain costs and risks at an acceptable level and keep in mind the expected benefits while managing co-risks, then we may dualize the ISO 31000 standards, used to support the process of risk management, in order to create standards for co-risk management. Below, we present a simplified scheme, showing the activity of an agent:

i. an agent, implementing and adapting different interaction plans, systematically estimates costs, risks, and benefits in order to relate them to her/his initial assumptions, and in case there are some significant deviations, she/he undertakes corrective actions,
ii. an agent systematically performs the SWOT analysis and looks for possible variants of interaction plans, within the limits defined by the previously assumed acceptable levels of costs, risks, and benefits, in order to maximize the co-risk,
iii. an agent, by performing a comparative analysis of different interaction plans, chooses, in an adaptive manner, the best variant and implements it,
iv. an agent evaluates, with adaptive judgment, her/his performance in relation to her/his expectations and on this basis, tries to improve her/his system.

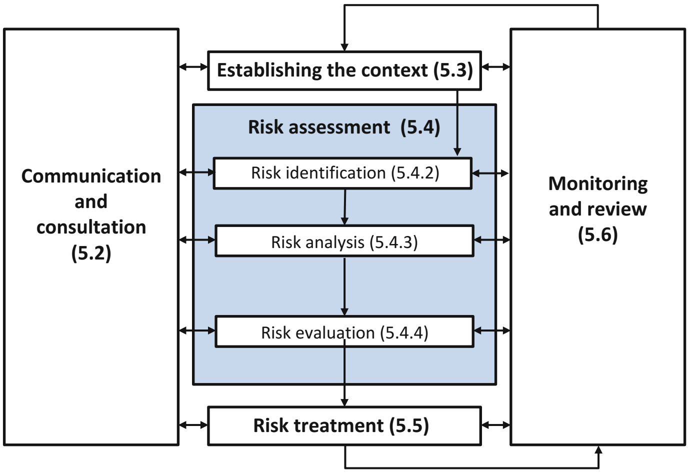


Fig. 33.1 Risk management scheme according to ISO 31K:2009. Numbers in brackets denote references to relevant sections in the standard ISO 31K:2009 describing the corresponding concept of the standard

The scheme presented above is based on the PDCA cycle [219, 476] as well as methods of planning, analyzing, acting, and monitoring, commonly used in economics and the theory of management. The SWOT analysis plays an especially important role here [190, 191, 376, 377].

The SWOT analysis helps to effectively identify key, alternative actions that can be initiated. Next, on the basis of a Cost/Benefit Analysis (CBA) and a concurrent risk assessment, an optimal variant of interaction plan, that is one that best fulfills the implementation conditions, expressed in the private language of an agent, is chosen. As part of the scheme presented above, it is important to increase the efficiency of an agent through the adaptability of techniques, used by an agent for reasoning and acting. The key aspect here is the ability to discover and predict the context and conditions in the agent's environment of activity.

The scheme presented above constitutes a generalization of the risk management process, as defined by the ISO 31K:2009 standard, illustrated in Fig. 33.1.

We suggest to modify the risk management scheme based on ISO 31K:2009, so that it distinguish and encompasses the concept of co-risk. The concurrent assessment of risk and co-risk is based on the SWOT analysis. As a result, all important risks and co-risks should be identified and evaluated. The analysis should lead to the integration of the SWOT results, on the basis of which, proposals for updating the agent's hierarchy of needs should be developed. Then, an agent should be able to generate proposals of interaction plans that could lead to the fulfillment of needs, created on the basis of the SWOT analysis. After a comparative analysis of different variants of interaction plans, an agent should make a decision which interaction plan will be used for managing risk and co-risk.

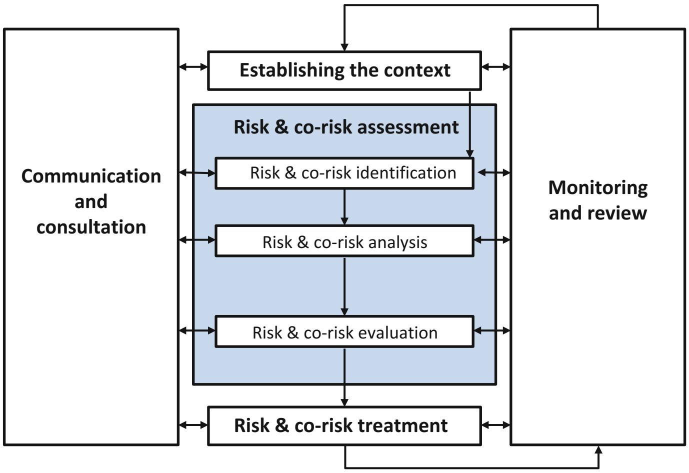


Fig. 33.2 Generalized efficiency management scheme—generalized risk and co-risk management

The scheme representing the integrated risk and co-risk management process, shown in Fig. 33.2, can be referred to as the agent's interaction plan for *efficiency management scheme* based on integrated risk and co-risk management process. Such integrated approach supports *agent's efficiency management*. More detailed illustration of the efficiency management scheme (based on WisTech approach) is in the Fig. 33.6 in Sect. 33.6.

33.4 Agent's Control and Reasoning Based on Agent's Efficiency Management as Illustrated by the Fire and Rescue Operation

There are numerous ways to elaborate the efficiency management scheme, presented in the previous Section, in Fig. 33.2, in terms of improving agent's reasoning capabilities with regard to the potential implementations of automatic efficiency management support systems. In order to illustrate this, we will present how to detail out the agent's control and reasoning models, independent of a specific domain knowledge.

In a sense, we are talking here about the laws of logics, that are important for all domains of the agent's activity (i.e., the laws of thinking independent of an application domain).

The simulation of the human capabilities to synchronize numerous threads of activities, performed as part of the PDCA cycle at various levels of granulation, will be our starting point. These threads include the agent's observation and her/his activities upon interaction scenes, which are within her/his field of attention. We may say that an agent focuses her/his attention on a particular scene (to use a metaphor, she/he sheds light on them with a torch, which stands for the agent's attention). The agent's attention is shifted on selected scenes, depending on her/his interpretation of the history of interactions with the environment and priorities of the already implemented activity scenarios. Such an agent may also implement a given scenario, without using her/his attention. It is possible that an agent starts and interaction, in a way that complies with the strategy devised by her/him, beyond her/his conscious realization (e.g., while she/he rests or is "consciously" engaged in other activities). By using her/his sensors/actuators, an agent may "consciously" be active in two scenes simultaneously—actual scene and imagined scene (created in a part of the agent's memory)—and autonomously adapt to the surrounding by means of adequate adaptive strategies.

In the previous paragraph, we proposed a framework for adaptation of the risk management processes (Fig. 33.1), which consists of efficiency management processes (Fig. 33.2).

When it comes to specific applications, the proposal needs to be elaborated on. Therefore, below, we present a bit more detailed approach to the agent's control and reasoning schemes, based on an agent, who supports the efficiency management through a commander of a fire and rescue operation based on the ICRA[3] [286] project.

In this approach, we propose how to detail out the general scheme, presented in Fig. 33.2, on the basis of the rough set theory. When referring to rough sets, we will use such concepts as IT systems, decision-making systems, rules, and classifiers. However, rough sets will be used here only for the sake of specification, not because they are necessary. Other approaches or their hybridizations could be used here as well (e.g., approaches based on fuzzy sets, neural networks, or evolutionary algorithms).

To start with, we must indicate decision-making systems, representing agent's hierarchies of needs and values, which will support an agent in the process of reasoning that leads to adaptive judgment of scenes and situations, recognized by an agent. In this case, we may assume that the hierarchies are based on such aims of an agent as given below.

1. Eliminating or minimizing losses in human and animal life, along with material losses and unnecessary costs related to the abuse of various materials, tools, and fire and rescue service resources.
2. Improving the techniques of conducting fire-rescue operations, education of a new personnel, and agents themselves.
3. Maximizing work comfort and optimizing the ongoing communication and cooperation between the participants of a fire and rescue operation.

First, let us assume that our agent has classifiers which, on the basis of attribute values, historical data, domain data, and sensory data, are able to:

1. Recognize the phase of a fire and rescue operation,
2. Select currently important action scenes and perform their optimal granulation (for example, we create a tree that is analogical to the hierarchy of scenes, represented in the form of maps in less and less precise scales),

[3]http://icra-project.org/?node=10.

3. Identify mutual relations between important scenes and significant information about potential risks and co-risks, which ought to be conveyed to the "surrounding" of a given scene,
4. Recognize potential optimal interaction plans, assigned to particular important scenes—with special regard to the interaction plans for computing attribute values, necessary for the analysis and assessment of risk and co-risk, as well as interaction plans for dealing with the most important risks and co-risks.

For simplicity, we assume that a single agent may act only at the level of a single scene. On the other hand, we may assume that we have a whole society of agents at our disposal. In this case, our agents act upon particular scenes and cooperate with each other so closely, that they create in fact one integrated meta-agent. As observers, we perceive and implement this action by watching the activities of a meta-agent or, to simplify, by looking at it from the perspective of an agent, who monitors only one particular scene of a fire and rescue operation. From the perspective of such an agent (assigned to a given scene), **we may, for example, perform the following activities within the PDCA cycle**:

1. Triggering adequate classifiers and possible communication between agents and/or domain experts, performing recognition of mereology of scenes and judgment of priorities of importance of scenes (and/or importance of parts of scenes), SWOT analyses and assessments for important scenes,—SWOT identification, analysis, evaluation, and assessment—to determine, in the quickest way possible, any opportunities and threats in relation to strengths and weaknesses (in particular vulnerabilities) for important scenes monitored by an agent.
   As one of the results of the SWOT analysis agent (or a team of agents) can identify: strengths, weaknesses, opportunities, and threats for important scenes. Moreover, the agent (or a team of agents) can construct the risk and co-risk matrix for hierarchy of important monitored scenes. In order to do this for each important scene agent (or a team of agents) may construct the matrix of likelihoods $p_{i,j}$ (Fig. 33.3), where $p_{i,j}$ is the likelihood of conjunction for two events: the first one is that the threat i uses the weakness j and the second one is that occur in the expected negative consequences of usage of the weakness j by the threat i for given scene. Let us denote by $c_{i,j}$ the expected cost of the negative consequences following usage of j by i. Thus for any i and j we get a risk as an aggregation of $p_{i,j}$ and $c_{i,j}$. In this situation we write: $risk_{i,j} = p_{i,j} * c_{i,j}$ (where symbol of $*$ denotes an aggregation, in particular, it could be multiplication or addition).
   Analogously, an agent may construct matrix of likelihoods $q_{i,j}$ (Fig. 33.3), where $q_{i,j}$ is the likelihood of conjunction of two events: the first one is that the strength j uses the opportunity i and the second one is that occur the expected positive consequences of usage of the opportunity i by the strength j. Let us denote by $b_{i,j}$ the expected benefit of the positive consequences following usage of i by j. Thus for any i and j we get a co-risk as an aggregation of $q_{i,j}$ and $b_{i,j}$. In this situation we write: $(co - risk)_{i,j} = q_{i,j} * b_{i,j}$.
   In the general case the aggregation of likelihoods and consequences in the risk and co-risk definitions (the aggregation is denoted by symbol $*$) can be very com-

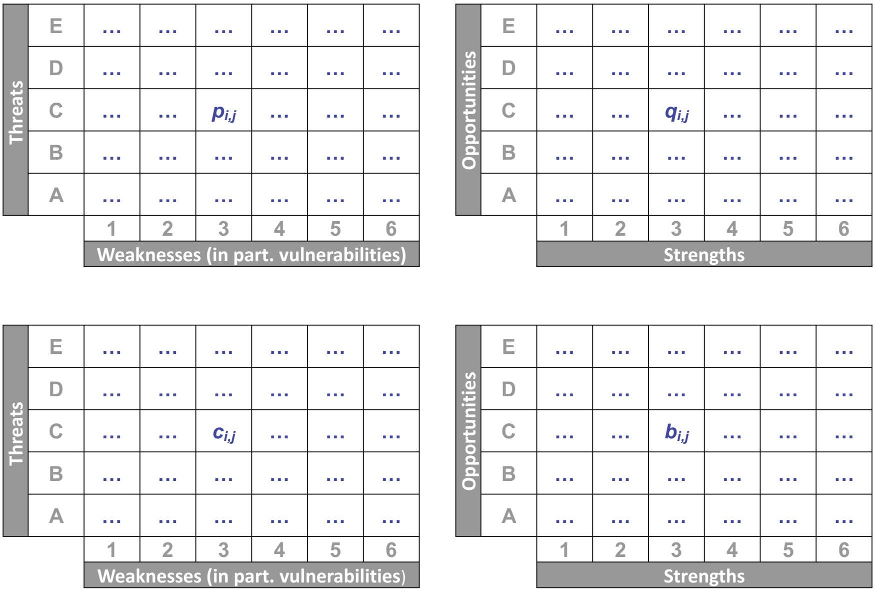


Fig. 33.3 Threat, opportunity, cost, and benefit matrices

plex. In the simplest case we may interpret the aggregation as calculation of the expected value (i.e., multiplication of probability and the value). In this situation $*$ can be just multiplication of real numbers. Thus we have:

$$risk_{i,j} = p_{i,j} * c_{i,j} \tag{33.1}$$

$$(co - risk)_{i,j} = q_{i,j} * b_{i,j}, \tag{33.2}$$

In simple practical applications, very often, **logarithmic scales are used for rating** of consequences and likelihoods. In such case, **addition rather than multiplication** is applied.

The risk and co-risk matrices may occur as shown in Fig. 33.4. In risk and co-risk matrices, risks and co-risks should be adequately placed and individualized in a way, which is convenient for the users. Having these risks and co-risks, an agent, cooperating with other agents and domain experts, may start working on the recommended interaction plans for further interactions.

It should be emphasized that for most of the complex practical applications **any ad hoc algebraic definition of the aggregation (of likelihoods and consequences) may lead to approximations of unacceptable quality**. Thus for many applications, any kind of algebraic approach based on presumed a priori formulas is not satisfactory for acceptable approximation of risk and co-risk ratings. In general case, **the aggregation (of likelihoods and consequences) should be a result of**

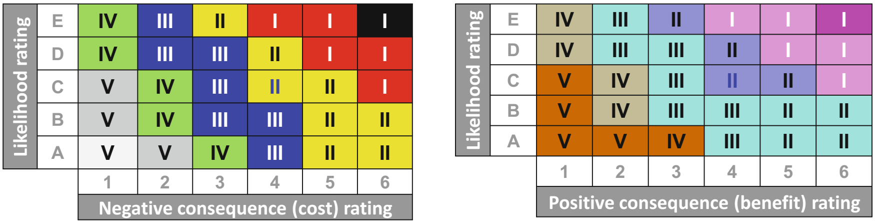


Fig. 33.4 Risk and co-risk matrices

adaptive judgment. In particular, the judgment result should be computed using adaptive strategies defined over properties of interactive computations implemented by c-granules.

The **strategies for computation of judgment results can be based on aggregation of relevant schemes of interactive granular computations. For example it can be aggregation of relevant schemas of:**

i. **linking c-granules and hunks**,
ii. reasoning (e.g., inductive, deductive, defeasible),
iii. heuristics,
iv. **testing and verification quality of c-granules**,
v. learning,
vi. discovering,
vii. acquiring,
viii. communication,
ix. cooperation.

These strategies are also controlled by relevant c-granules. The "societies" of c-granules, supporting the interactive computations, should represent the domain knowledge of the considered application. In particular, they should represent relevant contexts extracted from the domain ontology and they should provide links to additional knowledge (in particular, links to schemas of interaction plans) used for the aggregation (i.e., adaptive judgment) implemented by systematic adaptive control of interactive computations in the unpredictable environment.

2. Communicating the results of risks and co-risks to other agents (especially those from the neighboring scenes as well as superordinate and subordinate scenes).
3. Adapting the agent's hierarchy of risks and values and, if necessary, correcting the assumed levels of acceptable costs, risks, and expected benefits, based on the communication with other agents and ongoing judgment of a situation. For example, it may turn out that we are faced with a dilemma whether to save numerous people by sacrificing the life of a single person or risk the death of everyone.
4. Proposing at least four *basic interaction plans*, including those that focus on:

   i. Preventing the greatest risks.

ii. Deriving the greatest benefits from the greatest co-risks at an acceptable level of costs, risks, and benefits.
iii. Implementing the safest, in terms of protecting the values of an agent, variant of fulfilling the most important and current needs of an agent.
iv. Integration of the above interaction plans having in mind the most important task of the agent (FPW-14).

5. Updating the assumed levels of acceptable costs, risks, and expected benefits for three basic interaction plans.
6. Performing comparative analysis of particular interaction plans in terms of their efficiency, including the judgment of assumed levels of acceptable costs/benefits and the assessment of risks and co-risks.
7. Selecting the best variant of an interaction plan, on the basis of the comparative analysis.
8. Implementing the best variant of an interaction plan, based on its systematic efficiency management.
9. Verifying the quality of the implementation in relation to the expected results and, if necessary, proposing corrective actions and returning to the SWOT analysis (point 1).

Let us observe that choosing the efficiency management as the core of the agent's control and reasoning, enables an agent to fulfill her/his adaptively changing hierarchy of needs and values in an optimal and adaptive way. On the other hand, we should note that in fact, such methods as SWOT, Cost/Benefit, and Risk Assessment originate from the classic AI techniques of constructing and developing adaptive classifiers which make decisions, using the adaptive techniques of conflict settlement (i.e., finding a balance between arguments "for" and "against"). In practice, classifiers that are often used in the implementation of adaptive judgment mechanisms have a hierarchical or, in other words, network structure. Such a network of classifiers, along with the techniques for settling conflicts at various vertices (between arguments "for" and "against"), very often represent the ontology of concepts and reasoning techniques, that reflect an advanced domain knowledge.

The above ideas were presented in such a way as to make it easy to imagine their implementation, using rough set techniques, based on the processing of decision and information tables. In the approach proposed by us, these concepts are generalized into c-granules. As a result, we arrived at hierarchical c-granular structures which, in fact, may be also represented in the form of c-granules. The c-granules in these hierarchies are responsible for judging to what extent some complex, usually vague concepts, related to some important conditions which, once fulfilled, should trigger a particular action, are satisfied. For example, when a firefighter spots a fragment of a burning ceiling falling down on her/him, she/he should immediately rush to a safe place. In general, hierarchies of c-granules which receive "images" transfer these "images" to other c-granules, at higher levels, for processing. In this way, consecutive c-granules are activated. At some stage of the process, c-granules, which verify whether particular conditions, necessary to trigger certain actions, are fulfilled become activated. The ability of an agent, or a group of agents, to survive in a given environment in an "optimal" way depends on how "optimal" the actions undertaken by this agent, or the group of agents, are, and how "optimally" the reactions and actions of the environment are recognized. The situation resembles the process of learning the rules of an interactive game between an agent (or a group of agents) and the environment. In a sense, agents, by operating in an unknown environment, learn these rules. In other words, they become familiarized with the laws of a given environment in a slightly different language. Knowing these laws, or rules, it is easier to project the consequences of particular actions, and control them in order to effectively manage the efficiency of interaction plans, implemented by an agent. During the whole process, one should not forget about the hierarchical structures of management. For example, extinguishing a fire at the top floor and overlooking the fire spreading at the lower floors may lead to a tragedy. Therefore, agents from various scenes of fire and rescue operations make proposals regarding possible interaction plans, but they have to be coordinated and optimized from the perspective of a meta-agent, who monitors the whole operation. The situation is illustrated in Fig. 28.17.

In Fig. 28.17 we have three concepts that suggest undertaking three different actions. Final decisions should be made by a monitoring agent at a higher level, in agreement with a meta-agent who manages the whole operation. Of course, we should remember that during the entire operation, we also learn and improve our decision processes. The quality (and efficiency) of learning the rules of an interactive game between an agent (or a group of agents) and the environment depends largely on the agent's capability of broadly-defined judgment, which is used to evaluate and select both important processes that occur in the environment and knowledge necessary to identify and solve any possible problem. This brings us back to the concept of the agent's adaptive judgment and its functionality, as seen from a different perspective.

33.5 Engine for Efficiency Management (EEM)

In Sect. 33.4, we have introduced the idea of supporting the commander of a fire and rescue operation on the basis of WisTech, implemented using the "adaptive behavior" of co-operating, hierarchical "societies" of c-granules and/or WisTech agents (including domain experts). It should be noted that the idea is universal in nature, that is it can be adjusted to any application domain that supports the efficiency management. Further in the book, the idea will be expanded as various aspects of its possible implementation will be elaborated on. All these activities are aimed at proposing a framework for hierarchical architecture of a multi-agent system, in which the WisTech agents (including domain experts) will be processing c-granules, performing interactive granular computations to support the process of efficiency management in a given (arbitrary) domain of application. The system will be called the Engine for Efficiency Management (EEM).

EEM will deal with hierarchical computations or, to be more precise, computations, performed by co-operating hierarchies and/or competitive networks of WisTech agents and/or their c-granules. Intuitively speaking, the computations will be performed by concurrent cycles of mutually co-operating computational processes, resembling metaphorical PDCA cycles. As shown by the example of a fire and rescue operation, EEM computations are performed at various levels of specification by c-granules and/or networks of interrelated c-granules and/or WisTech agents.

At the topmost level of the model, there are c-granules (including WisTech agents) that implement a metaphorical PDCA cycle, responsible for the "processes connected with controlling the agent's attention" and for her/his activities, aimed at fulfilling her/his currently most important needs. To identify her/his needs and threats, an agent uses evolutionarily perfect attributes (c-granules), which express adequate properties of computations, observed by an agent. Therefore, agent's reasoning (which constitute interactions between c-granules of an agent) about processing the properties of processes, observed by her/him, are hierarchical in nature, which means that reasoning (computations) at lower levels are used for the construction of reasoning (computations) at higher levels.

For example, such a model of hierarchical reasoning, implemented by a model of hierarchical granular computations, performed as part of the PDCA cycles, may be presented, using the following examples of c-granules, responsible for particular components of a metaphorical PDCA cycle:

1. Monitoring hunks (including their configurations) in the environment as well as "images," representing the states of the agent's control (agent's "mind"), by means of her/his sensors—in other words, monitoring configurations of hunks, representing the agent's cache (an intuitive equivalent of RAM) and configurations of hunks, that implement the agent's control.
2. Monitoring and supervising c-granules that are currently, or has already been, activated by an agent, as well as their subordinate c-granules, supervised within a c-granular network.
3. Identifying c-granules which may lead an agent to adverse states (and/or desired states). In this process, the key role is played by the current hierarchy of c-granules, which represents the agent's hierarchy of needs and values.
4. Representing an action or a program (plan) of activities, aimed at reconstructing configurations of an agent and/or hunks in the environment and creating a new configuration, that meets the expectations of an agent.
5. Constructing a new configuration (represented by a new c-granule) as part of the previously planned and/or currently implemented action or program (plan) of activities to be performed by an agent.
6. Creating a new c-granule that represents a new perception of an agent regarding the actual configuration of hunks and agent's states, as they are at the moment of the agent's observation.
7. Monitoring the differences between a new configuration, as expected by an agent, and the actual new configuration, as perceived by an agent.
8. Determining the scope, form, timing, and conditions of corrective actions (adaptation), that may have to be performed by an agent.
9. Representing computations that indicate local moments of the agent's time.

The societies of co-operating agents may co-create next levels of hierarchical, interactive granular computations that perform more complex reasonings. Agents who cooperate with each other co-create an agent, called a meta-agent (in relation to the level of the agent's hierarchy that is subordinate to her/him). The framework of performing these computations at consecutive levels of computational hierarchy is relatively simple. Each agent (meta-agent, respectively) from a society of agents (meta-agents, respectively) remains in an interaction with the environment (surrounding). As a result, an agent determines next configurations of a computation, according to the local time of an agent (meta-agent, respectively).

Once granular computations become adequately advanced, it may turn out that the agent's configuration encompasses c-granules, which represent information about other c-granules—those that had already been activated by this very agent. These c-granules may represent both information about the environment and information about the state of an agent. In such a case, an agent tries to reconfigure its configuration (which is a specific c-granules in itself) into a new configuration (new c-granules) that, according to her/his expectations, is better than the previous one and/or leads to a better configuration. At this time, agent's functionalities, such as her/his ability of judgment (both in relation to particular situations and possible methods of generating plans of activities) and her/his ability to evaluate plans of activities become very important.

On the basis of her/his judgments, an agent strives to reconfigure her/his configuration into a new, better configuration, according to her/his expectations. Being equipped with adequate functionalities (especially the ability of judgment), she/he performs the reconfiguration, using an adequate c-granule, that is one that meets the expectations, resulting from such a transition (from an old configuration of hunks to a new one).

On the other hand, an agent monitors—within the limit of her/his "senses" and relationships among them—changes both in the environment and in her/his states. This leads to the creation (construction) of a new c-granule, which represents a new perception of an agent regarding the actual configuration of hunks, as it is at the moment of the agent's observation.

An agent has special c-granules to monitor the differences between these two configurations (i.e., a new configuration, as expected by an agent, and the actual configuration, as perceived by an agent).

An important issue, in the presented approach to the model of interactive computations on c-granules (observed from the perspective of a single agent), is that an agent observes the world and its context as she/he perceives it, using dynamic c-granules. On this basis, an agent generates a c-granule that represents an integrated image of a "desired" world and its context. The construction of such a c-granule should be integrated with the construction of a c-granule that represents a "feasible" plan of interaction, which will help to achieve the "desired" world. As part of this interaction plan, an agent undertakes proper activities. Next, she/he compares the actual results of her/his activities with her/his expectations. She/he analyzes the differences and does her/his best to perform corrective actions and/or corrects the currently implemented interaction plan. An agent may also decide to focus on a different area of activity.

Efficiency management should skillfully integrate and optimize risk and co-risk management. If we use CBA (Cost/Benefit Analysis) as a metaphor, we may understand the process of efficiency management simply as a multi-criteria optimization of a metaphorical quotient (Cost/Benefit) from the perspective of numerous needs, present in the agent's hierarchy of needs. At the same time, it is worth noticing that a reasonable WisTech agent has also a need to minimize risk and maximize co-risk. Therefore, co-risk maximization and risk minimization is equivalent to the broadly understood benefit, derived by an agent. However, dually speaking, increasing risk and decreasing co-risk leads to the broadly understood costs, incurred by an agent. Thus, in the case of such WisTech agents, CBA (Cost/Benefit Analysis) also encompasses the analysis of consequences, that reflect changes in the risk and co-risk of a given interaction plan.

The granular computations and the framework assumptions related to risk and co-risk computation techniques, included in Sect. 33.4, may be specified by defining the functionalities of specific c-granules (for computations of costs, benefits, risks, co-risks and IP_{BI}), which can become the components of the EEM system. For example, these can be c-granules which meet the functionalities described below:

1. Establishing of current assumptions, context and tools.
    a. Ontology of basic c-granules.
    b. Constraints and context.
    c. Hierarchy of needs.
    d. Classifiers.
    e. Optimizers.
    f. Judgment and adaptation strategies.
    g. Wisdom governance.
    h. Others.
2. Efficiency judgment.
    a. Scope.
        Cost and benefit.
        SWOT.
        Risk and co-risk.
    b. Main activities.
        Identification.
        Assessment.
        Aggregation of total cost, total benefit, risk, co-risk, and the need IP_{BI} for the family of perceived interaction plans (cf. FPW-13, point F).

3. Efficiency treatment.
    a. Adjustment of the hierarchy of needs.
    b. Variants of PDCA interaction plans.
    c. CBA for variants of PDCA plans.
    d. Selection of a PDCA plan.
    e. Implementation of the selected PDCA plan.
4. Efficiency management improvement.
    a. Performance judgment of efficiency management.
    b. Pareto analysis of the PDCA plans implementation.
    c. Improvement of assumptions, context and tools.
5. Monitoring and review.
6. Communication and consultation.

33.6 An Approach to CSE Based on WisTech Efficiency Management

The CSE project implementation model can be treated as a particular case of a model of interactive granular computations, that comply with the Framework Postulates for WisTech (FPW, Chap. 32). The aims of CSE projects are pursued by discovering, creating, processing, and adapting approximations of very complex c-granules in an interactive and systematic manner. Next, the approximations are activated and used to create new c-granules, which increasingly approximate the actual aims of a given project.

The implementation of a given project is also related to a concurrent generation of "evolving societies" of c-granules, that support the execution of physical and intellectual components of a project. These c-granules activate interactions, using their links. We may infer about the course of such processes, using efficiency management mechanisms, described in Part VII (Chap. 32) and Part VIII (Chap. 33).

In Sect. 33.5, we described the process of efficiency management based on WisTech. The process is represented in Fig. 33.5. It constitutes the basis for a framework architecture of the Engine for Efficiency Management (EEM). In the next sections of the book, we will discuss this model in relation to its possible application in CSE.

Let us recall Fig. 33.5, which illustrates the process of efficiency management in the context of WisTech.

Each box symbolizes processes, which, in our case, constitute meta-social c-granules. These meta-social c-granules are generated, processed, adapted, and archived by the societies of project's stakeholders, artificial agents, and agents from the environment (e.g., people and groups of people or artificial agents), who are affected by the results of interactions, related to the implementation of a project. We

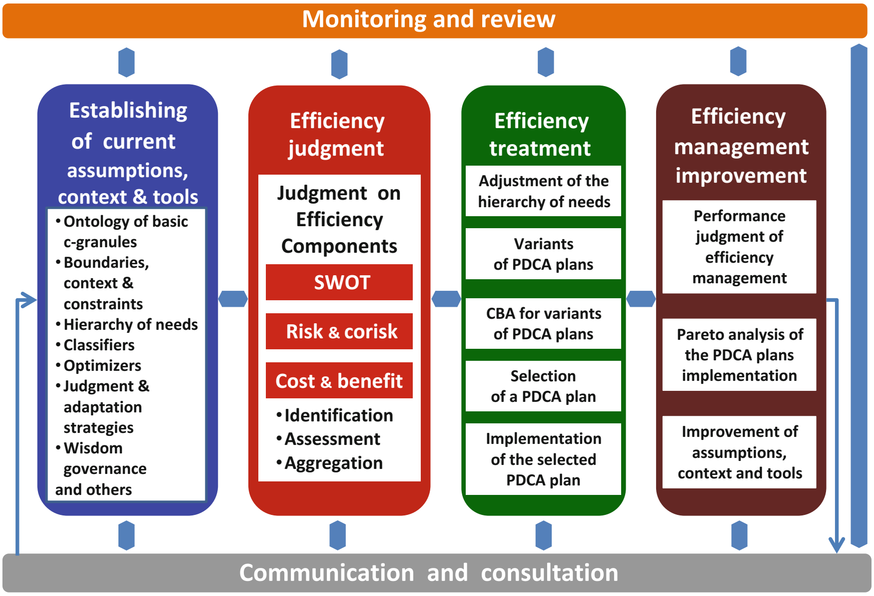


Fig. 33.5 The main components of a framework architecture for EEM

deal with artificial agents, e.g., in the case of CPS systems.[4] The c-granules presented in the form of boxes constitute aggregations of numerous other c-granules, some of which are presented inside the boxes. Their role in CSE projects is discussed below.

The arrows represent interactions between c-granules located in particular boxes and/or their c-granules, represented by these boxes. In the figure, we have two types of interactions:

A. Two-headed, broader arrows, which represent typical interactions, initiated by the links of c-granules in the boxes, which transmit the results of interactions to c-granules from one box to another. They encompass bi-directional communication and mutual transfer of the control between c-granules in particular boxes.
B. One-headed, narrow arrows, which represent interaction initiated by the links of c-granules in the boxes, from which the arrows start.
   The links transmit the results of interactions to c-granules from one box to another. They include communication in the direction indicated by the arrow and the transfer—also in the direction indicated by the arrow—of the control between c-granules in particular boxes. The arrows also initiate and end the process of efficiency management. They constitute the superordinate control of the PDCA cycles, performed as part of the whole scheme.

Below, we describe the most important c-granules, represented in Fig. 33.5, from the perspective of their application in CSE. Let us recall that the primary aim of

[4] http://en.wikipedia.org/wiki/Cyber-physical-system.

this society of c-granules is to support the process of efficiency management when it comes to generation, processing, and adaptation of c-granules, used by stakeholders, participants, and artificial agents, who implement CSE projects.

The most important c-granules, illustrated in Fig. 33.5, include:

1. Establishment of current assumptions, context, and tools. Each action plan and activity of an agent is based on her/his individual image (or m-image) of the environment, in which a given plan is to be implemented. This image is sometimes referred to as a *model of environment*. To put it more simply, **in order to implement a project, we have to make some *project assumptions*, that are all important assumptions concerning a given project (including assumptions related with the model of environment, techniques used in a project, and the context of the project implementation)**. At the same time, agents do not have any guarantee that their assumptions will conform with the actual state.
   Basically, in the case of complex projects, implemented in an unknown environment, **agents make decisions without having a full knowledge about the environment and therefore, may only be sure that the project assumptions made by them do not conform with the actual state to some unknown degree**. In such a situation, information about risk and co-risk evaluation, based on the degree of nonconformity of project assumptions with the actual state, as well as information about costs and benefits related to increasing the level of accuracy and certainty of project assumptions are of utmost importance for agents.

   a. Ontology of basic c-granules. The process of updating the concepts used by agents (which constitute examples of c-granules) is critical when it comes to the quality of each venture. In numerous CSE projects, their significance consists of discovering and using adequate c-granules. Very often, c-granules used in the project are too complex to comprehend for the majority of agents. Therefore, it is important to generate c-granules that constitute their proper approximations. These approximations should be adjusted to the level of the project participants and developed, along with broadening the knowledge of the users.
      A crucial aspect in this area is to establish and keep updating the project-related knowledge, especially when it comes to the understanding of key project concepts and the relationships between them (particularly, mereologies and taxonomies).
   b. Boundaries, context, and constraints. It is important to systematically update the knowledge about boundaries, which help to define the broadly understood scope of a project (e.g., space, time, content), identify important aspects of the context which may affect the course of a project, and obtain knowledge about constraints.
   c. Hierarchy of needs, mereologies, and taxonomies of needs. The hierarchy of needs includes aims, project requirements, and changing priorities of the current tasks. In order to realize how difficult it is to construct, update, and specify this c-granule, one should keep in mind the fact that management of user's

requirement in complex projects constitutes one of the main reasons why big IT projects ends in failure.

d. Classifiers. In each CSE project, it is necessary to continuously improve the accuracy and diversity of object classification. Often, the available classifiers do not take into account—at least to a satisfactory degree—the specificity of the domain knowledge in a given project. As a result, the systematic improvement of these classifiers is required.
e. Optimizers. Analogically to classifiers, when it comes to this c-granule, it is necessary to constantly improve the accuracy and diversity of optimal solutions to numerous problems (for instance, those related with efficiency management) in each CSE project. Often, the available optimizers do not take into account—at least to a satisfactory degree—the specificity of the domain knowledge in a given project. As a result, the systematic improvement of these optimizers is required.
f. Judgment and adaptation strategies. All decisions and practical conclusions are made, using available schemes of judgment. At the same time, the majority of important aspects (including c-granules) related to the implementation of CSE projects requires a systematic adaptation. Therefore, one may risk saying that the strategies of judgment and adaptation constitute the most important factors that decide for the success of CSE projects.
g. Wisdom governance. Let us recall that in our approach, the concept of Wisdom encompasses such concepts as: data, information, and knowledge. Therefore, when speaking about "wisdom governance," we also mean "data governance," "information governance," and "knowledge governance." By "governance," we understand such aspects as:

    i. *"wisdom management",* including: acquisition, classification (including taxonomies and mereologies), standards (including forms of c-granules, which represent appropriate types of wisdom), storage, sharing (including channels and forms of distribution and trainings), processing, quality, and security. It is also important to provide an access to "sources of wisdom" of suitable quality. This is iportant especially when it comes to necessary domain experts and packages of techniques, used to obtain wisdom from these experts, as well as a possibly comprehensive explanation of the context and wisdom interpretations, important for a given project.
    ii. *"wisdom policy rules and practices",* including the formation, implementation, enforcement, and improvement of a given policy, with particular emphasis on its synchronization with the policy of managing human resources, business partners, and competitors.

h. Others.

2. Efficiency judgment. Appropriate judgment of various aspects of efficiency, within the boundaries, context, and constraints of a project, constitutes the essence of a proper efficiency management. Some of these judgments should be aggregated in order to determine a global judgment of a project. Aspects that are less

significant should be judged locally, not to increase the cost of the project's development. At the same time, if stated that these less significant aspects may have an influence on the entire project, they should be included in the global judgment. It should be emphasized that the usefulness of a judgment depends on how fast particular information reaches to appropriate agents (including decision makers). The results of a judgment should enable an agent to react in a proper way.

a. Scope, mereologies, and taxonomies. The scope of a judgment depends on the specificity of a project. Establishing the scope of activities related to "efficiency judgment" is strictly connected with the ontology of concepts, related with specifying SWOT, risk, co-risk, cost, and benefit. Mereologies and taxonomies used by "efficiency judgment" are especially important. Within the scope of "efficiency judgment," the following three basic components should be taken into consideration:
   i. *SWOT.* The concept of SWOT is discussed as part of FPW in Sect. 32.3.7.
   ii. *Risk and co-risk.* The concepts of risk and co-risk are discussed as part of FPW in Sect. 32.4.4 and in this chapter.
   iii. *Cost and benefit.* The concepts of cost and benefit are discussed as part of FPW in Sect. 32.4.2.
b. Main activities. By intuition, the main activities related to "efficiency judgment" constitute the generalization of common activities, performed in the process of management, such as risk management, security management, and quality management during the identification, analysis, and evaluation of risk, security, and quality, respectively. These activities are performed in relation to the components that are within the scope of "efficiency judgment," that is, at least within the scope of such aspects as SWOT, risk, co-risk, cost, and benefit.
   i. *Identification.* In this respect, special importance should be attached to the processes of discovering and establishing appropriate concepts (c-granules, including their attributes), which are described in "Ontology of basic c-granules," as part of the paragraph entitled "Establishment of current assumptions, contexts, and tools." Without appropriate c-granules, it is impossible to identify key components, needed to perform correct analyses: for SWOT, risk, co-risk, cost, and benefit.
   ii. *Assessment.* Analysis and evaluation of values of various aspects of efficiency, which underwent the process of judgment in numerous areas, may be automatically supported by appropriate classifiers and optimizers.
   iii. *Aggregation.* In this area, it is extremely difficult to specify the techniques for aggregating the results of judgment from particular partial areas of judgments. The main objective of the aggregation is computation of the efficiency of a family of all interaction plans perceived by an agent (or groups of agents). The concept is defined in FPW-13, point D. The choice and interpretation of scales is also related to this problem. For example, if we were to adopt a probabilistic approach, we could treat the risk as an expected random variable, that expresses negative consequences. There-

fore, in this approach, it may seem convenient to treat the aggregation of opportunities and losses as the product of probability and cumulative amount of losses. However, in numerous applications, it is better to adopt a logarithmic approach in order to determine the amount of opportunities and losses. In such a situation, it may be more convenient to add opportunities and losses. Of course, there are many more possibilities of aggregating c-granules, which express opportunities and consequences.

3. Efficiency treatment. The treatment of efficiency judgment (including SWOT, risk, co-risk, cost, and benefit judgment) may be based on the aggregation of elementary actions and reaction procedures, related with unforeseen interactions in the environment. In a project, interaction plans should be implemented in PDCA cycles at each organizational level. These cycles should, on one hand, possess built-in mechanisms for a fast correction of actions, in response to various types of changes in the project environment, and, on the other hand, pave the way to improve the performance of tasks in a given project. While performing these tasks, it is important to remember about the following functionalities:

   a. Adjustment of the hierarchy of needs. The first reaction to the results of efficiency judgment should be an adequate modification of the hierarchy of needs in a given project.
   b. Variants of PDCA plans. On the basis of the already modified hierarchy of needs, possibly reasonable variants of PDCA interaction plans should be created.
   c. CBA for variants of PDCA plans. PDCA interaction plans should undergo cost/benefit analysis (in particular, the approximations of concepts related to SWOT, risk, co-risk, cost, and benefit should be specified for these plans).
   d. Selection of a PDCA plan. Cost/benefit analysis should help in the selection of an optimal PDCA interaction plan, that is to be launched.
   e. Implementation of the selected PDCA plan. Once PDCA interaction plans are launched, their implementation should be monitored in accordance with the previously accepted implementation conditions.

4. Efficiency management improvement. We assume that the complexity of dynamic changes in the environment is so great, that the entire project is implemented, **without having a full knowledge about its environment and changing aims**. Keeping this in mind, we should systematically improve an efficiency management system. At the same time, along with improving "efficiency management," one should not forget about all other aspects of this process (c-granule). The aspects described as part of the paragraph—"Establishment of current assumptions, context, and tool," are especially important. As part of these activities, it is recommended to pay particular attention to the following processes (c-granules):

   a. Performance judgment of efficiency management. Analogous to "efficiency judgment," the judgment of "efficiency management" is necessary to identify and prioritize potential areas of improvement. "Efficiency management" judgment is relatively easy and consists of finding any deviations from the expectations, that occur during the execution of PDCA interaction plans, which are generated and implemented as part of the "Efficiency treatment." Nevertheless, in practice, the implementation of PDCA interaction plans may proceed satisfactorily at the start, when suddenly, there appears a deviation which leads to a catastrophe. Therefore, even if the implementation of PDCA interaction plans goes smoothly, it is worth considering the possibility of introducing more subtle indicators.

b. Pareto analysis of the PDCA plans implementation. In the case of CSE projects, there usually appears numerous problems with the implementation of PDCA plans. In practice, it is a good idea to use the *Pareto principle*[5] in such situations. The principle is known in numerous forms, but in software engineering, the most popular is the one, discovered by Microsoft:

> Microsoft has learned that 80 percent of the errors and crashes in Windows and Office are caused by 20 percent of the entire pool of bugs detected, and that more than 50 percent of the headaches derive from a mere 1 percent of all flawed code[6]

c. Improvement of assumptions, context and tools. Within this area of activity, proposals of corrections to issues related to efficiency management, that need to be improved, are developed. The corrections are then tested and approved for implementation and next, they are introduced to an efficiency management system. It should be emphasized that all efforts undertaken in this area significantly depend on the regularity and acuteness of the activities, carried out as part of "Monitoring and review," and on the adequate quality of cooperation with all other areas of activities in "efficiency management." The quality of improvement in efficiency management is directly dependent on recording and verifying the conclusions, drawn from any problems and ineffective aspects, identified in the process of efficiency management.

5. Monitoring and review. These include typical activities, aimed at supporting the "efficiency management improvement." As a result of these activities, a register of problems and ineffective aspects, identified in the process of efficiency management, should be created.
6. Communication and consultation. In a sense, this point might sound like a truism, but taking into consideration the statistics of CSE projects' failures, the lack of proper communication (described in numerous research papers, quoted in Chap. 9) is often mentioned as the main reason behind these failures. In the case of such projects, it often turns out that near the end of their implementation, they constitute **a classical "death march" [545], due to the lack of on-time communication/consultation between project participants and adequate domain experts.**

---

[5] http://en.wikipedia.org/wiki/Pareto_principle.

[6] http://www.crn.com/news/security/18821726/microsofts-ceo-80-20, -rule-applies-to-bugs-not-just-features.htm.

The approach to efficiency management presented above could be treated as some kind of a generalization of numerous domain-specific management systems, including the following:

1. Quality management systems—e.g., ISO 9K.[7]
2. Energy management systems—e.g., ISO 50K.[8]
3. Environmental management systems—e.g., ISO 14K.[9]
4. Business Continuity Management (BCM) systems—e.g., ISO 22K.[10]
5. Guidance on Project Management systems—e.g., ISO 21K.[11]
6. Food safety management systems—e.g., ISO 22K.[12]
7. Distributed Application Platforms and Services.[13]
8. Big data management systems.[14]
9. Risk management systems—e.g., ISO 31K.[15]
10. Physical asset management systems—e.g., ISO 55K.[16]
11. Medical/health device communication standards—e.g., CEN ISO/IEEE 11K.[17]
12. Information security management systems—e.g., ISO/IEC 27K.[18]
13. IT service management systems—e.g., ISO/IEC 20K.[19]
14. Numerous other management systems and standards.

The examples above show that efficiency management may be used for the construction of universal management systems, that could be configured, using parameters specific to a given application domain. These specific parameters may "substitute" universal parameters, mainly as part of the point "Establishment of current assumption, context, and tools." Next, once the substitution is performed, the system can be used in the area of intended application. Such an approach also **facilitates the integration of various management domains (such as security, quality, resources, products) in a form of a single, integrated management system, which—in general—is more easily adaptable to the changing needs and scales of application than numerous, ineffectively co-operating subsystems (that are used in large organizations).**

[7] http://en.wikipedia.org/wiki/ISO_9000.
[8] http://en.wikipedia.org/wiki/ISO_50001.
[9] http://en.wikipedia.org/wiki/ISO_14000.
[10] http://www.bcmpedia.org/wiki/ISO22301.
[11] http://en.wikipedia.org/wiki/ISO_21500.
[12] http://en.wikipedia.org/wiki/ISO_22000.
[13] http://en.wikipedia.org/wiki/ISO/IEC_JTC_1/SC_38.
[14] www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1821.
[15] http://en.wikipedia.org/wiki/ISO_31000.
[16] http://en.wikipedia.org/wiki/PAS_55.
[17] http://en.wikipedia.org/wiki/ISO/IEEE_11073.
[18] http://en.wikipedia.org/wiki/ISO/IEC_27001:2005.
[19] http://en.wikipedia.org/wiki/ISO/IEC_20000.

It is worth emphasizing that the aforesaid configurability of an efficiency management system requires, above all, an efficient and user-friendly system for representing "wisdom" (that is c-granules, representing physical interactions, adaptive judgment mechanisms, and knowledge on data related to a given application domain).

The above remark may be treated as a starting point for underlining the significance of c-granules for the concept of an efficiency management system, presented in the book.

It may seem that in order to obtain a simple configurability of the aforesaid system, it is enough to specify its "parameters" in some formal language of modern mathematics. Theoretically, a system representing the ontology, knowledge, and mechanisms of adaptive judgment could be expressed, using such a formalization. However, in practice, this does not suffice.

Formalized languages of modern mathematics have their limitations, resulting from the static character of the language. For example, the static character of semiotic components (e.g., semantics, pragmatics, syntax) depends on the fact that these components do not undergo any changes, such as changes related to the necessity to adapt these components to encompass the results of a given interaction (both at the physical and granular level). A good illustration is the concept of "physical matter," which till the end of the 19th century, did not encompass the mass-energy equation (such an identification stems from the theory of relativity and $E = mc^2$ equation).

The above considerations show that the assumption of the open world (to new interactions, data, information, knowledge, and wisdom) requires accepting the dynamic and flexible character of the evolutionary development of language (including its semiotics).

In other words, an efficiency management system can be precisely described according to the currently predominant conventions. It means that for the sake of maintaining accuracy and precision, we should use the static language of modern mathematics. However, in this way, we impose the above-mentioned limitations (for example, limitations resulting from the closeness of the world, which is an important ontological paradigm of the language of modern mathematics, expressed in Cantor's set theory). Consequently, actual computations and their control—in an efficiency management system discussed in the book—would be carried out in a closed world. Such conceptual computations are based on a Turing machine.

However, once the components of an efficiency management system are described by means of c-granules and treated as c-granules, we may enter into an open world, described by the dynamically and semiotically flexible c-granules. In this case, any computations and control related to this very efficiency management system and performed upon these c-granules will be conducted in an open world, in which it is difficult to foresee some unexpected interactions with physical objects (hunks). Therefore, such computations constitute interactive granular computations and differ from the model based on a Turning machine.

In addition to the aforesaid benefits of using c-granules in an open world to describe efficiency management systems, another important positive consequence of using c-granules is their physical character, which was strongly emphasized by us when discussing the concept of a c-granules in the previous parts of the book. Taking into account the physical character of c-granules in the description of an efficiency management system indeed brings it closer to our practical needs.

The significance of c-granules and interactive granular computations makes it possible to unify the efficiency management control and to draw conclusions about the properties of these computations, in accordance with the mechanisms, described in the previous parts of the book.

Another important aspect of using c-granules, to express the efficiency management models is a great scalability of this approach to efficiency management modelling. C-granules enables us to model the phenomena related to the interactions (including management) both at the micro- and macro-scale level of the CSE project organization.

In a sense, **the role of the granular approach—using c-granules as understood by WisTech—in designing, implementing, and developing CSE projects may be compared to the significance of the object-oriented programming**[20] **during the 80 and 90s of the 20th century.**

At this point, let us recall that the "object-oriented approach" had been developed three decades earlier, that is between the 50 and 60s of the 20th century. The first, commonly-known environments that carried out research in this field were:

1. MIT, where in 1963, the adaptation of Algol was developed, known as AED-0[21] and
2. Norwegian Computing Center in Oslo, where in the 60s, a simulation programming language was developed—Simula 67.[22]

The difference between c-granules in WisTech and objects in object-oriented programming languages is that the definable entities in object-oriented programming languages are:

1. **Language of expression** (according to the above explanations of the semiotically stationary character of a formalized language). They describe the reality in a close world. C-granules, on the other hand, describe the reality in an open world and are semiotically dynamic.
2. **Model of mathematics**. The von Neumann universe[23] constitutes an example of such a stationary mathematical world. The objects processed by modern computers are connected to the physical world by means of an intended implementation of a programming language. C-granules, on the other hand, are ever-changing physical beings, that remain in continuous interactions with the often unpredictable environment. As a result, their implementation may be modified in a way, that is unforeseeable by human beings.

---

[20] http://en.wikipedia.org/wiki/Object-oriented_programming.

[21] http://en.wikipedia.org/wiki/ALGOL_X.

[22] http://en.wikipedia.org/wiki/Simula.

[23] https://en.wikipedia.org/wiki/Von_Neumann_universe.

33.7 "Parametrizalibility" of Efficiency Management Framework (EMF)

The approach presented in Sect. 33.6—the WisTech efficiency management approach to CSE— has a universal nature. Basically, the approach may be applied to all types of CSE projects. For this purpose, a project-specific "family of parameters," that is within the scope of activities specified in "Establishment of current assumptions, context, and tools," must be determined. It should be emphasized that the situation described here is much more complex than it is generally assumed in adaptive control systems (where adaptation is related with the vector of parameters, that are, for example, figures). In the situation discussed by us, **the "family of parameters" consists of a set of c-granules, which may represent ontologies, knowledge, and wisdom**. Developing such a "family of parameters" in a proper manner is usually a very challenging task. Therefore, in the case of CSE projects, we should use the approximations of c-granules that compose such a "family of parameters." In practice, this facilitates the initial configuration of the aforesaid approach, with respect to a specific application domain. Next, the approach makes it possible to adapt the iterative versions of an approximated "configuration" to further conclusions, drawn from the discovery of changing constraints and judgments, related to the implementation of a given project.

"Families of parameters" used to configure the techniques of project implementation are very complex in themselves. Therefore, for various application domains, it is advisable to develop and improve recommendations concerning the construction of such "families of parameters," characteristic of a given domain. It is especially important in relation to such aspects of the project implementation techniques, which have a decisive influence on the success or failure of a project in a given class of similar projects.

For example, **these aspects may represent a package of common problems, characteristic of a given class of projects, that need to be solved**. When solving these problems, valuable knowledge and practical experiences that decide for the success of a given project (in a given class of similar projects) may be represented. At the same time, ignoring the aforesaid knowledge and experiences in the implementation of a given project may have detrimental consequences for a given class of projects.

Chapter 34
Some Issues of WisTech Approach to CSE

34.1 Learning How to Perform Adaptive Judgment About Important Properties of Interactive Computations

The basic concept of WisTech, that is the concept of *judgment*, is treated in this book as an extension of the concept of reasoning. The concept of "reasoning" is used mainly to refer to the agent's control, that uses verified and repeated schemes for reconfiguring hunk configurations in the agent's memory (m-hunks). These schemes may apply to all types of reasoning. Therefore, these include such forms of reasoning as: deduction (e.g., in mathematics), induction (e.g., in physics), abduction (e.g., in medicine), and defeasible reasoning (e.g., in everyday life). Let us remind that by defeasible reasoning, roughly speaking, we mean reasoning the process of accepting an argument because of what normally happens, until one discovers it is not true.

Aggregations of these schemes—as intended by an agent—may constitute convincing (according to the agent's knowledge) justifications of theses, that arise from such a reasoning.

An intelligent agent, while performing the process of judgment, is practically always driven by her/his current priority of needs and takes into account the current needs of other agents, that are in any way related with her/him as far as the context of her/his situation is concerned. In other words, results of judgment depend on the current hierarchy of needs, current situation, and the context.

By identifying new needs, an agent may terminate, hinder, facilitate, or modify the process of judgment, performed by her/him.

An agent that implements a given interaction plan, in order to satisfy her/his urgent and top-priority need (e.g., helping an ambulance arrive to a hospital on time), often acts in a dynamically changing environment, under the pressure of time and risk. She/he is forced to make quick and almost automatic judgments about a given situation, on the basis of dynamically changing and incomplete information, that is available to her/him. Her/his—almost automatic—judgment and quick reactions are

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_34

not proceeded by detailed analyses and reasoning. Very often, in such a situation, the worst decision is the lack of any decision whatsoever.

In the situation presented above, an agent perceives various dynamically changing features of interactive computations in the surrounding world. Having an access to the incomplete knowledge about changes in an environment, an agent should be able to quickly grasp the **features of computations**, **that are important to her/him, in order to predict and influence the course of interactions according to her/his current needs**.

Frequently, in such situations, an agent is "condemned" to making simplified comparisons between "computations," that have been already identified and judged by her/him, by searching in her/his memory for analogical computations, that are "similar" to the "currently judged features of computations." On this basis, a judgment is being made. As an agent gains experiences, she/he may verify the justness of her/his judgment. If an agent makes a mistake, she/he may try to improve by taking into account some other features, that are already known to her/him. If this is not enough, an agent may create new features, which will help her/him to minimize the risk of a mistake in given circumstances. The quality of judgment significantly depends on the "experience of an agent," that is represented by observations recorded in her/his memory and her/his ability to judge the degree of similarity between these observations.

The process of learning how to perform adaptive judgment about important features of interactive computations, based on the mechanisms described above, resembles the decision-making techniques, known in AI and based on the k-nearest neighbors algorithm (k-NN)[1] [193], or Case Based Reasoning (CBR)[2] techniques [218].

Of course, in the case of WisTech, what we have in mind is the k-NN algorithm in the form of interactive computations, that examine the similarity between specific c-granules (particularly those, that represent the features of computations, observed and recorded by an agent) by means of other c-granules (e.g., those, that verify the similarity of c-granules and recommend making a particular decision).

Once a given interaction plan is implemented, an agent will probably have more time for making judgment and drawing conclusions from her/his experience. For this purpose, she/he may use another set of c-granules to verify, whether her/his judgments and decisions about a given interaction plan were right. If her/his judgments were not right, an agent should try to identify the reasons of her/his incorrect actions. This may require an agent to discover new *features* of computations, that will differentiate between various features, considered similar by an agent, and thus, prevent an agent from making similar mistakes in the future.

The correction and adaptation of the process of judgment may also consist of a proper modification of a c-granule, representing the relation of similarity between different c-granules.

It should be emphasized that our deliberations on the process of judgment, concerning features of computations, are mainly aimed at signalizing the "easiness" of an approach to modeling such processes in WisTech using interactive computations

[1] http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[2] http://en.wikipedia.org/wiki/Case-based_reasoning.

by c-granules. However, it is not our aim to engage in an in-depth discussion devoted to the concept of judgment.

Nevertheless, it is worth remembering that the discussion on various aspects of "judgment" has been present for ages, even in the oldest written documents of our civilization. What is more, how we understand the concept today is very much related to numerous important trends of discussions, dating back to antiquity.

Nowadays, the term is used as one of the main components of the decision-making processes. Therefore, it has a wide range of application. Particularly interesting examples, illustrating the mechanisms of judgment in the medical practice, can be found in the book by Kathryn Montgomery, entitled "How Doctors Think: Clinical Judgment and the Practice of Medicine" [349]. An in-depth analysis of the concept of practical judgment, as well as its contemporary meaning and role, may be found in the book by Leslie Paul, entitled "The Heart of Judgment Practical Wisdom, Neuroscience, and Narrative" [504]. The book includes, for instance, a very brief description of the concept and its contemporary meaning, embedded in the context of Aristotle's achievements (cf. Quotation 28.1.4).

In this context, the contemporary meaning of "judgment" is foregrounded in [504] (cf. Quotation 31.3). The description of practical judgment in this quotation and its contemporary meaning encompasses all key functionalities, which an intelligent agent should possess in order to simulate the processes, identified as practical judgment. In the first place, an agent must be able to use concepts that determine procedures and measures of importance, relative significance, values of relationships, and many other issues beyond the jurisdiction of reason. An agent, who is in a continuous interaction with her/his external and internal environment, recognizes various aspects of these environments.

For example, an agent who sees a wolf running towards her/him may become scared of it. Depending on the state of an agent, she/he may become paralyzed by fear or, quite the contrary, plan and undertake actions that will minimize the risk, connected with such an encounter. However, when a stress-resistant agent additionally spots a helicopter that is quietly, but quickly coming down to her/his direction (with its engine on), she/he will become less concerned about the wolf and instead, she/he will **shift her/his attention to action plans**, that can help her/him escape from the situation caused by helicoptetr. Such an agent may be so seasoned and stress-resistant player, that she/he will encourage the wolf to attack her/him in order to lure the wolf out to a place, where the helicopter is just about to crash. In other words, an agent performing the process of judgment should monitor the context, in which the process takes place on an ongoing basis. The context may pertain to phenomena in the agent's environment and/or phenomena related to her/his (e.g., emotions).

To describe such a context, an agent must be able to use certain concepts and relations between them (in our approach, these concepts and relations are represented by c-granules). These concepts may be referred to, in brief, as *judgmental concepts*. Intuitively, we may imagine them as a network of interrelated hierarchies that represent various concepts, involved in the process of judgment.

The intuitions behind judgmental concepts indicate that the concept itself encompasses both hierarchies of concepts, determining the context of the process of judgment (they may be intuitively treated as parameters of the judgment's control), and hierarchies of evaluations, that result from the processes of judgment.

For example, the hierarchies of evaluations include logical values (such as: truth, false), as well as various scales related with agent's needs, risk, benefits, and costs (cf. judgment scale in Fig. 32.2 and FPW-10). To illustrate, these may be a hierarchy of concepts, that describe the hierarchies of different types of losses, incurred by an agent, hierarchies of chances, related with taking advantage of the agent's susceptibility by various threats, as well as hierarchies of risks. The evaluation scales should be used in respective contexts, related to the environment and internal states of an agent (e.g., emotions, hunger, etc.). Undoubtedly, from the agent's point of view, hierarchies of concepts, describing the agent's hierarchies of needs that help her/him survive in a given environment, are especially important. They refer to the concept of the Maslow's hierarchy of needs.[3]

It is possible to illustrate the aforesaid approach towards modeling computations, which perform adaptive judgment about the features of computations, using a simplified example based on decision tables (cf. Sect. 28.2.3, Figs. 28.11, 28.12, and 28.14), as understood in the rough set approach.

For this purpose, let us assume that agent's c-granules are implemented by means of decision tables, that derive values of attributes from their surroundings and depending on the values derived, decision-making attributes are able to undertake some action. To generalize, we may imagine components of this process in the following way:

DT1. Interactions with the environment and their transmission by means of links, resulting in computing values of conditional attributes from a decision table.
DT2. Interpretations and judgment related to interaction results (e.g., by selecting rows with conditional attributes of a decision table, that satisfy the previously observed values).
DT3. Selection (which on itself constitutes judgment) of a postulated action from a decision table, along with its expected results (in this case, the results are represented by values of decision-making attributes from a decision table and possibly, while performing such a selection, we should be driven by the priorities, that stem from our current hierarchy of needs).
DT4. Initiation of a postulated action.
DT5. Actual execution of an action in the environment (which does not necessarily have to proceed in accordance with the attributes, that represent the expected results of these interactions) and the process of loading the actual results of these interactions.
DT6. Analysis and evaluation of disparities between the expected results of a postulated action and the actual results (e.g., by means of conditional attributes from a decision table, that partially support the evaluation of disparities between actual results and expected results of interactions).

[3] http://en.wikipedia.org/wiki/Maslow's_hierarchy_of_needs.

DT7. Depending on the results of the evaluation and knowing "right" actions that should have been initiated, an agent may recommend some corrective actions (e.g., by modifying conditional and decision-making attributes of a decision table).

DT8. Acquisition and representation of knowledge on modifying decision tables and discovering their new attributes, involved in DT1–DT7 components, from respective decision tables. The process of planning, implementing, and judging the results of experiments, followed by the implementation of new decision tables to DT1–DT7 components (including newly discovered attributes).

Depending on its ability to identify disparities between expectations and actual results, a decision table may lead an agent to recommend some corrective actions. In numerous practical applications, this approach, based on the use of a decision table, with easily computable attributes, is too simple.

To improve the results, the example may be modified and enhanced in numerous directions. For instance, the following modifications can be applied:

1. External modifications are aimed at expanding the structure of decision-making processes outside the tables. These include the following groups of modifications:

   i. Upward—modifications that expand the structures above a decision table. Intuitively speaking, instead of making a decision directly on the basis of decision-making attributes from decision tables (cf. the computational scheme above), we may make a decision on the basis of rules, automatically generated from the tables (or we may generate control classifiers, that will settle any possible conflicts between particular rules).
   ii. Downward—modifications that expand the structures, which are found inside a decision table (that is structures of conditional and decision-making attributes). In the example above, a decision table does not need to be simple and constructed out of easily computable conditional and decision-making attributes. Both conditional and decision-making attributes may be constructed on the basis of other decision tables. What is more, these recursive tables may also have attributes that were computed from other tables.
   iii. Lateral—modifications that expand the structures of network relations and principles, governing the functioning of a decision table, within a network of other tables or even within the network of other c-granules.

2. Internal modifications that expand only a decision table. For example, these can be done in the following two eways, called along and across.

   a. Along which is through adding new rows (e.g., an agent, along with her/his development, gains new experiences and records them in a form of new rows in a table).
   b. Across which is through discovering and adding new attributes (e.g., by using genetic algorithms or other techniques of generating new attributes and, in justified cases, by eliminating less useful attributes).

The example above, with computations performed by c-granules (implemented as decision tables), is based on a quite simple process. The process uses non-complex concepts close to sensory data processing, in order to choose an action that will enable an agent to satisfy her/his needs. Of course, one may also use a range of more advanced techniques based on rough sets (especially, various types of reducts, classifiers, and approximate reasoning). The example with a decision table should be treated as a particular case of a more general situation, in which instead of tables, there are other types of c-granules.

Moreover, it should be emphasized that reading the values of conditional attributes (or initiating an action by means of a decision-making attribute) requires implementing, in the case of more complex cases, c-granules that constitute links. In its most basic form, a link may be implemented by means of a hunk, which, in a repeatable manner, reacts to repeatable types of phenomena that occur in the environment. The reaction of a link—to repeatable types of phenomena occurring in the environment—consists of a repeatable modifications of its hunks. These modifications can be, in this very simple case, treated as the values of this attribute. For example, a thermometer attached to the wall may be treated as a sensor. Of course, it is an extremely simple situation. In general, a thermometer may be used as link in numerous situational contexts. What is more, used in a thermostat, it may even compute the values of an attribute, that corresponds to temperature. This intuitive example indicates that links are generally controlled. Even in the case of such a simple link related to a thermometer, that acts as a sensor, it is very easy to list a number of intuitive control parameters, that may be also represented by c-granules. For example, these may include several groups of control parameters, related to the following aspects of links' functioning:

1. LG – Link goal specification,
2. AC – Link access technique,
3. MT – Link measurement technique,
4. LC – Links configuration,
5. LS – Links sensing scope,
6. LA – Links acting type,
7. LI – Links interpretation type.
8. others.

Therefore, a link, understood in such a way and used by a c-granule, determines a technique for identifying entities that are the object of interest of a given c-granule (or, to be more general, of the agent's control that owns this c-granule); apart from that for arranging sensors in such a way, that through their interactions with the surroundings, input data can be retrieved and used to compute a value, determined by the link, which will be further interpreted and processed by a c-granule (or, in general, by an agent).

34.2 Cooperation of Agents and Cooperation of c-graunles

34.2.1 Some Roots of Wistech Approach to Language

According to the American Heritage Dictionary[4] the concept of language could be Used in the following contexts:

a. Communication of thoughts and feelings through a system of arbitrary signals, such as voice sounds, gestures, or written symbols.
b. Such a system including its rules for combining its components, such as words.
c. Such a system as used by a nation, people, or other distinct community etc.

The concept of language defined in such a way may be expressed in WisTech by substituting the phrase "thoughts and feeling" with "c-granule" and by assuming that the phrase "distinct community" may refer to both "distinct society of agents" and "distinct society of agent's c-granules."

Agent's functioning in the environment usually requires her/him to use a proper conceptual apparatus and language in order to express, process, and communicate concepts, available in a given language. A language for recognizing and communicating important (in a sense of the agent's hierarchy of needs) features perceived by c-granules of agents is especially significant. On this basis, an agent tries to identify and perform optimal (from the point of view of her/his current needs) actions. For this purpose, an agent has to use as optimal concepts as possible.

In the WisTech approach, this means that c-granules should be constructed in such a way as to maximize the efficiency of the agent's functioning (including the efficiency of c-granular application and processing). These concepts (that is c-granules in WisTech) constitute a unique compression of important (from the agent's perspective) aspects, related to changes in the environment. While making conclusions about any perceived properties of IGrC, an agent should have optimal tools and c-granular generators, as well as instruments for simulating (in the agent's memory and by means of hunks available in the environment) important development scenarios of particular situations, which could support her/him in predicting the course of interactions. Conclusions drawn from these processes are expressed by an agent in her/his own language and processed and/or communicated to other agents.

An agent who conveys some message about a complex situation (e.g., represented by a c-granule) to other agents does not need to communicate all details, but only some compressed information in the form of a previously accepted name, that describes a given situation. An agent assumes that other agents interpret the situation along the agreed lines (which are usually not precise) and according to their state of knowledge. Mechanisms related to how a language functions, also pertain to the internal communication between the "societies" of agent's c-granules, as part of interactive granular computations, performed by an agent (including adaptive judgment). Instead of recalling complex descriptions and knowledge about a given

[4]https://www.ahdictionary.com/.

concept, an agent may use their names, and process them. Of course, an agent does not have an endless "alphabet" to create "names" for all concepts (c-granules) that she/he is using. What is more, as part of WisTech, we may say that not all c-granules are semiotic (having specific semantics, syntax, and pragmatics).

Moreover, an agent does not have endless variants of all possible c-granules.

Taking into account these limitations, an agent who constructs new c-granules should be driven by criteria such as the efficiency of her/his actions. It is important for her/him to decide which one to be compromised between the ease of using c-granules (including their complexity and the length of their descriptions) and the properties of context in which they are used (e.g., frequency of use, importance in relation to the hierarchy of needs, understandability to addressees, and opacity to competitors). In a sense, we deal here with a metaphor of generating the most effective possible codes, known as Huffman coding.

Basically even in the ancient civilization, a linguistic study was originally motivated by the proper description of classical liturgical language or mythological language, (cf. e.g., the Sanskrit grammar by Pāṇini (4th century BCE), Tolkappiyam in Tamil, or the development of logic and rhetoric among Greeks). Beginning around the 4th century BCE, China also developed its own grammatical traditions, while Arabic and Hebrew grammars developed during the Middle Ages.

Numerous linguistic historians agree that the Indian grammarian called Pāṇini (approx. 520–460 BCE)[5] was one of the most important (if not the most important) creators of linguistics (in a sense that concerns us). He is best known for his formulation of 3,959 rules of Sanskrit morphology, which are used till this day. The Sanskrit grammar by Pāṇini is to a large extent systematized and technical. In this analytical approach, there appear such concepts as phoneme, morpheme, and root. Pāṇini rules provide a complete description of the Sanskrit morphology. In terms of accuracy, his grammar may be compared with modern mathematics. At the same time, he tries to arrive at "efficient" syntactic structures, which can convey, in a precise manner, important messages to adequately educated experts. As a result, for non-experts, the language has non-intuitive structure, resembling a contemporary "machine language."

The great and perhaps still underestimated achievement by Pāṇini was the development of rules of grammar for composing semantics from morphemic roots. Advanced logical rules and techniques developed by Pāṇini have a great influence on both ancient and contemporary linguistics.

The achievements made by Bhartṛihari, another key Indian linguist (approx. 450–510 CE),[6] are also very important. He developed a theory, according to which the act of speech consists of four stages:

1. Conceptualization of an idea.
2. Verbalization and sequencing.
3. Delivery of speech into physical world, such as atmospheric air, all performed by the speaker.

[5] https://en.wikipedia.org/wiki/Panini.

[6] http://en.wikipedia.org/wiki/Bhartrihari.

4. Interpretation of the speech by the listener—an interpreter who interprets the speech by negotiating its meaning and engaging in a dialogue with the speaker.

A very condensed summary of Bhartṛihari's views can be found in an article entitled "Bhartṛihari (c. 450–510 C.E.)", written by Stephanie Theodorou, The Internet Encyclopedia of Philosophy (ISSN 2161-0002)[7]:

> Bhartṛihari may be considered one of the most original philosophers of language and religion in ancient India. He is known primarily as a grammarian, but his works have great philosophical significance, especially with regard to the connections they posit between grammar, logic, semantics, and ontology. His thought may be characterized as part of the shabdadvaita (word monistic) school of thought, which asserts that cognition and language at an ultimate level are ontologically identical concepts that refer to one supreme reality, Brahman. Bhartṛiharii interprets the notion of the originary word (shabda) as transcending the bounds of spoken and written language and meaning. Understood as shabda tattva-the "word principle," this complex idea explains the nature of consciousness, the awareness of all forms of phenomenal appearances, and posits an identity obtains between these, which is none other than Brahman. It is thus language as a fundamentally ontological principle that accounts for how we are able to conceptualize and communicate the awareness of objects.

The works by Pāṇini and his followers (especially Bhartṛihari) had a considerable influence on a number of ideas, put forward by the Sanskrit lecturer, Ferdinand de Saussure, who is considered the father of the modern structural linguistics.

Of course, it is not our intention to present even a very limited overview of the history and current state of linguistic studies. Therefore, we will focus only on very selected issues, which are important for our research.

If we were to simplify, we could say that one of the basic criteria of evaluating the quality of artificial intelligence, developed by a man, is the ability to simulate human language behavior (cf. Turing Test).[8] Numerous contemporary researchers agree that the ability of automata to understand and convey messages included in written documents is critical for their further development.

In other words, one of the key problems of the contemporary times is to equip robots with reading and listening skills, as well as with the ability to comprehend natural language. Nowadays, we have numerous approaches towards comprehending natural language.

The approach presented in this book is based on granular computations. To some extent, it is based on our understanding of Wittgenstein's approach to natural language. The approach constitutes an essence of his Philiophical Investigations [539]. Wittgenstein's ideas continue to inspire numerous scientific discussions, aimed at enhancing our contemporary understanding of language and its relations with the processes of thinking. It should be emphasized that frequently, it turns out that even the closest disciples of Wittgenstein present quite disparate interpretations of his philosophical output in their papers. Therefore, when conducting one's own research, it is worth analyzing the original manuscripts of Wittgenstein and on this basis, try to understand various comments of those, who have continued the study on language,

---

[7] http://www.iep.utm.edu/bhartrihari/.

[8] http://en.wikipedia.org/wiki/Turing_test.

inspired by this well-known philosopher. In Wittgenstein's manuscripts, a crucial role is attached to the concepts of "social language game" and "private language," introduced in his Philosophical Investigations. Wittgenstein in Sect. 243 of his book, Philosophical Investigations, explained the meaning of the words of this language as follows [539]:

> The words of this language are to refer to what can be known only to the speaker; to his immediate, private, sensations. So another cannot understand the language.

To embrace the atmosphere of the discussion related to the concept of "social" language, introduced by Wittgenstein, we can read the following fragment of "Private Language" by Stewart Candlish and George Wrisley from the Stanford Encyclopedia of Philosophy (Summer 2012 Edition), Edward N. Zalta (ed.)[9]:

> [...] the community view is not easily reconciled with part of Wittgenstein's text, the matter is less clear than has been indicated so far. Significantly, even the most careful, insightful and sympathetic of Wittgenstein's commentators have divided on this matter (for example, Malcolm for the community view, and Baker and Hacker against it). The dispute is partly explained by the fact that the original texts (including some from Wittgenstein's manuscripts) seem to point two ways, some supporting the account given above (that the burden of the argument is that language must be potentially social), others the society view that language is essentially social. That is, textual support can be found for two apparently conflicting exegetical claims:
>
> 1. Language is essentially social.
> 2. It is conceptually (even if not psychologically) possible that a lifelong Crusoe (i.e., a human being isolated from birth) should employ some kind of linguistic system and follow rules in so doing.
>
> And the contending parties share the assumption that the conflict is genuine. There is, however, reason to believe that this assumption is false, for investigation of Wittgenstein's notions of essential, possible and lifelong Crusoe shows that admission of the first claim does not commit him to the denial of the second. To take the first notion: on Wittgenstein's view, while chess is essentially a game for two players, this does not exclude the possibility of playing it against oneself provided such solitary games are not regarded as paradigm instances of chess. Similarly, he can claim that language is essentially social, but still allow the possibility of exceptions provided these are peripheral cases.

Social language is available to an agent to an extent, to which she/he was able to learn it through social interactions. In other words, most often, and agent does not know hundred percent of this language. On the other hand, each agent has her/his own individual experiences, which may be related to concepts, that do not appear in the social language (or they might be related to concepts, that are unknown to an agent). Agent's survival in the environment largely depends on her/his ability to solve problems, based on the concepts that are available to her/him. For example, a proper understanding of the concept of energy (e.g., understanding that mass may be treated as a form of energy) enables an agent to create new sources of energy.

An agent may be in such a fortunate situation that she/he can be able to create new concepts on her/his own. Then, she/he will be able to create effective tools for

[9] http://plato.stanford.edu/archives/sum2012/entries/private-language.

solving problems in her/his environment. In this way, an agent may gain a competitive advantage over other agents around her/him. Metaphorically speaking, if someone discovers the relations between mass and energy, she/he may potentially gain an edge over others. This remark is true for all forms of human activities.

It is especially visible in building one's market position, discovering new drugs in medicine, and formulating new mathematical concepts. Those, who discover new concepts, name them in their private languages. Next, they propose using them in a social language. In practice, it often happens that the founders of some concepts create new names in their private languages for those concepts, that have been already discovered and named. The phenomenon is especially explicit when it comes to mathematical concepts. In order to explain this, let us assume that we have to solve the following problem:

> One pint of good wine costs 50 gold pieces, while one pint of poor wine costs 10. Two pints of wine are bought for 30 gold pieces. How much of each kind of wine was bought?

If we imagine Gauss, trying to find some solution to this problem, we will certainly think of the method described by him, whose key concept is the elimination of variables. Of course, Gauss could probably have not known that identical problem was investigated in ancient China. As early as in 200 BCE, the Chinese had devised similar method for solving systems of two linear equations with two unknowns. The method is illustrated in Chap. 7 of the Jiuzhang suanshu (*Nine Chapters in the Mathematical Art*).[10] This simple example illustrates two phenomena. The first one is that if we do not possess an appropriate conceptual apparatus (which, in this case, is necessary for solving two linear equations with two unknowns), we will be limited when it comes to the possibilities of providing solutions to practical problems.

The example above illustrates that the discovery of "new" concepts (also by mathematicians) does not necessarily mean that these are actually new concepts (and not concepts that have been known for 2,000 years, or even longer, and originated, for example, in ancient Mesopotamia).

At this point, it is worth noticing that an agent (especially a human being) has an access to her/his own private world of imagination. Therefore, she/he has very individualized experiences and, sometimes, an ill-conceived understanding of the concepts that originate from a social language. In a world of imagination and feelings, an agent can invent or introduce new concepts, only for the purpose of "talking to herself/himself."

For example, mathematicians, who discover new theories and proofs of theorems, act in such a way. What is more, usually, an agent holds a conversation with herself/himself for a significantly longer period of time than she/he "talks" to other agents. When "talking to herself/himself," an agent finds it easier to use and understand a given concept from her/his own private language. This is especially true for events that are known only to her/him (that is events that take place in the agent's mental or emotional sphere). For example, an infant, much earlier before it starts

[10] http://www-history.mcs.st-and.ac.uk/HistTopics/Nine_chapters.html.

using the word “mom,” creates in its imagination a model, which is named after the image of a person, whom the child believes to be its “mom.”

Here, it should be emphasized that the motivation behind using a private language is not only connected with names of concepts in someone’s imagination. This argumentation refers also to objects from the physical world. For a better understanding, let us imagine an agent who “sees” in a much broader spectrum of electromagnetic waves than the visible light. Such agent will be able to see phenomena that are unnoticeable by other agents, who do not have such abilities. Then, for a better survival of an agent, she/he must better understand the phenomena, that occur in the world, and be able to describe relations between them. In this way, an agent creates concepts, in her/his language, that are hardly understandable to other agents, who do not have these additional skills. It may also happen that the natural world, through its evolution, will “sharpen” the human vision so that we will be able to see subtle color differences. Eskimos are able to differentiate between white colors better than other people and they name these different shades of white more frequently (than, for example, people who live near equator) in order to solve problems that decide about their survival in a more effective way.

An analogical situation occurs when it comes to the internal world of an agent (a metaphorical Eskimo), which is unavailable to other agents (metaphorical people, who live near equator). Therefore, we may expect that an agent will use her/his private language for two out of four stages of the above-mentioned speech act (according to Bhartṛihari), that is the conceptualization of a given idea and its further verbalization and sequencing. In fact, an agent should use her/his private language for this purpose as she/he is then able to quickly change names of particular concepts, according to her/his needs, and dynamically create new concepts, that match her/his approach to a given problem (or even her/his dynamically created hierarchies of concepts).

34.2.2 Communication by C-granules and/or Agents

The roots of the WisTech approach to language and communication, presented in the previous sections (cf. Sect. 34.2.1), are very close to the basic assumptions of WisTech, included in the chapter “Framework Postulates for WisTech” (Chap. 32). These roots are directly reflected in FPW-15 (Sect. 32.4.6).

A number of important aspects related to communication (including communication in CSE and IGrC) were explained in more details, e.g., in Sects. 28.4 and 31.2. In these sections, the role of c-granules in relation to languages is emphasized. C-granules represent concepts as well as ways of expressing these concepts in a given language and ways of communicating these concepts to other c-granules and agents. Moreover, the communication between c-granules and agents is conducted by means of c-granules. In the case of communication between agents, these are the societies of c-granules which can be implemented by means of *communication channels*, as understood by Barwise and Seligman [26]. What is more, an agent may be involved in communication with the external world, and in with her/his internal world, that is between her/his c-granules, using those c-granules in which links are responsible for conveying particular messages.

We should also take into account an important principle, that is striving towards the efficiency of c-granular processing from the perspective of the most important tasks of an agent, described in FPW-14 (Sect. 32.4.5). This includes the attempts to minimize the size and complexity of c-granular representation, along with the maximization of positive results and *robustness* to any perturbations, and built-in errors, present in these c-granules. The increase in the efficiency of c-granular processing is mainly achieved by various forms of granulation (aggregation, decomposition, and compression with acceptable loss of less important details).

The efficiency of c-granular processing, performed by an agent and society of c-granules (in the case of a society of agents), translates directly into the efficiency and effectiveness of the agent's adaptive judgment, which is, in fact, a particular form of c-granular processing. A language provides an agent with an additional layer, that enables her/him to trace and perform judgments about interactive computations on the already existing c-granules. Analogous to a single agent, in the case of a society of agents and society c-granules, the communication between agents (based on language) provides agents with an additional layer, that enables them to trace and perform judgments about interactive computations on society c-granules.

Given the importance of adaptive judgment in performing the most crucial tasks of an agent, it may be concluded that language and communication are useful to such an extent, to which they help increase the efficiency of processes, related to adaptive judgment. It should be remembered that the processes of judgment are performed using both c-granules that represent processes of judgment and those that represent concepts, present in the agent's hierarchy of needs. Such c-granules are formed, for instance, as a result of self-organizing processes of c-granules into more perfect networks of hierarchical c-granules.

Self-organization[11] [59, 110, 130, 136, 161, 184, 287, 293, 311, 343, 365, 405, 420] may be treated as a specific case of *natural computing* [420]. The first natural computing model was developed by Ulam and Neumann in the 40 s of the 20th century and was known as the cellular automaton.[12]

The computational model of cellular automaton may be treated as a reference point to more general interactive granular computations, leading to a *self-organization* of a family of c-granules or a society of agents.

In order to illustrate the self-organization of a society of agents (or c-granules, respectively), let us assume that each agent (or c-granule respectively) from this society may receive, by means of her/his c-granules (or links of c-granules, which may be treated as c-granules), the results of interactions from her/his surroundings. This surroundings may be conventionally referred to as the agent's (c-granule's) communicational surroundings. In the agent's (or c-granule's) communicational surroundings, there are both "senders" and "receivers" of communicational signals. The richer the language of communication, the more complex the process of self-

[11] http://en.wikipedia.org/wiki/Self-organization.

[12] http://en.wikipedia.org/wiki/Cellular_automaton.

organization of a society of agents (or c-granules). Each agent (or c-granule) in a given society compares, on an ongoing basis, her/his current priority of needs (or conditions, *determined by the agent's control*) with the results of interactions in her/his communicational surroundings.

The needs (or conditions, determined by the agent's control), capacity to act, and control mechanisms of an agent determine certain constraints on reactions, permissible by an agent, as part of the process of self-organization. When repeating this process iteratively, interactions with other agents (or c-granules) and with the agent's (or c-granule's) environment of activity may be formed. The process may be stabilized if in a given society of agents (or c-granules), a new c-granule is created. When it comes to the societies of agents, hierarchies of needs of these societies, which often differ from the private needs of individual agents, are also formed. Society needs show that it is important to pursue not only one's private goals, but also certain global aims.

In cellular automata, a proper language is frequently used to specify conditions of constraints, that are to be satisfied in order to preserve particular cells. If these conditions are properly defined, the iterative process converges to a stable pattern of behavior, followed by these cells, which indicate a solution to the initial problem. The mechanism may be transferred into the process of self-organization in a society of agents (or c-granules).

The society of agents (or a family of c-granules), which was created as a result of iteration stabilization in the process of self-organization, may also be a meta-agent (or a c-granule). By repeating the process of stabilization in an inductive manner, we obtain hierarchies of agents (or c-granules). Such hierarchical structures may be built from the families of artificial agents and societies of experts. They were usefully applied in the approximation of complex concepts [5, 27, 363]. As part of these studies, new domains were developed, such as "cooperative problem solving"[13] or "collaborative intelligence".[14]

Some emerging patterns of behaviors, followed by very large, self-organizing societies of agents (or c-granules) with n-control parameters for determining initial positions, may, in simple terms, be represented as functions from an n-dimensional unit cube in a set of real numbers. We may say that an emerging pattern is represented through the indication of painting a point on an n-dimensional cube (which represents a vector of the agent's control parameters in a given point).

A similar situation occurs with a real hierarchical classifier, which assigns real values to each vector from an n-dimensional cube. If we assume that a problem, which concerns us, is represented by a continuous function, we may consider whether such a function could be presented as a simple aggregation of a one variable function and addition. It turns out that the problem is closely related to Hilbert's thirteenth problem, delivered in his Paris lecture in 1900.[15] In 1957, Kolmogorov presented a remarkable fact that any continuous function of many variables (from the Cartesian product of

---

[13] http://en.wikipedia.org/wiki/Cooperative_distributed_problem_solving.

[14] http://en.wikipedia.org/wiki/Collaborative_intelligence.

[15] http://en.wikipedia.org/wiki/Hilbert%27s_thirteenth_problem.

the unit interval [0, 1] into reals) can be represented by superpositions of continuous functions of one variable and addition [278]. One can interpret the superposition of the function in this problem as interactions, which support the communication (by intuition, these are the channels for the transmission of signals). Then, it can be inferred that in the case of interactive emergence of patterns,[16] represented by real continuous functions (of n-arguments), a given agent, computing the sum of all transmitted values, will need only single communication channels and single connections with agents, who compute and transmit the output value of the unary function. This also applies to hierarchical classifiers, which represent real continuous functions of n-variables. These classifiers have n real input signals and compute the real value at the output. Such a classifiers may be a composition of addition and one variable function. The relations from Kolmogorov's theorem were particularly examined in the case of classifiers, that constitute neural networks. For example, it was proved that the universal approximation theorem,[17] which states that a feed-forward networks with a single hidden layer, consisting of a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of R^n (under 'mild' assumptions on the activation function). In other words, the simple neural networks can represent a wide variety of interesting functions, when appropriate parameters are selected. The considerations devoted to neural networks may be transferred to c-granular networks.

34.2.3 Cooperation and Competition of C-granules and/or Agents

In general, the ability of a single agent to survive in a complex and changing environment is very low. However, the society of agents, that cooperate with each other on an adaptive basis, shows a radically higher chance of survival, both when it comes to the entire society and the majority of agents that compose the society. The phenomenon is illustrated, for instance, in Sect. 31.1, with an example (based on work [42]) that depicts the adaptive cooperation in a bacterial colony. The example emphasizes how important the mechanisms of cooperation between agents are in their struggle for existence. The mechanisms are also crucial in WisTech. In fact, if we were to simplify, we may say that the essence of WisTech consists in planning and implementing as effective mechanisms of cooperation as possible to increase the chance and possibility of effective execution of the most important tasks of cooperating agents. Therefore, virtually all key concepts of WisTech should help to improve the understanding and efficiency of cooperation among agents.

One group of WisTech postulates is dedicated to issues connected with "Agent Team Cooperation for Problem Solving" (FPW-16, Sect. 32.4.7). Of course, the remaining WisTech postulates are also very important when it comes to this. For

[16] http://en.wikipedia.org/wiki/Emergence.

[17] http://en.wikipedia.org/wiki.

example, it is difficult to imagine the cooperation of agents without any communication between them (FPW-15, Sect. 32.4.6) or creative cooperation of agents without understanding by them major (and common) their objectives, judged on the basis of their dynamically changing hierarchies of needs (FPW-14, Sect. 32.4.5). There is a slight chance that an agent, starting from scratch— that is the "tabula rasa" state of mind—is able to learn and understand, by herself/himself, her/his most important tasks, including tasks resulting from her/his potential role in a team of cooperating agents.

A potential mechanism of cooperation between agents may be understood as follows: agents from a family of agents create, by their behaviors, a new meta-agent, who has her/his own hierarchy of needs. Such a hierarchy is an aggregation of several hierarchies of needs, specific to particular agents, who comprise of a given society of cooperating agents. In such situations, mechanisms for settling conflicts between agents as well as concepts that enable agents, in the most efficient way possible, to understand and perform specific tasks, resulting from their changing role in a team of cooperating agents, are the most significant.

The illustration of a model, simulating interactive granular computations that implements the idea of cooperating and/or competing groups of agents, may be presented using a simulation of behaviors, displayed by automatic traders on financial markets. For this purpose, we may adopt the following assumptions:

1. Each agent is assigned to a specific *row of agent*, which is a natural number. For $n > 1$, *agents of* n *row* are *leaders* in *teams* of some $n - 1$ row agents. We assume that a number of agents in a team of n row is limited by a maximum preset number of participants in a team of n row.
2. Each agent may trade on a *real account* and/or a *virtual account*. Therefore, each agent has two portfolios of instruments, that constitute the object of trade: real and virtual. An agent *ag* of $n + 1$ row has a portfolio which is the sum of all portfolios of n-row agents for which *ag* is the leader. By trade results and a team's portfolio, we mean, respectively, trade results achieved by a team leader and her/his portfolio.
3. An n row agent may be assigned to a maximum of one $n + 1$ row agent. In such a situation, we may say that an $n + 1$ row agent is a *leader* of an n row agent and that an n row agent is a *subordinate* of an $n + 1$ row agent. An n row agent has no leader and therefore is referred to as an n row free agent.
4. A family of n row agents with its leader is referred to as an *n row team*.
5. $n + 1$ row agents may select players to their teams out of n-row free agents.
6. A 1st row agent may suggest to her/his leader performing certain actions of purchasing and/or selling assets on a real and/or virtual market (DEMO account). A 1st row agent may perform such an action only after obtaining her/his leader's acceptance.
7. A free agent of $n > 1$ row may make an independent decision about performing a particular action on a real and/or virtual market (DEMO account).

8. An agent of $n > 1$ row, which is not free, may suggest to her/his leaders performing an action by a team, that is subordinate to her/him, on a real and/or virtual market (DEMO account). An agent may perform such an action only after obtaining her/his leader's acceptance.
9. Each agent is equipped with the following lists:

   a. List of parameters for characterizing parameters (including portfolios) and trade results, achieved directly by her/his subordinate agents.
   b. List with sets of parameters for characterizing an agent. These may include such parameters as: current private risk and co-risk.
      i. Private assumptions of an agent related to needs that her/his current private risk and co-risk should satisfy during a given game (e.g., the risk lower than..., co-risk higher than... and maximization of co-risk at the level of risk and maintenance of loss at the level, which does not exceed...).
      ii. Private rules of an agent, related to changes in risk and co-risk during a game. For example, an agent, who has rich resources of some assets, may afford relaxing her/his risk requirements and increasing the level of expected co-risk. If an agent has poor resources, she/he may decide to avoid risky transactions.
      iii. Sets of game strategies at different levels of acceptable risk and co-risk.
      iv. Classification of agents in a team, from whom an agent would like to learn something in order to adapt her/his game strategies at different levels of risk. Lists of potential questions to other agents and mechanisms for aggregating answers (including settling conflicts in case of contradictory answers).
      v. Classification of agents in a team, from whom an agent would like to learn something in order to form a coalition with players within a given team (e.g., for protecting their games, as well as undertaking risk and co-risk together). Lists of potential messages to other agents within a given team (questions and answers) and mechanisms for aggregating answers (including settling conflicts in case of contradictory answers).
      vi. Current status and profile of the agent's portfolio (for example, a portfolio may include some financial instruments or assets, that are real or virtual).
   c. Four lists with classes of SWOT attributes (sensors and actuators), that is attributes which determine strengths and weaknesses of an agent as well as opportunities and threats during an analysis of a market situation and contextual attributes (attributes that describe a market situation),
   d. Adaptive judgment classifiers, which compute the values of a SWOT analysis (strengths and weaknesses of an agent as well as opportunities and threats).
   e. Adaptive judgment classifiers, which compute risk and co-risk of a current and changing market situation.
   f. Adaptive judgment classifiers, which compute sets of actions that are to be undertaken in response to risks and co-risks that have been identified.

g. Adaptive judgment classifiers related to the efficiency of potential interaction plans, that are to be launched and/or corrected by an agent.
h. Adaptive classifier for the CBA analysis and the selection of an interaction plan, that is to be implemented and/or corrected.
i. List of rule based classifiers with algorithm parameters for their adaptive reconstruction.

10. A leader of each team has classifiers for determining which coalition of players and individual players within her/his team are allowed to trade on a real and/or virtual market. The members of particular teams may suggest to their leaders the composition of a coalition between players, who work together to protect the positions that are being taken. A leader makes her/his decision on the basis of the SWOT and CBA analyses.
11. There is a special agent called the Market Broker Agent, who receives transaction orders and organizes their execution on both real and virtual account. The Market Broker Agent may be an interface to a stream of data from external sources (real data, historical data, Monte Carlo data) or participate in generating streams of data with the other societies of agents.
12. There is a special agent called the Public Teacher Agent, who represents knowledge resources shared by all agents, including data that is "publicly available" to the society of all agents. She/he has large sets of features, algorithms, including parameters used in algorithms, propositions of action plans, evaluation algorithms, and interpretations of various market phenomena. It can be assumed that the Public Teacher Agent obtains some of her/his wisdom from certain good players.

We assume that in the model presented above, each agent is equipped with c-granules that represent her/his knowledge and wisdom, related to situational judgment, and supports the selection of actions which are to be undertaken.

This simple outline of an IGrC model, presented above, is extremely flexible and can be elaborated on and modified in numerous ways. For example, we may assume that agents choose sets of parameters for important algorithms from the best (in their opinion) agents. Next, the sets of these parameters constitute an input population for selected metaheuristics (e.g., genetic algorithms), which compute optimal parameters for individual purposes of agents. There are numerous such mechanisms and therefore, it would seem natural to construct agents, based on the metaphor of genetic algorithms.

Moreover, we may assume that the construction of an agent's structure is encoded in her/his genetic "DNA." Then, agents could exchange parts of their "DNAs" with each other or parts of their DNAs could undergo random mutations. In this way, new agents could emerge as part of the evaluation, gradually superseding the population of weak agents.

It is worth noting that the example above has a quite universal character, which illustrates the use of mechanisms for cooperation and/or competition between *traders*, implemented by means of WisTech computational models. In this example, agents try to gain as much resources, whose values are measured by financial ratios, as possible, having a given range of possible basic interactions at their disposal. However, by treating these financial ratios and interactions with the market as a metaphor of the fight for resources in a partially known environment, we may expand this example to a much greater area of application. The system for supporting leadership in a fire and rescue operation, discussed in the following section (Sect. 35.3), constitutes an example.

34.2.4 Agent Language and Communication for Cooperation

34.2.4.1 Agent's Private Language and Infogranules

We assume that each c-granule is assigned a *name of a c-granule* and a *type of a c-granule* (referred to, respectively, as "G name" and "G type"). In our approach, names and types of c-granules are also c-granules. Frequently, numerous c-granules have the same type and there are also situations, in which several c-granules have the same name. Among names and types of c-granules, there can exist some relations, which are also c-granules. By relations, we also understand c-granules that compute the indicator function of these relations.[18]

It should be emphasized that c-granules process "input" c-granules, that have recognizable and acceptable types of input c-granules. For example, a c-granule that matches the specification of a task "blue painting for rooms in building" may be used in reference to c-granules, that represent such concepts as "a room in a building." However, there is usually no point in using such a c-granule in relation to other types. For example, **it does not make sense to paint a set of numbers less than 100 blue. This remarks illustrates how significant the concept of G-type is in our approach to IGrC**. To be more precise, c-granules may be aggregated and used only if they satisfy certain conditions.

In practice, the *subtype relation* is very important when it comes to c-granules, provided that we assume that it constitutes a partial order. It is sometimes assumed that the relation between types is represented by a more general mathematical structure. For example, it may be assumed that the relation between types is represented by a category [39, 296, 318, 329]. Regardless of whether the c-granule subtype relation is a partial order or a category, it constitutes a transitive relation. Hence, a c-granule may usually have numerous different types. In such situations, it is assumed that each c-granule has its smallest type, that is referred to as the *main type of a c-granule*.

It is convenient to assume that the name of a c-granule is unambiguous within its main type. The agent's control may have a mechanism for selecting c-granules with identical names (or of identical types). For example, the outcome of such a selection may arise from the context in which particular names of c-granules were used and from the scope of available information about a given c-granule. The control is able to perceive the subtype relation, for example by means of m-interactions and m-hunks.

[18] http://en.wikipedia.org/wiki/Indicator_function.

Names and types of c-granules help the agent's control in the processing of c-granules. Before starting the processing, the control first verifies types of c-granules in order to determine whether they qualify for a given type of processing. However, the use of names, instead of entire c-granules, significantly increases the efficiency of the whole process. It stems from the fact that in numerous situations, the agent's control may use only the name of a given c-granule or its other feature, instead of performing—frequently expensive—operations, related to the retrieval and processing of the entire c-granule.

What is more, it should be underlined that in practice, it is important to have an access to a given c-granule through its name, which can be used as a sort of indicator of the memory, available to an agent, or to a potential place, where a given form of a c-granule is archived and stored.

It is convenient to assume that an agent has some special c-granules for operating on c-granules and their names. For example, these may include such operations as: assigning a name, changing a name, establishing a name, retrieving a c-granule of a given name, or determining some feature based on the familiarity of a given c-granule. Names, types, and other features of c-granules may be computed by means of adequate decomposition operators. It should be remembered that both an agent and her/his c-granules are also placed in some context, which has frequently a great influence on the course of interactions that are executed. Therefore, it is important for the agent's control to be able to properly recognize and understand the *current context of the agent's activity*, which includes such factors as:

a. Current important (according to the agent's judgment) results of perception in the agent's environment of activity.
b. Important (according to the agent's judgment) results of perception (from the past), associated and recorded by the agent's control, concerning similar environments of the agent's activity.
c. Current priority needs of an agent (including needs that result from interaction plans, executed by an agent).
d. Important (according to the agent's judgment) possibilities of an agent to launch and/or modify interaction plans.
e. Current results of the SWOT analysis.

From the perspective of an agent, who pursues some goals and uses a given c-granule G, a c-granule, referred to as the *contextual interpretation of the c-granule* G, is extremely important. This c-granule assigns prioritized (according to the needs that result from the current context in which the c-granule is used) links to the c-granule G and contextual judgment. These links indicate examples that demonstrate to the agent's control such aspects of the c-granule G's meaning as given below.

1. Patterns of c-granular instances, judged in situations (in line with the context, in which a given c-granule was used), and development scenarios of particular situations, along with indicating potentially important results of these scenarios (both when it comes to increasing and decreasing the satisfaction of an agent).

2. Patterns of interaction plans, determining potentially important ways of using a given c-granule (or its components) and other resources, available to an agent in the current context, which enable her/him to fulfill her/his aims.
3. M-programs (Sect. 28.1.3) which support the agent's control in the processing of m-hunks, present in the agent's a-mirror (Sect. 28.1.3), in order to interpret a given c-granule in the current context. These may include various types of m-classifiers and other m-programs, which represent agent's domain knowledge, characteristic of a given c-granule, as well as its specificity and the current context of use by the agent's control.
4. Alternative names, along with their context of use and ways of archiving.
5. Plans of interactions, leading to the development and adaptation of a c-granule in its current form, along with its current or emergent contextual interpretation, through proper interactions.

Let us distinguish a c-granule SG, which is an aggregation of c-granules with two c-granules at the input:

- $G1$ – a c-granule with contextual interpretation,
- $G2$ – a contextual interpretation of $G1$.

However, at the output, there is an integrated c-granule $SG(G1, G2)$, with two c-granules at the input: $G1$ and $G2$. The c-granule SG is referred to as the *aggregation of a c-granule with its interpretation* and the c-granule $SG(G1, G2)$, which is the result of this aggregation, is referred to as a *semiotic c-granule*. The name stems from the fact that such a c-granule, formed as a result of aggregation, encompasses basic aspects that are part of the semiotic study.[19] Any semiotic c-granule represent syntax, semantics, *pragmatics*, and broadly-understood relations, which determine references to the examples of a given concept, represented by a c-granule, its context of appearance, and ways of use in specific contexts, along with the resulting, practical meaning.

At this point, it is worth recalling the meaning of two concepts: *pragmatics* and *practicism*, which is related to the former one in a practical aspect. By pragmatics, we understand such a field of semiotics, which studies the relations between a symbol, its linguistic expression, user, and a broadly-understood influence of the context, in which a given symbol is used, on its meaning. This is illustrated by a well-known quote, which is cited in the very first lines of an article, taken from the Stanford Encyclopedia of Philosophy, which constitutes an introduction to pragmatics[20]:

> When a diplomat says yes, he means 'perhaps';
> When he says perhaps, he means 'no';
> When he says no, he is not a diplomat.

In modern times, Charles Sanders Peirce noticed that the concept of "meaning" does not indicate a binary relation between objects and their properties. He stated that in this relation, a huge role is played by a situational context, in which a given

[19] http://en.wikipedia.org/wiki/Semiotics.

[20] http://plato.stanford.edu/entries/pragmatics/.

sentence is uttered (or in which a given message, related to this sentence, is conveyed). In this way, he arrived at a concept which is known today as the "Peirce's semiotic triad".[21] It should be emphasized that Peirce was particularly interested in practical aspects related to the context of meaning of particular expressions and on this basis, originated a trend in philosophy and logic, known as pragmatism.[22]

The above digression, aimed at recalling the concept of pragmatics and practicism, illustrates a great potential role of semiotic c-granules, a concept introduced in this section.

A class of all semiotic c-granules, used by an agent, is referred to as a *private language of an agent*. Of course, a **private language of an agent evolves with time**.

Analogous to the rules of grammar, used in a natural language, in the case of a private language of an agent, we may assume that an agent has a distinguished class of aggregations, which aggregate semiotic c-granules so that a new semiotic c-granules is created as a result. Such aggregations are called *semiotic aggregations*. Semiotic aggregations are process specific semiotic c-granules. Often, they constitute more general *schemes of semiotic aggregations*. These schemes operate on types of semiotic c-granules, rather than on specific c-granules. However, after substituting them with c-granules that have acceptable types at the input, a new integrated semiotic c-granule may be formed.

Schemes of semiotic aggregations may be extremely complex. For example, we may imagine that an agent has a camera with sensors, that register single pixels. Each pixel may then be treated as a pair, composed of a c-granule and its contextual interpretation. On the other hand, the aggregation of all these c-granules into a contextual interpretation, comprehensible to an agent, is usually a highly non-trivial task. Luckily, we also have numerous very simple examples of such schemes, which illustrate the intuition behind this concept.

Some examples include Boolean operators, which connect sentences by means of logical connectives, such as conjunction, disjunction, implication, or negation, and modal operators. In other words, if we have two sentences, p and q, at the input, along with their interpretations, we can easily construct a sentence (p or q), along with its interpretation, by using a scheme of semiotic aggregation for p and q. This example leads to another very important concept. In the example, we used a very simple scheme of a semiotic aggregation.

Most often, an agent has a distinguished class of schemes for selected types of c-granules in her/his private language. These schemes of semiotic aggregations are usually in the form of general and repeatable computational schemes. Such a class of c-granular aggregation is referred to as *the elementary grammar of the agent's private language*. For example, the elementary grammar in the case of the Boolean logic of a given mathematical theorem is based on the schemes in which mathematical sentences of this theorem are combined by means of Boolean operators.

---

[21] http://plato.stanford.edu/entries/peirce-semiotics/,http://visual-memory.co.uk/daniel/Documents/S4B/sem02.html.

[22] http://plato.stanford.edu/entries/pragmatism/.

C-granules which belong to the private language of an agent, irreducible in terms of its elementary grammar, are called *elementary c-granules* (cf. Sect. 28.4.1.4). Typical examples of elementary c-granules in IT systems include c-granules which correspond to descriptors [463]. According to the definition in Sect. 28.4.1.4 hierarchical aggregations produced by elementary grammar of the agent's private language are called *infogranules*.

Let us note that the major task of an agent—FPW-14, described in Sect. 32.4.1—determines the aims and roles which a private language of an agent should fulfill. In simple terms, it is about equipping an agent with effective tools, that could support the survival of c-granules, so that an agent is able to execute her/his major task in an optimal way. Very often, in practice, it means that the process of c-granular processing should lead to an efficient construction of interaction plans, aimed at:

- identifying and decomposing prioritized problems, which hinder the execution of the major task, and
- providing solutions and aggregations of solutions to the previously decomposed prioritized problems.

Chapter 35
A WisTech Approach to CSE

35.1 Main Objectives for WisTech Approach to Frameworks for CSE

In Part III of the book, we discussed problems that stem from differences between the theory and practice of CSE. The collection of these problems were referred to, in brief, as TPGP (Chap. 11). In Chap. 12, we presented a number of common *issues*, which contribute to TPGP. We discussed, e.g., some common causes of TPGP (Sect. 12.1). Among these causes, the so-called "Fundamental Precause of CSE Crisis" (FP3C) (cf. Sect. 1.2.2) seems to be especially important (cf. Chap. 13).

Of course, when trying to develop effective techniques for counteracting the most negative consequences of TPGP in projects like CSE, it is important to account for, and understand the causes, which form the FP3C.

In Part III of the book, we use certain examples to explain the level of difficulty with managing the FP3C in CSE projects. We focus on causes, which have their roots in contemporary barriers to the development of modern AI technologies. For this purpose, we use (Sect. 13.2) such examples of contemporary technological barriers as:

1. Granulation (Sect. 13.2.1).
2. Distributed computing (Sect. 13.2.2).
3. Interactions and adaptive strategies (Sect. 31.1).
4. Adaptive reasoning in IGrC, induction of local models, their aggregation and aggregation of aggregations (Sect. 13.2.4).

Each of these barriers is in itself a great challenge for the modern civilization. It should be emphasized that there are many more causes, which affect the level of difficulty with the FP3C, and they could be discussed from numerous perspectives.

Probably the most easy to explain and, at the same time, one of the oldest perspectives of looking at the causes of TPGP is related to the consequences of the Brook's observation, which says that "the complexity of software is an essential property" [68] (cf. Sect. 12.2).

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_35

Taking into account the level of difficulty with TPGP presented in this book, it is not a big surprise that we do not know any satisfactory techniques of dealing with the FP3C, let alone the fact that none of these techniques have ever been verified in practice.

The essence of our approach, to the aforesaid problem, is aimed at reducing the level of difficulty, associated with constructing a model for the implementation of complex CSE projects. In this approach, CSE projects will be treated as models of IGrC, which are in the line with the FPW postulates, included in Chap. 32. This means that such projects should be treated, according to our approach, as adaptive complex systems, which are the aggregations of numerous changing in time local models (c-granules) and methods of reasoning about them, represented by c-granules, selected on the basis of a **wise judgment**.

It should be emphasized that the solutions proposed in this book constitute only the first step toweard combating the problem. What is more, this road departs, by assumption, from the ontological foundations of knowledge based on modern classical mathematics (judged in the classical predicate calculus, expressed in Cantor's ontology). Therefore, at the current stage, the gloomy prognosis made by Brooks in his paper, entitled "No Silver Bullet—Essence and Accidents of Software Engineering" [68], are still valid:

1. [...] *there is no single development, in either technology or management technique, which by itself promises even one order of magnitude [tenfold] improvement within a decade in productivity, in reliability, in simplicity.*
2. [...] *we cannot expect ever to see two-fold gains every two years in software development, like there is in hardware development (Moore's law).*

To recapitulate our above discussion and its broader context, discussed in Part III of the book, at the current stage, the application of WisTech in CSE is not meant to provide a complete solution to the problem of TPGP. The actual objectives were presented in *WisTech for 'Silver Bullet' in Searching for the TPGP Approximate Solutions* (Chap. 15). They mainly concern applying WisTech to *efficiently* manage the generation and application of "optimal" approximations of c-granules, used in CSE.

These approximations should—at each stage of implementing a CSE project—constitute a possibly effective compromise between the costs (including risks) of their generation and application as well as the benefits (including co-risks) related to their degree of accuracy.

Therefore, one of the key aspects of our approach to using WisTech in CSE is an idea presented in the previous chapters, **related to supporting the process of efficiency management in terms of generating, processing, and adapting c-granules used by stakeholders, project participants, and artificial agents, who implement CSE projects**. An outline of such an idea is presented in the next section of the book.

35.2 CSE Projects for the Generation of New AI Technologies

In this section, we present our idea of the approach towards particularly important aspects of the projects, of which aim is to generate new technological solutions related to the AI application. Our approach constitutes a very condensed summary of experiences, gained in numerous projects of this type. Some of them were described in Part IV of the book, in Chaps. 17, 18, 19 and 20.

Our solution will be referred to, in short, as the *Crocos framework*. It is a detailed version of the efficiency management system in CSE projects, presented in Chap. 33. The components of the efficiency management process are shown in Fig. 33.5. The special new components, representing c-granules, that are specific to projects, are aimed at generating new technological solutions related to the AI application, have been added in Fig. 35.1. Also the meaning of and the most important relations between basic c-granules present in the Crocos framework, are illustrated in Fig. 35.1.

The Crocos framework consists, to a large extent, of implementation of a project, based on a well-balanced (in terms of efficiency) and close cooperation (carried out by means of interactions) between simultaneously conducted eight complex processes (c-granules that are aggregations of numerous other c-granules), including the following:

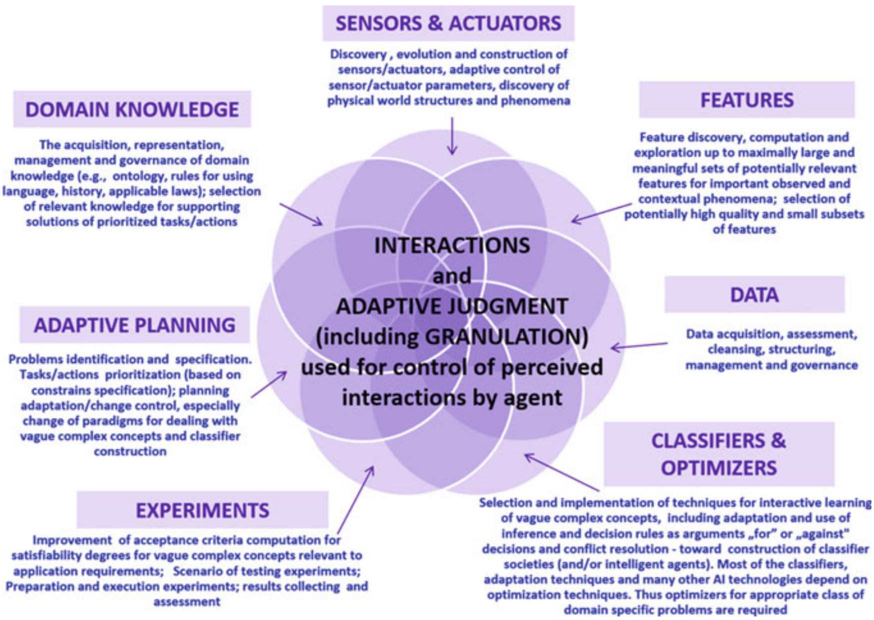


Fig. 35.1 A scheme illustrating the cooperation between c-granules, responsible for particular areas of competencies, as part of the Crocos framework

1. INTERACTIONS & ADAPTIVE JUDGMENT (including GRANULATION).

Functions: Mainly used for controlling and integration of perceived interactions.
Remarks: In terms of control, this is the most significant Crocos c-granule. It is responsible for the integration of all processes and adaptive judgment of the results of interactions between other c-granules, as well as for a broadly-understood granulation, that is a transformation of c-granules so that they can serve other c-granules in an optimal way.

2. SENSORS & ACTUATORS.

Functions: Discovery, evolution and construction of sensors/actuators, adaptive control of sensor/actuator parameters, and discovery of physical world structures and phenomena.
Remarks: The development of each new AI technology requires an access to good-quality data and experiments with these data, taking into account various possible applications of a given technology. The quality of these data, tests, and technology after its implementation depends on the quality of c-granules, that correspond to sensors and actuators.

3. FEATURES.

Functions: Feature discovery, computation and exploration up to maximally large and meaningful sets of potentially relevant features for important observed and contextual phenomena; selection of potentially high quality and small subsets of features.
Remarks: In a sense, all AI technologies can be reduced to the processing of concepts (understood as c-granules). The majority of these concepts is so complex, that it will never be described by means of attributes (more generally, c-granules), available to scientists and computers. Therefore, we are doomed to the approximations of these concepts, by discovering increasingly better features. In this context, the discovery and proper use of adequate features, adjusted to the specificity of application of a given AI technology, constitutes one of the most difficult phases, leading to the creation of new technologies.

4. DATA.

Functions: Data acquisition, assessment, cleansing, structuring, management and governance.
Remarks: Instead of discussing the role of data in AI at this point, we will quote two outstanding experts from the 20th century, the co-creators of the modern quality management and thus, the co-creators of the “PDCA cycle,” which is extensively used in WisTech:

i. E. Deming[1]:

In God we trust, all others bring data.

ii. W. Shewhart[2]:

Data has no meaning apart from its context.

It should be emphasized that **data, meaning of which is not understood, are usually worthless**. As part of the Crocos framework, the meaning of data is mostly determined by two c-granules: “DOMAIN KNOWLEDGE” and “EXPERIMENTS.”

5. CLASSIFIERS & OPTIMIZERS.

Functions: Selection and implementation of techniques for the interactive learning of vague complex concepts, including the adaptation and use of inference and decision rules, such as arguments “for” and “against” particular decisions, and resolution of conflicts,—aimed at the construction of the societies of classifiers (and/or intelligent agents). A great number of classifiers, adaptation techniques, and many other AI technologies depend on optimization techniques. Thus, optimization designed for an appropriate class of domain-specific problems is required.

Remarks: When developing any new technology, numerous different classifiers and optimizers are needed at each stage of its development.

It should be emphasized that both in the case of classifiers and optimizers, there is a significant difficulty in the fact that **in practical application, both a decision, determined by a classifier and an optimized concept are complex and vague. Therefore, it is important to use increasingly perfect approximations of these concepts**.

In general, these approximations are represented using measures, that reflect the intuition behind the degree of approximation in the case of a given concept. These measures are obtained as part of the “FEATURES” functionality of c-granules. New strategies for improving and optimizing classifiers are constructed and developed as part of a c-granule, referred to as “CLASSIFIERS & OPTIMIZERS.”

In common applications, well-known basic classifiers and optimizers are taken into account in the first place.

However, later, the domain knowledge, related to the specificity of a given technology that is being developed, is gradually better represented.

In practice, it is difficult to choose the best classifiers and optimizers. Therefore, it is important to skillfully plan, implement, and analyze the results of experiments in order to design as good hybridization of these results as possible for a given domain of application.

[1]http://en.wikipedia.org/wiki/W._Edwards_Deming.

[2]http://en.wikipedia.org/wiki/Walter_A._Shewhart.

6. EXPERIMENTS.

Functions: Improvement of assessment and acceptance criteria to determine the satisfiability degree of particular vague complex concepts, relevant to the requirements imposed on a given application in an experiment; Scenario of testing experiments; Preparation and execution of experiments; Collection of results and their assessment.

Remarks: By experiments, we understand here the generalization of the concept of testing. Recommended techniques for the implementation of plans, both as part of the Crocos framework and superordinate framework for efficiency management, consist of the execution of all non-trivial tasks in a PDCA cycle.

7. ADAPTIVE PLANNING

Functions: Identification and specification of problems. Prioritization of tasks and actions (based on the specification of constrains); planning adaptation/change control, especially change of paradigms for dealing with vague complex concepts and classifier construction.

Remarks: We should emphasize the adaptability of the planning process, which should encompass important conclusions, drawn on the basis of the activities in all other c-granules of the Crocos framework.

8. DOMAIN KNOWLEDGE.

Functions: The acquisition, representation, management, and governance of the domain knowledge (e.g., ontology, rules for using language, history, applicable laws); selection of relevant knowledge to support solving prioritized tasks/actions.

Remarks: This is a critical c-granule which should ensure a proper interpretation of the results of activities, performed by other c-granules, as well as support solving difficult and important problems, which are beyond the capabilities of other c-granules in the Crocos framework.

It is worth emphasizing that each of the eight c-granules, described above, constitutes an aggregation of numerous other c-granules. What is more, one should remember that when speaking about the generation of c-granules in terms of eight main c-granules, described above, we also mean the generation of their physical parts. For example, GRANULATION does not mean some abstract operations on abstract entities. This is an operation that leads to the formation of new hunks, which represent c-granules that were previously created.

Aspects of the Crocos framework, presented above, have a very flexible nature and therefore, can be used in numerous different projects devoted to the development of new AI technologies and their application.

35.3 C-granular Fire and Rescue Operations: A WisTech Approach to an Intelligent Commander's Remote Control

In the previous sections (Sects. 33.4, 33.5, and 34.2.3), we discussed some examples of WisTech concepts and mechanisms, which referred to the design and construction of a system that could potentially support a commander of a firefighting brigade during a fire and rescue operation.

Basically, the essence of our approach is in the specification of the following c-granules, which will be used to form PDCA cycles for such a system:

1. C-granules that classify situations and contexts.
2. C-granules that support the current SWOT analysis for given situations, scenes, and their contexts (cf. Sect. 33.4).
3. C-granules that support the current analysis and estimation of risk and co-risk for given situations, scenes, and their contexts (cf. Sect. 33.5).
4. C-granules that support the current selection of action variants for given situations, scenes, and their contexts (cf. Sect. 33.5).
5. C-granules that support the CBA analysis for variants of interaction plans for given situations, scenes, and their contexts (cf. Sect. 33.5).
6. C-granules that support the improvement of a system, particularly the improvement of the currently used c-granules, and which generate simulation scenarios to verify new solutions (cf. Sect. 33.5).

The model presented above is partially implemented in the ICRA [286] project.[3]

[3]http://www.icra-project.org/?node=10.

Chapter 36
Conclusions and Future Research

36.1 Conclusions

Practical conclusions from the implementation of numerous CSE projects (cf. Part III and Part IV of this book) indicate that there still is a very large gap between theory and practice (cf. TPGP in Chap. 11).

Among common causes of TPGP, the so-called “Fundamental Precause of CSE Crisis” (FP3C) (cf. Sects. 1.2.2 and 35.1) is especially important.

Providing solutions to the problems, that constitute FP3C requires solving numerous different partial problems. To illustrate the intuition behind one of such numerous partial problems, let us illustrate some intuitions behind the complexity of FP3C. For this purpose, we may use a paraphrase of a well-known quotation by F. Brooks [68] (cf. Sect. 13.2).

In the approach towards CSE projects, proposed in this book, we represent the implementation of innovative CSE projects as a metaphor of interactive granular computations that are in the line of the FPW postulates, presented in Chap. 32. In order to better illustrate the intuition behind this approach, we may simplify it by treating interactive granular computation as the generalization of the concept of computations, performed on traditional computers. In such a simplification, the metaphor of treating CSE projects as computations performed on “computers” may be presented as follows:

cse-1. Metaphor of a computer. In our metaphor, “computer” is represented by the dynamically changing “configurations” of people (including participants that share information with each other and process knowledge), who participate in interactions that should enable them to cooperate with each other, with the support of a proper infrastructure and tools from the environment.
The configurations of these teams and interactions between them are systematically modified and adjusted for the execution of the changing tasks. There are numerous possible variants of these configurations.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5_36

For example, in the case of a project in which 30 people are engaged, we may, theoretically configure about 1 billion different subteams. With a project conducted by 300 people, the number of potential teams is larger than the number of atoms in the observable universe. Of course, on one hand, not all teams are used, on the other the composition of consecutive teams depends, e.g., on numerous dynamic factors like competencies of the project participants, timely completion of tasks, and availability of resources for constructing new teams.

Huge numbers of potential teams and complex relations between their tasks, reflect the level of difficulty with selecting proper constraints on algorithms to optimize the project's implementation efficiency.

Any mistakes in the configuration of teams can be minimized by a proper execution of tasks assigned to a given team and a proper execution of tasks assigned to the other teams, which is ill-configured.

cse-2. *Metaphor of an operating system*. "Operating system" is mainly represented by: the language of communication, work culture, knowledge, skills, work motivation mechanisms, and operating languages that constitute the infrastructure, which supports project teams.

cse-3. *Metaphor of an input*. "Input" is represented by a network of interrelated hierarchies of needs, specific to a given project, relations between these hierarchies, constraints, and available resources.

Relations that determine some conditions in a form of constraints, e.g., invariants that should be fulfilled during "computations," are extremely important. Material resources that are not part of a "computer," but which result from the needs of a given project, also constitute an input to the "computations" performed as part of a project. For example, these may include: materials, tools, infrastructure, regulations, and other resources that ensure a proper initiation and implementation of a project.

However, in a typical CSE project, input hierarchies of needs determine much more than only the demand for material resources. They determine aims, techniques of approximation and processing key complex concepts (especially those that are included in the specification of a project), acceptance criteria, rules concerning the implementation of a project, HR motivational mechanisms, and rules for sharing, using, and reconciling resources. The key concepts present in such hierarchies are very complex and vague. In typical CSE projects, concepts that constitute an "input" are usually so complex, that no single project participant is able to comprehend their full meaning. Particular individuals may only approximate various aspects of these concepts. In such a situations, it is difficult to process aggregations of approximations, resulting from various aspects of these complex concepts. In order to facilitate the qualitative judgment of aggregations, used in a project, we try to approximate them by means of measurable attributes. In CSE projects, the establishment of these attributes is a great challenge.

For example, the risk of failure in the case of a given project may be measured in numerous ways. As long as project participants are not aware of the key risk factors, they do not take them into account in the measures they use. On the other hand, one should take into consideration the fact that often, only once a given CSE project approaches the end, concepts that express a satisfactory approximation of acceptance criteria for final products are available. Hence, we should not expect that the attributes used for approximation of these concepts will be included in the project failure risk measures.

The needs, their hierarchies, relations between them, and available resources all constitute examples of "input." The "input" is subject to changes with time (including changes in meaning of needs and/or other changes in the meaning of concepts related to agent's knowledge). This constitutes an additional impediment for "computer programs," that deal with metaphorical "computer computations" described by us. In other words, the "input" of our metaphorical "computer program" changes while "computations" are being performed and therefore, a good "computer program" should have a special "software," that would control its adaptation with the changing conditions (including the "input" modification).

cse-4. *Metaphor of computations*. "Computations" are represented by interactions (especially interactions of c-granules) that support a computer (cf. Point cse-1) in the processing of complex, vague concepts, along with their approximation, decomposition, and selection, and in the processing of proper material resources. We assume that once computations are initiated and performed, a "computer" has an access to all necessary "input" components and is able to process them (cf. Point cse-3).

These computations are based on a broadly-understood process of granulation, that is the transformation of some c-granules into other c-granules. These transformations resemble interactions that are also controlled by c-granules.

cse-5. *Metaphor of an output*. "Output" is represented by the ultimate families of concepts, experiences, and material products, which, as intended by the project's control system, should comply with the project requirements.

cse-6. *Metaphor of software entities*. "Software entities" are represented by components of the soft_suit (e.g., m-hunks). Among them modeling techniques, reasoning techniques, interactions plans, and/or the hierarchies of aggregations based on input concepts, built over the dynamically changing models of interactive granular computations and performed in an open environment, are especially important. Computational models performed in an open environment also constitute aggregations. Software entities should take into account both potential changes in aggregated models and the results of interactions with the environment, which are either easy or difficult to anticipate. Thus, the metaphorical "software" is represented by some rules and/or principles often governing unexpected interactions instead of an exact list of actions for a given computer program. For example, a driver does not need such a precise list of actions. She/he only needs some rules important for safe driving. Metaphorical "software", understood in this way, has the aforesaid "input" (cf. Point cse-3), and by means of an "operating system" (cf. Point cse-2), controls computations (cf. Point cse-4) performed on a "computer" (cf. Point cse-1), whose operations are often disrupted by unexpected interactions with the environment. It should be emphasized that there is a significant difference between metaphorical "software" for a project and a traditional project implementation planning.

Traditional planning results in the creation of traditional plans, e.g., in the form of Gantt Chart or PERT Chart. In such a planning, all tasks and relations between these tasks are known. However, in the case of innovative CSE projects, these data are generalized to a high level. For example, usually, they do not take into account the specificity that results from the adaptive processing of concept approximations. Therefore, "software entities" for an innovative CSE project are more related to controlling the whole course of a project, perceived through a process of collaborative reasoning, that processes input knowledge and wisdom (that constitute "input," as defined in Point cse-3) into innovative, more complex products.

Programming for a project has variant solutions, and may include typical constructions, characteristic of programming languages (especially concurrent processes). In a sense, an ordinary computer program, which operates in a "closed, stable, and controlled world" of a computer, is a unique example of a "software entity."

When it comes to the stability and controllability of modern computers, the computational conditions in the case of ordinary computer programs are very comfortable, in comparison to the conditions of computations performed by the aforesaid "computer" (cf. Point cse-1). In the case of "software entities" for projects, the programming of co-operating PDCA cycles is very important. As part of these cycles, conditions determining the correction and/or the exit from a cycle are being improved. This constitutes a significant difference in comparison with classical programming languages.

As with ordinary computer programming, when programming the implementation of CSE projects, it is useful to apply model schemes at a right moment. For example, these could be the efficiency management schemes (cf. Fig. 33.5) together with the Crocos framework (cf. Fig. 35.1).

Nevertheless, as shown from the experience, each innovative CSE project has its own specificity, which should be taken into account in its implementation program. This unique specificity stems from the fact that one aspects of innovative means introducing something new or different. Therefore, the production of an innovative technology usually entails the necessity to solve very complex and difficult technical and conceptual problems, that has never been addressed by anyone before.

What is more, if we agree that the motto of the modern management is the following statement by Heraclitus:

> The Only Thing That Is Constant Is Change,

then the implementation of each complex project occurs in a "new environment" with "new conditions" and is performed by "new teams" of people.

Therefore, there is a small chance that a CSE project implementation program, that was once developed, could be applied once again, to another project exactly in the same form.

Practice shows that the implementation of every new innovative project requires a specific action program aim of which is the adaptation of knowledge, experience, work culture, and habits, exhibited by the new project participants. This remark applies also to numerous other aspects of innovative CSE projects. Therefore, each implementation program for innovative CSE projects is, in its detailed design, intended for a single application.

The above-mentioned intuitions behind the metaphorical treatment of a CSE project implementation as computations illustrate the degree of complexity related to the process of modeling such computations. It should be emphasized, however, that the aforesaid metaphor do not illustrates all aspects of difficulties with handling problems that constitute the FP3C (cf. Sect. 1.2.2). A slightly different type of such difficulties consist of constraints, that stem from the ontology of modern mathematics, based on a theory of sets, expressed in classical logic (e.g., ZFC theory).[1]

Set theory clearly describes the properties of certain variants of the static universe, often represented in the form of the von Neumann's static universe.[2] In this universe, sets are defined by means of *crisp concepts*.

In order to prove the validity of a given sentence, it is enough to prove that the supposition of its falsehood implies a contradiction. For example, if we prove that the negation of some property of sets is not true for all worlds perceived (permitted) by us (e.g., represented by models for ZFC theory), we may conclude that this property is true.

In a sense, this is a version of the "closed-world assumption," well-known in the field of IT, as extended to infinite sets.[3] We may apply this assumption to systems that have an access to full information. For example, some transaction databases (e.g., airline ticket reservation register) fulfill this assumption to a great extent.

In the 70 s of the 20th century, it was noted that numerous practical IT applications do not fulfill the "closed-world assumption." In these applications, it was necessary to process complex vague concepts, about which we do not have full information and, what is worse, whose meaning changes with time. Rule-based expert systems, for a particular domain of knowledge, that is still developing, constitute classical examples of such IT applications (e.g., MYCIN[4]). Therefore, for numerous decades, many attempts have been undertaken to manage the processing of vague concepts, about which we do not have full information, in a better way. The attempts at combating the constraints related to the "closed-world assumption" took a lot of work in a number of research centers.

As part of these attempts, numerous research trends have emerged, such as statistical reasoning methods (e.g., Bayesian networks, SVM), fuzzy logic, rough sets, and non-monotonic logic. These trends often refer to the so-called vague complex concept processing, on the basis of the "open-world assumption".[5] The problem of

[1] http://en.wikipedia.org/wiki/Zermelo-Fraenkel_set_theory.

[2] http://en.wikipedia.org/wiki/Von_Neumann_universe.

[3] http://en.wikipedia.org/wiki/Closed-world_assumption.

[4] http://en.wikipedia.org/wiki/Mycin.

[5] http://en.wikipedia.org/wiki/Open-world_assumption.

how to better cope with FP3C corresponds to these research trends very well. In CSE, we deal precisely with the processing of vague complex concepts in an open and partially known world.

Basically, there is no consensus between scientists and practitioners when it comes to searching for effective tools, that could support problem-solving in this area. Numerous scientists agree that this is still a great challenge of the 21st century.

On the basis of our practical experiences, presented in Part IV, and gathered from scientific research, conducted in collaboration with Professor Skowron, a WisTech approach towards the aforesaid problem was formed and discussed in this book. The approach is based on models of IGrC, that are in line with the FPW postulates (cf. Chap. 32).

The starting point for our approach was an observation that concepts used by teams of people, implementing CSE projects, are generated, developed, and modified by agents via interactions with the environment. These interactions are initiated by means of c-granules, so that the needs of an agent are expressed and satisfied in the best manner possible. The content behind of these concepts is encoded in hunks, which are available to agents. The possibilities of interactions with the environment and other concepts are closely related to each such concept. These possibilities are implemented by means of links, which are grounded in hunks, capable of transmitting the results of a given type of interactions.

This intuition constitutes a basis for introducing a concept, which is central to our approach, the concept of a c-granule.

C-granules constitute the generalization of ordinary concepts, but they may change over time (especially, their meaning may be modified), and as they are in continuous interactions (by means of links and hard suit) with other c-granules and the environment, they may gain experiences from the surrounding world and adapt to it. The concept of a c-granule is explained in more details in Chaps. 28 and 32.

The concept of a c-granule is a starting point for constructing models of interactive granular computations performed by the societies of agents in their environment of activity, which constitutes a part of the "open world." In this way, computational models significantly differ from models of computations performed on a Turing machine in a "closed world." The computational model proposed in this book is based on the postulates, presented in Chap. 32. These postulates may be interpreted in numerous ways, may differ in terms of the level of detail, and may include the specificity of a given application domain, as in the case of efficiency management (cf. Fig. 33.5) or the Crocos framework (cf. Fig. 35.1).

In the following part of the book, we will present a **condensed summary to show how the models of interactive granular computations**, proposed by us, function from the perspective of WisTech.

A simplified version of a WisTech computational model is governed by the following principles:

igc-1. The model is implemented in an "open" physical environment, in which agents are present.

igc-2. Each agent is equipped with, at least, control, memory, clock, sensors, and actuators.

igc-3. In the agent's memory, there are representations of c-granules, which can be either active or passive. The agent's control and/or active c-granules may activate or deactivate other c-granules.

igc-4. By means of interactions, an agent may partially recognize and identify phenomena and physical objects, which are conventionally referred to as (*four dimensional spatiotemporal) portions of matter or hunks*, in short [201]. In this way, an agent perceives and registers the results of interactions. Given the incompleteness of information available to an agent, only some *features of these results* may be *perceived* and *registered* by an agent.

igc-5. An agent may identify configurations of hunks, and on this basis constitute other c-granules, and/or interpret the content of a c-granule by means of other c-granules.

igc-6. *Interactive granular computations* of an agent are grounded on physical interactions of hunks. Agent uses a given c-granule to control these interactions.

igc-7. C-granules consist of three layers of hunk configurations:

  i. **Soft_suit** – configurations of hunks used by c-granules, that represent the properties recognized by the agent's control, related to the agent's memory and its states (m-hunks).
  ii. **Link_suit** – configurations of hunks, also known as *links*, that convey the results of interactions which , e.g., occur in m-hunks—known as *link beginnings*—and among other hunks (including m-hunks)—known as link endings. Through the link_suit, the results of interactions that occur in m-hunks, located in the soft_suit, can be transmitted. A c-granule uses links to, e.g., determine current relations between the beginnings and endings of a link. What is more, links support the functioning of sensors and actuators, available through c-granules.
  iii. **Hard_suit** – all hunks are indicated by links from a c-granule, whose mutual interactions and interactions with the environment support the functioning of the soft_suit and the link_suit. It may occur that some unforeseeable interactions of hunks with the hard_suit can disturb the interactions (expected by the agent's control), initiated by a c-granule.

igc-8. An agent, using her/his c-granules, may combine configurations of hunks that are available to her/him in order to initiate interactions, which lead to the results that the very agent expects.

igc-9. An agent may record scenes of actions, configurations of hunks initiating interactions, and some results of interactions that she/he perceives using c-granules.

igc-10. By means of c-granules, an agent may aggregate, reconfigure, and select families of hunk configurations (which may correspond with agent's c-granules) in order to initiate complex interactions and achieve complex results (or in order to generate new c-granules).

Agent's c-granules may carry out interactions that support the implementation of granulation processes by which other c-granules are reconfigured, aggregated, decomposed, and modified.
For this purpose, an agent may use the right to self-organization, held by the sets of hunk configurations that are in mutual interactions with each another.

igc-11. By means of c-granules, agent perceives some features related to the states of the agent's control. These include *agent's needs* and *relations between these needs*, along with the *agent's hierarchy of needs*. The states of the agent's control, perceived by an agent, are represented by c-granules.
An agent systematically aggregates the current versions of her/his hierarchies of needs and represent these aggregations in a form of a c-granule, which is referred to as *the current priority needs of an agent*, stemming from her/his current hierarchy of needs.

igc-12. The agent's control determines the priorities and the scope of interactions, initiated by an agent, on the basis of her/his current priority needs.

igc-13. Agent's perception of a perceived situation is a process of interaction in which all of sensors, actuators, agent's control, and agent's hierarchies of needs participate. The process leads to an understanding of a perceived situation, including the execution of the following tasks:

i. Assigning a context to a perceived situation.
ii. Determining the boundaries of a situation, as well as its relations with other situations form the past, situations anticipated in the future, and situations that are currently perceived by an agent.
iii. Choosing and/or constructing features of a situation, which can be important for the agent's hierarchy of needs in an identified situational context.
iv. Judging the significance of selected and/or constructed features of a situation from the perspective of the agent's hierarchy of needs and an identified situational context.
v. Pre-updating the agent's hierarchy of needs and her/his priority of needs.
vi. Selecting and/or constructing agent's plans of interactions and coordinating them with other interaction plans that are currently implemented. As a result, we should obtain c-granules that specify such features of an interaction plan as given below.

a. Expected final results of an interaction plan and conditions related to its completion.
b. Conditions which, once satisfied, enable and/or obligate an agent to start the implementation of an interaction plan.
c. Specification of basic actions for the initiation and or correction of an agent's interaction plan, whose completion should result in the initiation and correction of an interaction plan.
The specification should assume that the actions are implemented in the PDCA cycle, meaning that they encompass such conditions and techniques related to their initiation, implementation, and completion as:

AC1. Correcting and providing details of an interaction plan.
AC2. Performing an action.
AC3. Verifying and assessing the correctness of the action's course and its results.
AC4. Correcting an action during its implementation and/or upon its completion.

d. Specifications—concerning the relations—of a network of phases, related to the implementation of an interaction plan. These include such specifications as:
   (s1) specifications of relations between phases and specifications of the actions of transition between various sets of phases,
   (s2) sequences of consecutive conditions, expected by an agent to be satisfied in the course of a given phase; to these sequences should be assigned networks of consecutive actions, planned for the implementation in the PDCA cycle (along with conditions and techniques of their initiation, implementation, and completion, including the above-mentioned conditions—AC1, AC2, AC3, and AC4),
   (s3) conditions under which it is recommended to initiate ready-made interaction plans, that are potentially available to the agent's memory,
   (s4) sequences of conditions which, once satisfied, should result in the modification of an interaction plan within a current phase of its implementation and/or within other phases of its implementation.
e. Conditions determining the need for an emergency termination of an interaction plan implementation.
f. Actions initiating the emergency termination of an interaction plan implementation. In particular, this concerns

vii. Initiating, implementing, verifying, and correcting a selected interaction plan within the PDCA cycle.
viii. Updating the agent's hierarchy of needs and priority of needs on an ongoing basis, which may lead to the termination and/or modification of an interaction plan.

igc-14. An agent may perform the SWOT analysis at the level of accuracy, determined by the adaptively developed:

a. agent's hierarchies of needs, and
b. base of c-granules available to an agent (including attributes, concepts, classifiers, optimizers, and interaction techniques).

igc-15. The agent's control and its c-granules may be activated or deactivated and they can use other c-granules in numerous ways.
For example, let us assume that G is an agent's c-granule which may be used, e.g., in the following way:

a. By means of its links, G monitors and records the properties of observable interactions. On this basis, G may compute the values of attributes which help to recognize, identify, and classify the properties of objects and phenomena in relation to a perceived situation.
b. G may conduct decision-making processes related to recognizing, identifying, and classifying the properties of objects and phenomena by means of judging the aggregations of results, obtained through the verification of the degrees to which the criteria of such arguments as "FOR," "AGAINST," and "I DON'T KNOW" were satisfied.
c. G may be equipped with mechanisms by which it systematically confronts the actual results and conditions of interactions with the results expected by this c-granule and/or a given agent. It may also have mechanisms by which it systematically makes attempts at correcting the actual results and conditions of interactions.

igc-16. An agent tries to evaluate a network of available concepts (represented by c-granules) in and adaptive way, with particular emphasis on those concepts, which support the following agent's activities:

a. Perceiving, understanding, and improving the agent's adaptive hierarchy of needs.
b. Adaptively judging current concern of the situation of an agent in a given environment, including the SWOT analysis, which is interactively synchronized, in the best manner possible, with the agent's adaptive hierarchy of needs and the state of the environment.
c. Activities aimed at satisfying the hierarchy of needs by means of generating, selecting, initiating, verifying, and modifying appropriate interaction plans. An agent tries to ensure that her/his interaction plans have in-built mechanisms for correcting interactions, whose implementation differs from the agent's expectations. In the case of complex plans, an agent tries to ensure that the plans have an interrelated hierarchical nature in the form of PDCA cycles.
d. Activities aimed at supporting the efficiency management of interaction plans, generated and implemented by an agent. An agent tries to maximize the benefits, achieved at an acceptable (under given circumstances) and possibly low level of costs and risk.
e. Identification of packages of erroneous decisions made by an agent and any failures in actions that have been undertaken (especially those that resulted in major negative consequences) in order to initiate processes, aimed at clarifying the causes and minimizing the risk of repeating these errors and failures in the future.

igc-17. An agent may be formed as a result of interactions between a team of other agents. These agents cooperate with each other to determine the structure, functionality, meaning, and rules for the co-use of mutually shared c-granules. These mutually shared c-granules are called the *society c-granules*.

igc-18. The agent's control over interactive granular computations for complex applications may have a decentralized character. It may include controlling a self-organizing society of agents [130, 210, 311], aimed at establishing (e.g., by some specialized agents) networks of communication that convey information to various groups of agents about their behaviors. These networks (i.e., channels of communication and constraints which determine some limitations imposed on agents) may adapt themselves to the changing conditions of the environment.

It should be emphasized that the above illustration of how interactive granular models function (Points igc-1 – igc-18) is rather simplified and does not encompass numerous important aspects, that have been discussed in this book. The basic catalogue of the most important IGrC aspects, from the perspective of WisTech, may be found in the chapter on FWP (cf. Chap. 32). At the same time, the above illustration (Points igc-1 – igc-18) shows the great scope and complexity of interactive granular computation modeling. A more detailed explanation of the issues related to the scope and complexity of the IGrC modeling ontology can be found in Chap. 31. We have also explained some causes of the difficulties with grasping the entire complexity of the IGrC ontological model in the same chapter. Despite of these difficulties, when taking into account the role these computational models play in reducing the negative consequences of TPGP and constructing computational models for AI (cf. Sect. 35.2), it is certainly worth to continue expanding our knowledge and skills related to the construction of advanced IGrC models.

The metaphor presented above (Points from cse-1 to cse-6) indicates a possible way of using WisTech computational models (e.g., in a simplified form, presented in Points from igc-1 to igc-18) for describing, investigating, and applying IGrC models during the implementation of innovative CSE projects.

It should be emphasized that the majority of contemporary computational models, used in modern IT technologies (including software engineering) and modern computer architectures, are based on a computational model of a Turing machine[6] [508]. This computational model is referred to as the Turing computational model. It is based on a thesis, known as the Church Thesis (or Church-Turing thesis), which states that all intuitively effective computations may be expressed, using the Turing computational model.[7] In the literature one can find a discussion on the relationship of Turing model [508] of computations to criteria of "intelligent" behaviour (e.g., Turing test [510]). One can also find some attempts to modify the Turing test [436] (e.g., in the form of physically embodied Turing test [375]).

It should be noted that there are some basic differences between Turing computations and granular computations based on WisTech. The summary of these differences are presented in Tables 36.1–36.9.

[6] http://en.wikipedia.org/wiki/Turing_machine.

[7] http://en.wikipedia.org/wiki/Church-Turing_thesis.

Table 36.1 "**Hardware**": Church Thesis model versus WisTech IGrC model

Classical mathematical model of a Turing machine	A society of the physical WisTech agents, interacting with the environment that is open to external influences

Table 36.2 "**Control**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
Turing machine's control is defined in set theory, using crisp sets that include: states, set of states, and transition relation. In the Turing model, the control is not integrated with the process of monitoring the changes of events and/or conditions of computations that are being performed due to the fact of other processes independent from the Turing machine processes, occur in the computing environment	Physical processes in the environment, consistent with the laws of nature, cause changes to complex (most often defined by complex vague concepts) control states. The transition relation is also a complex vague relation evolving with time according to the laws of nature. The states and transition relation are approximated using relevant vague concepts. These approximations are also evolving with time. The control, using c-granules, may initiate interactions in the environment and partially perceive their results in the form of the recorded properties of configurations of hunks. The control is integrated with the process of monitoring the changes in events and/or conditions, which occur in the environment of computing. These changes create the base for the judgment of the agent's situation (in particular, for SWOT analysis), as well as for the adaptation of the hierarchy of needs and priorities of the tasks that are being performed. Proper comprehension of the perceived situations (i.e., perception of situations), including the identification of particular aspects of a given situation, important for an agent, is the basis for the selection of appropriate actions to execute later on. The effects of these actions may (or may not) satisfy the properties expected by an agent, which reflect the consequences of these actions. The perception and verification of the expected consequences of actions are implemented by relevant c-granules

Table 36.3 "**Memory**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
A set representing the tape of a Turing machine	Agent's c-granules, perceived by the agent's control, which are recorded in the states of the agent's memory

Table 36.4 "**Operating system**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
Managing the compliance with the Turing machine instructions	Managing (by the agent's control) the compliance with the instructions using meta-social c-granules and societies of agents behaving in a partially understandable way in order to initiate relevant interactions

Table 36.5 "**Input component**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
A sequence of number (more precisely, symbols) in the closed world of a set theory model, which can be interpreted by the Turing machine's control after representing it on the tape of a Turing machine. An input is accepted by the Turing machine if it can be recognized by the Turing machine's control. An input is precisely recognized by the Turing machine,	Any c-granule is grounded in physical objects, which act according to the laws of nature. C-granules interact with the open physical world. Each c-granule consists of links conjugating (assuring the transmission of interaction results) two worlds: (i) the external open world of interactions and (ii) the internal world of the properties exhibited by the states of the agent's control and states of the agent's c-granules (created by configurations of hunks). Interaction results, transmitted by links, consist of only partial information about the interactions. Any input constitutes imperfect (in particular, uncertain) information about the physical input and, in general, is not precisely recognizable within the scope available to the agent.

Table 36.6 "**Computations**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
The processing of crisp and stationary (in time) concepts. Computations are represented by the sequences of the Turing machine's configurations according to the stationary transition relation of a Turing machine. The computation may be repeated many times and not depend from changes taking place in the environment.	The processing of vague and dynamic (non-stationary) concepts is represented by c-granules. In general, complex computations may be repeated many times under the condition, that a relevant isolation from changes taking place in the computing environment (which may cause interactions in the environment), is provided. Typical computations, to a large degree, depend on changes taking place in the computing environment and on their interpretation by the agent's control in the context of the agent hierarchy of needs. Computations are subject to the laws of nature and they cannot be inconsistent with these laws.

Table 36.7 "**Output**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
Results and information represented in the form of numbers and/or images are obtained by decoding the contents of relevant tape fragments in the final computational state.	Acquired knowledge and wisdom (including adaptive judgment), represented by c-granules, interact both with one another and with the environment.

Table 36.8 "**Software**": Church Thesis model versus WisTech IGrC model

Turing model	WisTech IGrC model
Instructions understood by a Turing machine are directly and immediately interpreted by the machine's control.	Knowledge and wisdom expressed in the form adaptive schemes representing the (adaptive) behavior of an agent are used to develop and initiate interaction plans in the context of hierarchical, mutually linked interaction plans carried out in the PDCA cycles. These interaction plans can generate some adaptive rules of the agent's behavior. These rules can be assigned to specific types of situations in which an agent may be found in an open world.

Table 36.9 "**Main difference**": Computer and Church Thesis model versus WisTech IGrC model

Computer and Church Thesis model	WisTech IGrC model
Computations are performed in a stationary (in time) and 'closed' set theory model, simulated to a satisfactory degree by hardware which performs computations.	Computations are performed on c-granules. Their links constitute a mechanism, conjugating the results of interactions in the environment and in the agent's control.

The summary of the differences between the Church thesis model and the WisTech IGrC model is very condensed. From the point of view of experienced managers, who deals with innovative CSE projects, it is particularly important to take into account all properties of WisTech models that are related to their physical nature. One should also allow the fact that these models act in line with the mechanisms prevailing in nature, that are only partially known to agents by means of c-granules (which connect the "agent's mind" with the surrounding environment).

The key to success in managing a project is a skillful approximation of concepts, represented by c-granules, and a skillful use of c-granules by those, who are in charge of a given project. It applies both to input society c-granules and input private c-granules (for example, those that constitute an "input" as understood by the metaphor cse-3, presented above), as well as c-granules related to substantive and organizational aspects of a project and c-granules related to the project's control. **The WisTech language and models should allow for a more in-depth and uniform approach towards understanding and solving problems in CSE projects. At the same time, the language should provide a transfer of knowledge and experiences between projects with very remote fields of application**.

36.2 Future Research

In Sect. 31.1, we have explained that a full comprehension of theoretical and practical aspects related to the ontology of WisTech models would mean a major progress in dealing with scientific and technical problems, that are qualified as the greatest and still open challenges of the 21st century (cf. 1.1). Therefore, it is not an easy task to expand our knowledge on WisTech models that support practical problem-solving but it is certainly necessary to put a lot of effort in it.

It seems that an important milestone in this direction may be a more precise comprehension and development of practical applications, especially when it comes to the following issues:

1. Determining and clarifying the mechanisms of granulation, understood as a constructive, quick, and efficient formation of "optimal" c-granules for solving a given class of problems.
2. Determining and clarifying more precise relations between granulation and efficiency management.
3. Expanding our knowledge and skills in terms of specifying an optimally efficient method of controlling computations in a WisTech model, in order to use it for effective solving of a given class of problems (based on the metaphor of declarative programming, e.g., analogically to logic programming).

It should be emphasized that the proposals presented above constitute only some examples of further research directions. The issues related to WisTech models seem potentially important for all areas of modern engineering. Nevertheless, the progress will be possible if we stick to the practice and broadly-understood material (physical) on fundamentals of WisTech models. However, the verification and development of such an approach requires large teams of people and, of course, heavy expenditures.

Appendix A
Principle of Anchoring a Project in Top Management—Examples of Detailed Principles

By and large, the *Principle of Anchoring a Project in Top Management* (cf. Sect. 14.3.2.1) in particular, means **ensuring a proper implementation of the following tasks by top management**:

a1. Continuous overview and revision of the *current organizational strategy*, **with regard to all important projects** (especially mission critical projects) implemented by a given organization, including overview and revision of its mission, politics, objectives, SWOT,[1] risks, portfolio of implemented and planned projects (including the anchored project), mutual relations between projects from the portfolio (including their priorities), and overview and revision of hierarchies of needs of a given organization.

a2. Positioning of an anchored project in the current organizational strategy, with special regard to:

   a2a. optimal understanding of project values for the current organizational strategy and the understanding of the *current constraints, context, and specificity of implementation*,

   a2b. potential synergies with other projects (already implemented, currently implemented, and planned) and as part of the cooperation with strategic partners and customers.

a3. Continuous overview and revision of missions, objectives, motivations, risks, and the state of the implementation of **potential projects to be anchored** with regard to the current constraints, context, and specificity of a project.

a4. Continuous provision of the best possible staff and resources necessary for the implementation of an anchored potential projects. It particularly involves launching a system for managing knowledge related to a project (including continuous, internal project trainings) and synchronizing it with systems for managing knowledge in an organization.

a5. Transfer of appropriate project tools to a project manager and to his/her team:

[1] https://en.wikipedia.org/wiki/SWOT_analysis.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

- a5a. granting a project manager the authorization to make all decisions that are necessary for a proper implementation of a project, in line with expected project requirements. These decisions may, for example, include:
  - i. decisions concerning organizational and technical changes that will shape the final form of end and intermediate products in a project,
  - ii. binding resolving of conflicts among project participants, experts, and end users (especially, end users who force proposals of contradictory project requirements),
- a5b. establishing procedures for managing changes in expected project requirements,
- a5c. providing resources necessary for an effective and efficient introduction of all consequences of essential changes in the implementation of a project.

a6. Continuous communication of top managerial decisions . Especially, communication to project participants and employees (of a given organization) about a successful execution of a project and its current state of implementation, while listening to constructive criticism and establishing a relevant dialog with groups of opponents (especially agents that have a negative influence on a project).

a7. Provision of effective mechanisms for a successful accomplishment of tasks assigned to main project stakeholders throughout the entire implementation of a project.

a8. Continuous improvement of communication and supporting mechanisms (including the "help-desk", learning, and knowledge management mechanisms) among project participants and users of products distributed for tests and pilot solutions exploitation.

a9. Launching and improvement of procedures for implementing new project participants and new end users.

a10. Implementation and improvement of the IV&V principle.

Appendix B
Principle of Providing a Proper Role of the Human Factor in Management—Examples of Detailed Principles

The Principle of Providing a Proper Role of the Human Factor in Management (cf. Sect. 14.3.2.3) means, by and large, a continuous implementation of many tasks related to the human factor, e.g.:

hf01. **Reinforcing positive motivations and emotions** of all project stakeholders (particularly key members of a project team) by appropriate project management.

hf02. Providing a good and positive atmosphere within project teams to support effective and efficient teamwork.

hf03. Preparing a specification of requirements related to the desired properties of the human factor in processes related to the project's implementation. Focusing on requirements needed for the implementation of a project and related to psychological profiles and professional human resources, including the scope of project-related knowledge and technical skills of domain and technical experts. Quantification of character of project participants based on mostly needed project requirements relevant to human factor (e.g., https://hbr.org/2015/04/measuring-the-return-on-character). Ensuring a continuous adaptation (to the current needs) of specification of requirements related to the human factor.

hf04. Launching and maintaining a continuous adaptation of plans aimed at recruiting the optimal human resources according to requirements related to the human factor.

hf05. Evaluating and developing human resources (including ongoing trainings of the staff) in a project. The evaluation criteria should include both project-related criteria and criteria related to personal traits, which influence the ability of creative (in a positive sense) and efficient cooperation in a team.

hf06. Ensuring a proper scope, form, and motivation of domain experts' involvement.

hf07. Managing the positive motivation, ergonomics, attractiveness, and comfort of the work of a project team and end users.

hf08. Managing changes related to various aspects of the human factor, including customs and habits of a project team and future end users.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

hf09. Managing the human factor in processes supporting quality and security management, especially in terms of mechanisms for independent (from a project manager) verification and validation of progresses in the project's implementation.

hf10. Managing the human factor in processes supporting communication within project teams and with stakeholders. Paying particular attention to the role of the human factor in identifying and communicating requirements related to end products, articulated by the future end users, domain experts, and key stakeholders.

hf11. Managing the human factor in processes related to project safety management and organization of people involved in the project's implementation.

Appendix C
Principle of Providing a Proper Level of Communication—Examples of Detailed Principles

Communication is maintained between people and information processing systems (cf. Sect. 14.3.2.4). In general, people are assigned specific project roles. The most common project roles include:

r01. end user of products and/or project results,
r02. sponsor,
r03. top management (of an organization responsible for a proper implementation of a project),
r04. project manager,
r05. architect,
r06. system engineer,
r07. domain expert,
r08. analytic,
r09. designer,
r10. programmer,
r11. tester,
r12. administration support.

Basis issues that affect the quality of communication are especially connected with the answers to such questions as:

c01. What are the criteria for establishing current priorities (with regard to particular groups of recipients) of a message conveyed by a sender?
c02. What are the criteria for establishing current priorities (with regard to particular groups of senders) of a message received by a recipient?
c03. In what language and form should messages be conveyed?
c04. Is there a need for "translators" who will turn the message from a sender's language into a recipient's language? One should remember that within a given language, e.g., English, there are numerous dialects of various groups of experts (e.g., finance experts use a different dialect from experts of molecular structures, who compute homology groups of simplicial complexes in space).

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

c05. At what time and how fast should messages be conveyed?
c06. What are the expected results and further activities stemming from these messages?
c07. Who and in what form should authorize messages that are to be conveyed?
c08. Which communication channels are normal, which are reserved, and which are extraordinary?
c09. What messages are required in order to defuse tense situations between people involved in a project and who should send them (for a project, contradictory messages sent to project participants by a project manager and stakeholders (including top management) are particularly dangerous)?
c10. What are the expected rate of information flow and capacity of particular communication channels?
c11. What are the procedures for settling conflicts arising from receiving contradictory messages?
c12. What are the key communication channels in a project and what are the communication standards and procedures associated with them?
c13. How to verify whether a message reached the recipient, whether it was properly understood by him/her, and whether he/she has taken the desired actions?

In the case of CSE projects, people very often underestimate the significance of the quality of communication in terms of supporting a proper aggregation of the current knowledge within the scope that covers the following issues:

K1. *Expected project requirements*, which should be satisfied by intermediate and end products (including the acceptance criteria), as well as requirements related to the way in which a given project is implemented. Expected project requirements may concern:

K1.1. *the list of end and intermediate products*, with the identification of project managers,
K1.2. acceptance criteria for end and intermediate products,
K1.3. *legal and contextual requirements* (including cultural requirements) that ensure a proper implementation of a project, including standards related to the techniques (methods and tools) used for the implementation of a project, as well as standards related to end and intermediate products,
K1.4. roles (including rights and responsibilities) in a project (e.g., roles identified in points r01-r12 and/or other roles characteristic of a given project and its needs) and *organizational structure*,
K1.5. principles of *having rights and resources* at one's disposal,
K1.6. *communication* channels (including such channels as: meetings and/or project teleconferences, information bulletins, intranet multimedia knowledge portals (with training materials), points with current knowledge about contact details and availability of project participants),
K1.7. register of changes to important *specific properties of a project and its context*, which have and/or may have a significant influence on project constraints (e.g., such a register may be used in a form of the results of a

complex SWOT[2] analysis and/or as an identification of the current, critical success factors[3]),

K1.8. estimated demand for *resources, measures, and tools* that are necessary for the project's implementation,

K1.9. *project milestones*[4] and currently expected *implementation dates of project milestones, as well as acceptance criteria for the fulfillment of particular milestones*.

One should remember that expected project requirements are adaptive in nature and thus, ought to be modified along with the current changes in the state of the project's implementation and project constraints.

K2. *Register of the current versions of artifacts (including products and documents)* that constitute the currently approved documents, including those that encompass expected project requirements, and the register of the current versions of products generated as part of a project.

K3. *Judgment on the state of the project's implementation, including current results obtained as part of the verification and validation of progresses in the implementation of expected project requirements*. This knowledge encompasses the identification and estimation of possible differences between expected project requirements and the actual state, as well as judgment on the possibilities and validity of satisfying expected project requirements.

K4. *Proposals how to streamline the implementation of a project and/or how to introduce changes in expected project requirements*. These proposals should include conclusions from the judgment on the state of the project's implementation (cf. K3).

K5. *Judgment on the level of motivation, activities, and intentions of project stakeholders.*

K6. *Judgment on the level of aggression, activities, and intentions of agents, that could have a negative impact on a project*.

K7. *Register of risks and project issues* that are to be tackled, along with the identification of people responsible for providing solutions and informing about the deadlines and actual states of the solving process.

K8. *Analysis and judgment of current costs/benefits and risks/co-risks (cf. FPW-13 in Chap.* 32*) to recommend changes* in the expected project requirements. This should also include analyzing factors that signalize a need for a possible recommendation of a decision about introducing changes to expected project requirements (including a decision about an early termination of a project).

K9. *Maps and search engines for artifacts (including documentation and project products)*. With the development of each CSE project, the scope and complexity of artifacts grow. It is particularly important to provide project

[2]https://en.wikipedia.org/wiki/SWOT_analysis.

[3]https://en.wikipedia.org/wiki/Critical_success_factor.

[4]https://en.wikipedia.org/wiki/Milestone_(project_management).

participants (according to their respective rights) with an easy and quick access to necessary fragments of the project documentation (both its current versions and the archived versions). For this purpose, supporting tools, such as an "intelligent" guide and assistant for searching project artifacts, should be made available to all project participants. Such a tool is referred to by us as an *integrated map and search engine of project artifacts*.

K10. *Register of currently introduced changes to expected project requirements and judgment on the validity of their implementation*. This includes verifying and validating the degree to which these changes have been implemented and the degree to which the expected results of these changes are satisfied.

The scope and specificity of knowledge possessed by a project participant should be adjusted to her/his project role. Consequently, scopes of knowledge encompassing points K1-K10 listed above are distributed among different project roles.

By the *Principle of Providing a Proper Level of Communication* (cf. Sect. 14.3.2.4), we mean establishing and implementing possibly the best (as far as the resources allocated for a given project allow) communication mechanisms in a given project. These mechanisms should comply with the following requirements:

(cl-1) communication mechanisms encompass all important roles (e.g., roles r01-r12) that appear in a project,

(cl-2) communication mechanisms are closely integrated with the Principle of Providing a Proper Role of the Human Factor in Project Management (including points hf01-hf11),

(cl-3) communication mechanisms include settlements of problems (including resolving of conflicts) which are part of the basic issues that affect the level of communication (including points c01-c13),

(cl-4) communication mechanisms include mechanisms for a proper synchronization of current knowledge (K1-K10), distributed among project participants and project stakeholders.

Appendix D
IV&V Team—Examples of Detailed Principles

During the implementation of a project, an **IV&V team (cf.** Sect. 14.3.2.5**) should continuously satisfy**—at the best possible level of quality—the following requirements:

i1. All tasks and assessments are performed by an IV&V team in a way that ensures as greatest **independence from a project manager** as possible. Technical, financial, and managerial independence from structures subordinate to a project manager is also important.

i2. The basic tasks of an IV&V team include:

- continuous **verification of processes related to designing, creating, and testing project products** (an IV&V team should particularly verify whether principles defined in project requirements are properly implemented),
- continuous **validation of the quality with which project products satisfy the requirements** of their future end users.

i3. An IV&V team manager has a **direct (independent from a project manager) and continuous access (e.g., for at least a quarter of a year) to top management of a given organization and project sponsors as well as a continuous access (e.g., at least once a week) to top management representatives** and representatives of project sponsors.

i4. An IV&V team **has essential resources and competencies for an independent (from a project manager) verification, validation, and integration of the judgment on the state of the project's implementation**, with a particular emphasis on the current state of knowledge within the scope of points K1–K10 from Appendix C.

i5. An IV&V team manager closely **cooperates with a project manager** to provide her/his team with a necessary access to the current documentation and products and to minimize any possible disturbances related to the activity of this team in the implementation of a project.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

i6. An IV&V team records and documents (independently from a project manager) any possible **risks and project issues that could lead to risks, generated by top management and/or sponsors and/or other stakeholders**.

i7. An IV&V team continuously **documents and aggregates** (also in a form of management sum-up reports for top management) consecutive states of knowledge possessed by the team and related to progresses in the implementation of a project (with a particular emphasis on points K1-K10 from Appendix C). These reports should include a short analysis of the most important changes in a project and their dynamics, as well as the identification of new important trends in works related to a project.

Appendix E
Principle of an Appropriate Document Management—Examples of Detailed Principles

In the case of common CSE projects, from the very first stages of conceptual works, the basic documentation (cf. Sect. 14.3.2.7) should clearly define the current state of knowledge about a planned and/or implemented project. Generally, this state of knowledge encompasses scope of points a^*–d^* in Sect. 14.3.2.6. In particular, it covers the following aspects of a project:

a. **Project motivations, objectives, and project requirements, as well as expected and acceptable scopes of changes in project requirements** that are part of the project's implementation, including legal, organizational, and technical changes which stem from a successful execution of a project. Project requirements should include all factors that influence the implementation of a project, as well as requirements and acceptance criteria concerning end products and results.
b. **Project plans and principles (including standards, practices, procedures of change control, and CSF)**, techniques and implementation plans, as well as selection and evaluation criteria of a project manager.
c. Formal **appointment of a project manager, implementation organizational structures and communication procedures, as well as the allocation of scopes of responsibilities and resources necessary for the project's implementation**. These resources should encompass both human and financial resources, as well as essential competencies of decision-making (that satisfies project requirements) regarding the final shape of solutions, principles followed as part of a project, settlements of project conflicts, and introduction of changes (to the extent permitted by a given project).
d. Principle of independent verification and validation (IV&V).

In the case of complex projects whose total cost (of using internal and external resources) exceeds the approved threshold (e.g., several million euros), it is recommended to be more specific when it comes to the minimal content of the basic documentation from the very early stages of project works. For example, the basic documentation may be prepared according to the following scheme:

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

A. **Motivations for launching a project, objectives, and project requirements, as well as expected and acceptable scopes of changes in project requirements**. In this regard, the documentation should include the current state of knowledge, encompassing such aspects as:

1. Project justification and motivation.
2. Most important objectives, as well as key end products and results.
3. Expected project requirements, which usually encompass expected project constraints (e.g., time, resources, relations to other projects), requirements and acceptance criteria concerning end products and results, scopes of possible changes to project requirements and specifications of project decisions reserved to the top management. The recommended scope of expected project requirements also encompasses the requirements related to issues K1.1–K1.9 from Appendix C. "Expected project requirements" should be general at first. At the same time, they should be specific enough to perform a feasibility study. Next, these requirements should be specified and modified, along with progresses in the project's implementation and changes in project constraints. These modifications should also be subject to proper procedures of changes.
4. Expected and acceptable scopes of changes in project requirements that are part of the project's implementation, along with legal, organizational, and technical changes which stem from a successful execution of a project. Expected degree of changeability in the scopes of changes related to legal aspects, technical aspects, organization, and competencies that are to be made during the project's implementation.
5. Basic limitations, consequences, constraints (including relations to other projects) of the project's implementation.

B. **Project plans and principles (including standards, practices, procedures of change control, CSF), techniques and implementation plans**. In this regard, the documentations should include the current state of knowledge, encompassing such aspects as:

1. Family of principles (including standards and practices) obeyed in a given project (at the beginning, a family of principles should be relatively easy to implement) and a plan for the adaptive development of principles in a project.
2. Basic techniques (i.e., methods and tools), project implementation plans, and procedures of introducing changes, including acceptable mechanisms used to modify basic project documentation and implement the results of these changes.
3. Techniques for iteractive specification and revision of requirements and acceptance criteria related to the quality of end products (mock-ups, prototypes, pilot solutions, evolutionary character of implemented versions).
4. List of *partial deliverables* stemming from the project's implementation, along with the quality acceptance criteria.

5. Decomposition of a project plan and detailed plans. Expected results of key intermediate tasks and criteria for their acceptance.
6. Estimates of resources (including the minimum scope of knowledge and skills) necessary to perform key tasks.
7. Specification of a minimum scope of professional and decision-making competencies of a project manager responsible for ensuring a proper implementation of a project. The scope should be sufficient for implementing all organizational and technical changes that are expected during the project's implementation.
8. Detailed project's implementation plans.
9. Procedures of introducing changes in the project's implementation. Acceptable mechanisms used to modify basic project documentation and implement the results of these changes.
10. Selection and evaluation criteria of a project's manager.

C. Formal **appointment of a project manager, implementation of organizational structures and communication procedures, as well as the allocation of scopes of responsibilities and resources necessary for the project's implementation**. In this regard, the documentations should include the current state of knowledge, encompassing such aspects as:

1. Appointment of exactly one person for the position of a project manager that is responsible for a proper implementation of a project and current decision-making process, according to project objectives and requirements.
2. Estimating the scope of necessary resources and measures allocated for the implementation of a project and establishing rules for the use and settlement of these resources. They should encompass both financial and human resources, as well as competencies of decision-making (that satisfies project requirements) regarding the final shape of solutions, principles followed as part of a project, settlements of project conflicts, and introduction of changes (to the extent permitted by a given project).
3. Specification and appointment of structures related to management, organization, and communication.
4. Transfer of project tools and resources to a project manager and implementation of principles for the use and settlement of these resources.
5. Transfer of role-specific responsibilities and obligations to a project manager. In particular, he/she should be responsible for:

   – achieving project objectives in the most efficient way possible, at the same time fulfilling project requirements,
   – continuous reporting (to the top management and key sponsors) about progresses in the project's implementation (reporting the state of knowledge within the scope of points K1–K10 in Appendix C is a good practice),
   – current basic record keeping.

D. IV&V principle. In this regard, the documentation should include the current state of knowledge, encompassing issues K1–K10 defined in Appendix C.

Both with relatively easy and more complex CSE projects, one should remember that apart from *basic project documentation*, it is necessary to include *technical project documentation*. This type of documentation depends of the techniques and standards used for the implementation of a project. Consequently, technical project documentation should be specified as part of *partial project deliverables*. In general, technical project documentation includes:

td1. **Glossary of concepts** used in a project. Such a glossary may be in a form of ontology, represented by definitions of concepts (including rules of using them) and *class diagrams*,[5] visualizing mutual relations between concepts. A project-dedicated web portal in a form of Project Wikipedia constitutes a convenient tool for storing glossaries and project documentation.
td2. **Requirements of users, stakeholders, and external systems**.
td3. **Assessment criteria and acceptance scenarios** of solutions that are to be the project's end products.
td4. **Ideas of solutions** that are to fulfill the requirements of users and stakeholders.
td5. **Technical projects** of solutions that are to fulfill the requirements of users and stakeholders.
td6. Consecutive versions of **iterations of target solutions** that are to fulfill the requirements of users and stakeholders (including those concerning, for example, mock-ups, prototypes, pilot implementations).
td7. Current versions of **user-dedicated documentation and training materials**.
td8. Requirements concerning the **processes of implementation, trainings, user support** (according to the Service Level Agreement (SLA)[6]) and requirements concerning the stabilization of solutions and related processes (during pilot implementations, proper implementation, and initial exploitation of a system).
td9. **Register of technical issues** and/or problems to solve related to the processes of generating and implementing solutions. The register should include the statuses and people responsible for developing particular issues, along with the current deadlines.

[5]https://en.wikipedia.org/wiki/Class_diagram.

[6]https://en.wikipedia.org/wiki/Service-level_agreement.

Appendix F
Examples of Basic Principles of CSE Project Development (BPCD)

The BPCD principles (cf. Sect. 14.3.2.9) should be introduced in an evolutionary manner and consistently adapted to the current specificity and context of a project. A recommended family of BPCD principles may, at the initial stages of a project, include the following principles:

bp01. **Principle of Anchoring a Project in Management** (cf. Sect. 14.3.2.1).
bp02. **Continuous Identification, Judgment and Adaptation of a Family of Currently Used Principles** (cf. Sect. 14.3.2.2).
bp03. **Principle of Providing a Proper Role of the Human Factor in Project Management** (cf. Sect. 14.3.2.3).
bp04. **Principle of Providing a Proper Level of Communication** (cf. Sect. 14.3.2.4).
bp05. **IV&V Principle** (cf. Sect. 14.3.2.5).
bp06. **ABC Book Principle** (cf. Sect. 14.3.2.6).
bp07. **Principle of an Appropriate Document Management** (cf. Sect. 14.3.2.7).
bp08. **Basic Survival Principle** (cf. Sect. 14.3.2.8).
bp09. **Provision of the actual, not sham**, support in solving project problems (including the support of a project manager) by **key stakeholders and all indispensible domain experts**.
bp10. **Continuous improvement of human resources involved in a project, including trainings, increasing motivation, and developing the quality of cooperation between a project team and stakeholders**. This means improving the quality of teamwork and increasing positive motivation in a team, which includes such processes as:

- Verifying whether key project roles at particular stages of a project, along with specifications of requirements concerning people that are assigned these roles, were correctly identified.
- Developing and updating a map of requirements for competencies, skills, experiences and requirements concerning the safety and responsibility for particular project roles.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

– Maintaining a continuous synchronization of HR needs with recruitment processes aimed at recruiting the best possible staff and staff reserves.
– Determining and implementing training schedules for the members of a project team and shareholders. Verifying the quality of these trainings and, if justified, repeating them.
– Implementing motivational mechanisms for a project team, adjusted to project requirements.
– Ensuring properly adjusted (to project requirements) representation and involvement of domain experts, key users, and other project stakeholders (it particularly concerns project sponsors and top management representatives).

bp11. **Continuous evaluation of staff prepared for the accomplishment of project tasks and staff that accomplishes these tasks**. The staff should meet business needs as well as motivational, emotional, ethical, and psychological requirements. They should have communication skills and be able to work in a project team.

bp12. **Scalability and reusability of project components**. It is connected with a need for a great scope of scalability of project solutions. If it has a substantive and economic justification, the implementation of adaptive project objectives can be made on the basis of scalable and reusable (especially in terms of other projects) project components (e.g., development architecture components).

bp13. **Interoperability and interfaces**. The principle is to ensure the greatest possible interoperability of a constructed solution in its environment of activity. It includes identifying, integrating, and managing the evolutionary processes of all types of important interfaces. These interfaces include interfaces which operate between project teams, interfaces between a human being and devices, interfaces connected with software components and their architecture, data interfaces, and interfaces that connect a given system to the surroundings. Additionally, apart from the evolution of interfaces, the process of management should also apply to managing ontology/standards and managing a system devoted to the interface quality assurance. At the very beginning of any CSE project, a control process should be initiated. A particular attention should be paid to the ergonomics and legacy of users' interfaces, as well as their habits and customs (not necessarily related to their workplace).

bp14. **Separation of IT environments which deal with systems' mock-ups, prototypes, design, coding, tests, pilot implementations, proper implementation, stabilization, production, and marketing.**

bp15. **Management of communication quality indicators**. This is a specification of the Principle of Providing a Proper Level of Communication. It includes the introduction of quality and communication efficiency measures, as well as the identification of "bottlenecks" and implementation of mechanisms for improving communication.

bp16. **Configuration management**.[7] Configuration management involves technical, administrational, and managerial activities, aimed at accomplishing the following tasks:

- identifying and versioning all important project artifacts (including documentation, partial products, and end products), groups of these artifacts, and their current versions referred to as *project configuration*,
- managing configuration changes (including planning changes, analyzing costs and benefits of changes, accepting changes, launching and supervising procedures of introducing changes).

Maintaining single responsibility for managing changes (according to the binding procedures) within each artifact in the project configuration is a good practice.

bp17. **Stabilization**. The aim of this principle is to stabilize solutions which constitute the results and/or products of CSE projects. The principle requires special attention. It particularly concern complex projects, where we deal with an intensive software use. Stabilization requires great efforts at the initial phase of the system's exploitation. The principle is based on an observation, whereby any complex system that uses complex (especially new) software is very likely to have errors. Discovering these errors is only a matter of time. Therefore, it is very important to prepare special testing, verifying, and validating procedures, as well as procedures for controlling changes introduced during the initial stages of the project's implementation. These steps are crucial in the processes of risk management. The procedures should be subject to a proper adaptation. Some of the elements should be active for the whole period of the system's activity and some, only during the initial stages.

bp18. **Synchronization of the legal and financial infrastructure with expected project requirements, especially the Project's Milestones**. The principle encompasses mechanisms for designing and developing proper adaptive contracts, legal regulations, as well as organizational and contractual mechanisms for suppliers, which support the efficient cooperation in achieving project's objectives and minimize and/or shift the risk associated with a project. It is important to treat subcontractors as partners on a "win-win" basis.

Legal and financial infrastructure should be adaptively adjusted to the specificity of a project and the context of its implementation. Mechanisms involved in the infrastructure should make the quality of implementing the most significant project objectives conditional on payment. They should include mechanisms for the final acceptance of all deliveries, covered by the order.

Moreover, in the case of CSE projects, one must foresee the necessity of introducing changes to the project's implementation in legal and financial mechanisms. Sometimes, these changes result in great costs, which should not be borne by a contractor. In such a situation, a contract is no longer "win-win" and often, problems may arise for all parties involved.

[7] https://en.wikipedia.org/wiki/Configuration_management.

bp19. **Uniformity and adaptability of principles and techniques used for the implementation of a project and its management**. The aim of this principle is to increase the quality of communication and boost the efficiency of trainings for project participants through a gradual elimination of those areas of activity, within which different standards, methods, and tools are used. It may lead to a decrease in the quality of communication and an increase in the costs of trainings for project participants.
As a result, it is important to monitor and provide as great uniformity of principles and techniques used for the implementation of a project as possible. For example, using different notation and modeling standards by different project subgroups significantly increases errors and risk in the project's implementation. Adaptability of decisions concerning the selection of optimal principles, implementation techniques, and management techniques is equally important. A typical set of techniques includes appropriate: standards, methods, tools, organizational schemes, cooperation scenarios, testing criteria, change control procedures (which may be a part of configuration management process), knowledge management procedures, communication procedures, and structuring processes in integrated management (especially with regard to life-cycle structuring). Choices should be made based on how the changing specificity and context of a project as well as the results of benefit/cost/risk analysis influence the decisions about introducing a change.

bp20. **Electronic Project Documentation Management**. In the case of very complex projects, it is recommended to implement and adapt an electronic document management system in order to support the quality of processes conducted as part of the project's implementation. Moreover, it is recommended to synchronize this system with project configuration management (including procedures of introducing changes) and a solution that is properly tailored to the project's needs, e.g., in a form of a Project Wikipedia.

bp21. **Identification of the areas of activities, in which it is possible to use ready made templates of solutions**. These may include the templates of data models, ontologies, reasoning schemes, or simply organizational templates.

bp22. **Safe, reasonable, and under top management control of outsourcing**. Top management should ensure that the core business is not deteriorated (and weakened) as a result of unskillful outsourcing and/or failure to follow adequate outsourcing principles. Moreover, outsourcing should be done in such a way as to secure the possibility of changing a subcontractor in the shortest possible time without any business losses for the contracting party.

bp23. **Continuous improvement of the organization's maturity level in terms of an appropriate adaptation of the Capability Maturity Model Integration (CMMI)**.[8] In the 90s, a predecessor of CMMI was used, known as the Capability Maturity Model (CMM).[9]

---

[8] https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration.

[9] https://en.wikipedia.org/wiki/Capability_Maturity_Model.

bp24. **Continuous improvement of integration, efficiency, and results of mutual synergies of all management subsystems**, particularly with regard to such subsystems as:

1. quality management,[10, 11, 12]
2. information security management,[13]
3. service management,[14] integrated with other management subsystems (e.g., with the security management subsystem[15]).

bp25. **Development of all complex solutions by iterative prototyping using the PDCA cycle.** It is recommended to perform the continuous improvement of integration and efficiency in a PDCA cycle[16] that synchronizes PDCA cycles in order to improve particular management subsystems.

---

[10] https://en.wikipedia.org/wiki/Quality_assurance.

[11] https://en.wikipedia.org/wiki/Six_Sigma.

[12] https://en.wikipedia.org/wiki/Lean_Six_Sigma.

[13] https://en.wikipedia.org/wiki/ISO/IEC_27001:2005.

[14] https://en.wikipedia.org/wiki/ISO/IEC_20000.

[15] http://www.iso.org/iso/news.htm?refid=Ref1696.

[16] https://en.wikipedia.org/wiki/PDCA.

Appendix G
Criteria for the Optimization of a Family of Currently Used Principles—Examples

When using the Principle of a Continuous Identification, Judgment, and Adaptation of a Family of Currently Used Principles (cf. Sect. 14.3.2.2), it is recommended to optimize a family of currently used principles. It is good idea to use general practices from widely accepted standards (e.g., ISO/IEC 42010, TOGAF), Some of the practices are listed in Sect. 16.3.7. Having in mind this, it is worth considering the possibility of using the following optimization criteria:

C1. **Usability**. Usability is particularly important for arriving at solutions to all important project issues. It is also very important for increasing the efficiency and effectiveness of cooperation among a project team.

C2. **Understandability**. Principles should be easily and quickly grasped by people working within the entire organization (especially for people, who are obliged to follow appropriate principles).

C3. **Efficiency of an implementation strategy**. The principle should help (if economically justified) improve the efficiency of implementing strategic objectives (and not only project objectives) of an organization, which deals with the implementation of a project.

C4. **Robustness.** The principle should support the decision-making process, even in the most unpredictable situations and within a hardly predictable scope. It particularly concerns the decisions related to system architecture, as well as techniques and plans for the project's implementation. Each principle should be sufficiently specific and precise. Principles should support a consistent implementation of distributed decision-making processes, which occur under potentially unpredictable circumstances.

C5. **Stability**. The best principles should be stable. One should strive to ensure that they are in force, even if unpredictable changes occur. Each change results in additional costs and risk factors. Theoretically, an ideal principle should be properly adjusted so that it can be used in all possible situations that occur during the project's implementation.

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

References

1. Cyber-physical and ultra-large scale systems (2013). http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=85282
2. ACT-R cognitive architecture (2015). http://act-r.psy.cmu.edu/
3. Soar Publications (2015). http://soar.eecs.umich.edu/
4. Abramsky S, Coecke B (2007) Physics from computer science. Int J Unconv Comput 3:179–197
5. Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: The role of humans in interactive machine learning. AI Mag 35:105–120
6. Anderson JR (2007) How can the human mind occur in the physical universe?. Oxford University Press, New York, NY
7. Anthony M, Biggs N (1992) Computational learning theory. An introduction. Cambridge tracts in theoretical computer science. Cambridge University Press, Cambridge UK
8. Arbib MA (ed) (2002) The handbook of brain theory and neural networks. MIT Press, Cambridge, MA
9. Ariew R, Garber D (eds) (1989) Leibniz: philosophical essays. Hackett Publishing Company, Indianapolis
10. Aristotle (2014) On Interpretation (Trans.). The University of Adelaide, Adelaide. www.ebooks.adelaide.edu.au/a/aristotle/interpretation/
11. Aristotle (2014) Topics (Trans.). The University of Adelaide, Adelaide. https://ebooks.adelaide.edu.au/a/aristotle/a8t/
12. Åström KJ, Wittenmark B (eds) (2008) Adaptive control. Dover Publications, New York, NY
13. Augustine (2014). Confessions (Trans.). CreateSpace Independent Publishing Platform, North Charleston, SC. http://faculty.georgetown.edu/jod/augustine/Pusey/book01
14. Auyang SY (2008) Foundations of complex-system theories in economics, evolutionary biology, and statistical physics. Cambridge University Press, Cambridge, UK
15. Aven T (2008) Risk analysis assessing uncertainties beyond expected values and probabilities. Wiley, New York, NY
16. Awodey S, Coquand T, Voevodsky V (eds) (2013) Homotopy type theory: univalent foundations of mathematics. Univalent Foundations Program, Institute for Advanced Study. http://homotopytypetheory.org/book/
17. Baker GP, Hacker PMS (eds) Wittgenstein: understanding and meaning: Volume 1 of an analytical commentary on the philosophical investigations, Part II: Exegesis ß1-184, 2nd edn. Blackwell, Malden, MA
18. Bar-Yam Y (1997) Dynamics of complex systems. Studies in nonlinearity. Addison-Wesley, Reading, MA

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5

19. Bar-Yam Y (2003) The when systems engineering fails - toward complex systems engineering. In: Proceedings of the 2003 IEEE international conference on systems, man & cybernetics, vol 2, pp 16–31. Washington, DC
20. Barabasi AL (2016) Network science. Cambridge University Press, Cambridge, UK
21. Bargiela A, Pedrycz W (2003) Granular computing: an introduction. Kluwer Academic Publishers, Dordrecht
22. Barker R (1997) Case*Method: tasks and deliverables, Revised edn. Addison-Wesley, New York, NY
23. Barrat A, Barthélemy M, Vespignani A (2008) Dynamical processes on complex networks. Cambridge University Press, Cambridge, UK
24. Barwise J (1972) Absolute logics and $l_{\infty,\omega}$. Ann Math Logic 4:309–340
25. Barwise J, Cooper R (1981) Generalized quantifiers and natural language. Linguist Philos 4(2):159–219
26. Barwise J, Seligman J (1997) Information flow: the logic of distributed systems. Cambridge tracts in theoretical computer science. Cambridge University Press, Cambridge, UK
27. Bazan J (2008) Hierarchical classifiers for complex spatio-temporal concepts. Trans Rough Sets: J Subline 9:474–750 (LNCS 5390; Springer, Heidelberg)
28. Bazan J, Buregwa-Czuma S, Jankowski A (2013) A domain knowledge as a tool for improving classifiers. Fund Inf 127(1–4):495–511
29. Bazan J, Nguyen HS, Nguyen SH, Synak P, Wróblewski J (2000) Rough set algorithms in classification problems. In: Polkowski L, Lin TY, Tsumoto S (eds) Rough set methods and applications: new developments in knowledge discovery in information systems. Studies in Fuzziness and Soft Computing, vol. 56. Springer-Verlag/Physica-Verlag, Heidelberg, pp 49–88
30. Bazan JG, Nguyen HS, Peters JF, Skowron A, Szczuka M (2003) Rough set approach to pattern extraction from classifiers. In: Proceedings of the workshop on rough sets in knowledge discovery and soft computing at ETAPS 2003, April 12–13, Warsaw, 2003, pp 20–29
31. Bazan JG, Nguyen HS, Skowron A, Szczuka M (2003) A view on rough set concept approximation. In: Proceedings of the 9th international conference on rough sets, fuzzy sets, data mining, and granular computing (RSFDGrC'2003), Chongqing, China, May 26–29, 2003, Lecture notes in computer science, vol 2639, pp 181–188. Springer, Heidelberg
32. Bazan JG, Peters JF, Skowron A (2017) Behavioral pattern identification through rough set modelling. In: Ślęzak et al. [472], pp 688–697
33. Bazan JG, Skowron A (2017) Classifiers based on approximate reasoning schemes. In: Dunin-Kęplicz et al. [121], pp 191–202
34. Bazan JG, Skowron A, Swiniarski R (2005) Rough sets and vague concept approximation: from sample approximation to adaptive learning. Trans Rough Sets 5: J Subline, LNCS, vol 4100 pp 39–62. Springer, Heidelberg
35. Beckett D, Hager P (eds) (2002) Life, work and learning: practice in postmodernity. Routledge, London
36. Behnke S (2003) Hierarchical neural networks for image interpretation, vol 2766. Lecture notes in computer science. Springer, Heidelberg
37. Bell DA (1979) Frege's theory of judgement. Oxford University Press, New York, NY
38. Bell JL (2011) Set theory Boolean-valued models and independence proofs. Oxford logic guides, vol 47. Oxford University Press, New York, NY
39. Bell JL (2012) Types, sets, and categories, vol 6 of Gabbay et al. [150]
40. Bello R, Falcón R, Pedrycz W (eds) (2010) Granular computing: at the junction of rough sets and fuzzy sets. Studies in fuzziness and soft computing, vol 234. Springer, Heidelberg
41. Ben-Jacob E (1998) Bacterial wisdom, Gödel's theorem and creative genomic webs. Physica A 248:57–76
42. Ben-Jacob E (2009) Learning from bacteria about natural information processing. Annals of the New York Academy of Sciences: Natural genetic engineering and natural genome editing, vol 1178, pp 79–90

43. Bench-Capon TJM, Dunne PE (2007) Argumentation in artificial intelligence. Artif Intell 171:619–641
44. Bennett CH, Landauer R (1985) Fundamental physical limitations of the computational process. Sci Am 253:48–56
45. van Benthem J (2014) Logical dynamics of information and interaction. Cambridge University Press, Cambridge, UK
46. van Benthem J, Heinzmann G, Rebuschi M, Visser H (eds) (2009) The age of alternative logics: assessing philosophy of logic and mathematics today. Logic, epistemology, and the unity of science, Springer, Heidelberg
47. van Benthem J, Pacuit E, Roy O (2011) Toward a theory of play: a logical perspective on games and interaction. Games 2(1):52–86
48. Beziau J-Y (2005) Logica universalis: towards a general theory of logic. Birkhäuser, Basel
49. Béziau J-Y (ed) (2012) Universal logic: an anthology from Paul Hertz to Dov Gabbay. Birkhauser, Basel
50. Bianchini M, Maggini M, Jain LC (eds) (2013) Handbook on neural information processing. Series Intelligent systems reference library. Springer, Heidelberg
51. Birkhoff G (1967) Lattice theory (3rd ed), AMS Colloquium Publications, vol 25. American Mathematical Society, Providence, RI
52. Blanchard BS (2008) System engineering management. Wiley, Hoboken, NJ
53. Blass A (1984) Existence of bases implies the axiom of choice. Contemp Math 31:31–33
54. Bloch M, Blumberg S, Laartz J (2012) Delivering large-scale it projects on time, on budget, and on value. http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value
55. Bloom SL, Brown D, Suszko R (1970) Some theorems on abstract logics. In: Béziau [49], pp 257–262
56. Blume L, Durlauf S (eds) (2005) The economy as an evolving complex system, III. Current perspectives and future directions. Santa Fe Institute Studies on the Sciences of Complexity, Oxford University Press, Oxford, UK
57. Boardman A, Greenberg D, Vining A, Weimer D (2010) Cost-benefit analysis: concepts and practice, 2nd edn. Prentice Hall, Upper Saddle River, NJ
58. Boccalettia S, Latorab V, Morenod Y, Chavezf M, Hwang D-U (2006) Complex networks: structure and dynamics. Phys Rep 424:175–308
59. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence. From natural to artificial systems. Oxford University Press, Oxford, UK
60. Boniecki Z (1991) Poltax od środka. Computer World PL 17, 18 Sept 1991 (in Polish)
61. Boole G (1948) The mathematical analysis of logic (1847) (reprinted by Philosophical Library). Philosophical Library, New York, NY
62. Boole G (1954) An investigation of the laws of thought (1854) (reprinted by Dover Books). Dover Books, New York, NY
63. Borel E (1972) Oeuvres de Emile Borel. Ed. du Centre National de la Recherche Scientifique, Paris, 1972 (4 vols)
64. Bower JM, Bolouri H (eds) (2001) Computational modeling of genetic and biochemical networks. MIT Press, Cambridge, MA
65. Box GEP, Draper NR (1987) Empirical model building and response surfaces. Wiley, New York, NY
66. Boy GA (ed) (2011) The handbook of human-machine interaction. Ashgate, Farnham UK
67. Brandt F, Conitzer V, Endriss U, Lang J, Procaccia AD (eds) (2016) Handbook of computational social choice. Cambridge University Press, New York, NY
68. Brooks FP (1975) The mythical man-month: essays on software engineering. Addison-Wesley, Boston (extended Anniversary Edition in 1995)
69. Brooks FP (1986) No silver bullet—essence and accident in software engineering. In: Kugle HJ (ed) Proceedings of the IFIP tenth world computing conference, Amsterdam, Elsevier, pp 1069–1076

70. Brouwer L (1975) Collected works I: philosophy and foundations of mathematics (Heyting A, ed). North-Holland, Amsterdam
71. Brouwer L (1907) Over de grondslagen der wiskunde, PhD thesis, 1907. In: Collected works I: philosophy and foundations of mathematics, (English transl) [70], pp 11–101
72. Brown F (1990) Boolean reasoning. Kluwer Academic Publishers, Dordrecht
73. Buchanan B, Shortliffe E (1984) The MYCIN experiments of the stanford heuristic programming project. Addison-Wesley, Reading, MA
74. Bums LC (1991) Vagueness. An investigation into natural languages and the sorites paradox Kluwer, Dordrecht
75. Burgin M (2005) Super-recursive algorithms. Springer, Heidelberg
76. Burnham D (ed) (2000) An introduction to Kant's critique of judgement. Edinburgh University Press
77. Chang HKC (2012) Model theory. Dover Publications, Mineola, NY
78. Caldarelli G (2007) Scale-free networks. Complex webs in nature and technology. Oxford University Press, New York, NY
79. Caldarelli G, Vespignani A (2007) Large scale structure and dynamics of complex networks. From information technology to finance and natural science. World Scientific, Singapore
80. Calvert GA, Spence C, Stein BE (eds) (2004) The handbook of multisensory processes. The MIT Press, Cambridge, MA
81. Calvo P, Symons J (eds) (2008) The architecture of cognition. Rethinking Fodor and Pylyshyn's systematicity challenge. The MIT Press, Cambridge, MA
82. Cambria E, White B (2014) Jumping NLP curves: a review of natural language processing research. IEEE Comput Intell Mag 9:48–57
83. Cantor G (1895) Beiträge zur Begründung der transfiniten Mengenlehre. Mathematische Annalen 46:481–512
84. Chen G, Wang X, Li X (2015) Fundamentals of complex networks: models. Structures and dynamics. Wiley, Hoboken, NJ
85. Cherifi H, Gonçalves B, Menezes R, Sinatra R (eds) Complex networks VII. In: Proceedings of the 7th workshop on complex networks (CompleNet 2016). Studies in computational intelligence, vol 644. Springer, Heidelberg
86. Chilton P (2014) Language, space and mind. The conceptual geometry of linguistic meaning. Cambridge University Press, Cambridge, UK
87. Chiong R (2010) Intelligent systems for automated learning and adaptation: emerging trends and applications. IGI Global, Herehey, PA
88. Church A (1936) A note on the Entscheidungsproblem. J Symbol Logic 1:40–41
89. Cohen R, Havlin S (2010) Complex networks. Structure, robustness and function. Cambridge University Press, New York, NY
90. Commonwealth of Massachusetts (2016) Independent verification and validation (iv&v). https://wiki.state.ma.us/confluence/download/attachments/69894234/CommonWay
91. Cooper NG, Eckhardt R, Shera N (eds) (1989) From cardinals to chaos: reflection on the life and legacy of Stanislaw Ulam. Cambridge University Press, Cambridge, UK
92. Cox LAJ (2009) Risk analysis of complex and uncertain systems. Springer Science & Business Media, New York, NY
93. Craddock A, Roberts B, Godwin J, Richards K, Tudor D (2014) The DSDM Agile project framework. https://www.agilebusiness.org/resources/dsdm-handbooks/the-dsdm-agile-project-framework-2014-onwards
94. Crowe M (1975) Ten 'laws' concerning patterns of change in the history of mathematics. Historia Mathematica 2:161–166
95. Csermely P (2006) Weak links. Stabilizers of complex systems from proteins to social networks. Springer, Heidelberg
96. Curry H (1952) Lessons on algebraic logic. Introduction and Chapters I and II. In: Béziau [49], pp 133–162
97. D'Agostino G, Scala A (eds) (2014) Networks of networks: the last frontier of complexity. Springer, Heidelberg

98. Darley V, Outkin AV (2007) A NASDAQ insights on a major market from the science of complex adaptive systems. World Scientific, Singapore
99. Dauben J (2017) Conceptual revolutions and the history of mathematics: Two studies in the growth of knowledge. In: Revolutions in mathematics [162], pp 49–71
100. Davis A (1995) 201 principles of software development. McGraw-Hill, New York, NY
101. Debbabi M, Hassaine F, Jarraya Y (2010) Verification and validation in systems engineering: assessing UML/SysML design models. Springer, Heidelberg
102. Dehmer M (ed) (2011) Structural analysis of complex networks. Birkhäuser, Springer, Heidelberg
103. Dehmer M, Emmert-Streib F (eds) (2009) Analysis of complex networks. From biology to linguistics. Wiley-VCH Verlag GmbH & Co. KGaA, Heidelberg
104. Dehmer M, Emmert-Streib F, Mehler A (eds) (2011) Towards an information theory of complex networks. Statistical methods and applications. Birkhäuser, Springer, Heidelberg
105. Deja R, Skowron A (2002) On some conflict models and conflict resolution. Rom J Inf Sci Technol 5(1–2):69–82
106. Delligatti L (2013) SysML distilled: a brief guide to the systems modeling language. Addison-Wesley Professional, Upper Saddle River, NJ
107. Department of Defense Systems Management College (2001) Systems engineering fundamentals. CreateSpace Independent Publishing Platform, Fort Belvoir, VA
108. Department of Health and Human Services (2010) Department of health and human services enterprise performance life cycle for iv&v. http://www.hhs.gov/ocio/eplc/EPLC
109. Desai A (2005) Adaptive complex enterprises. Commun ACM 45:32–35
110. Desai R, Kapral R (2009) Dynamics of self-organized and self-assembled structures. Cambridge University Press, Cambridge, UK
111. Descartes R (2012) Discourse on method and meditations on first philosophy. CreateSpace Independent Publishing Platform, Seattle, Washington
112. Deutsch D, Ekert A, Lupacchini R (2000) Machines, logic and quantum physics. Neural Comput 6:265–283
113. Diaconescu R (1975) Axiom of choice and complementation. Proc Am Math Soc 51:176–178
114. van Ditmarsch H, Lang J, Ju S (eds) Logic, rationality, and interaction. In: Proceedings of the third international workshop (LORI 2011), Guangzhou, China, 10–13 Oct 2011, Lecture notes in computer science, vol 6953. Springer, Heidelberg
115. Dodig-Crnkovic G (2011) Significance of models of computation, from Turing model to natural computing. Minds Mach 21:301–322
116. Doherty ME (ed) (1996) Social judgement theory: a special double issue of the journal thinking and reasoning. Psychology Press
117. Doherty P, Łukaszewicz W, Skowron A, Szałas A (2006) Knowledge engineering: a rough set approach. Studies in fuzziness and soft computing, vol 202. Springer, Heidelberg
118. Dubois D, Godo L, Prade H, Esteva F (2005) An information-based discussion of vagueness. In: Cohen H, Lefebvre C (eds) Handbook of categorization in cognitive science. Amsterdam, Elsevier, pp 892–913
119. Duda R, Hart P, Stork R (2002) Pattern classification. Wiley, New York, NY
120. Dummett M (2000) The elements of intuitionism. Oxford University Press, New York, NY
121. Dunin-Kęplicz B, Jankowski A, Skowron A, Szczuka M (eds) (2005) Monitoring, security, and rescue tasks in multiagent systems (MSRAS'2004). Advances in soft computing, vol 28. Springer, Heidelberg
122. Edghill EM (2007) On Interpretation by Aristotle (trans). The University of Adelaide, Adelaide. http://ebooks.adelaide.edu.au/a/aristotle/
123. Ehrenfeucht A, Marek V, Srebrny M (eds) (2008) Andrzej Mostowski and foundational studies. IOS Press, Amsterdam
124. Ehrentreich N (ed) (2008) Agent based modeling. Artificial stock market model revisited. The Santa Fe Institute. Lecture notes in economics and mathematical systems, vol 602. Springer, Heidelberg

125. Einstein A (1921) Geometrie und Erfahrung (Geometry and Experience). Julius Springer, Berlin, 1921. (An expanded form of an Address to the Prussian Academy of Sciences in Berlin on January 27th, 1921)
126. Eliasmith C (2013) How to build a brain. A neural architecture for biological cognition. Oxford University Press, New York, NY
127. Ellis NC, Larsen-Freeman D (eds) (2009) Language as a complex adaptive system. University of Michigan, MI, Language Learning Research Club
128. Engel A (2010) Verification, validation, and testing of engineered systems. Wiley, Hoboken, NJ
129. Engel A (2010) Verification, validation and testing of engineered systems. Wiley series in systems engineering and management, Wiley-Blackwell, New York, NY
130. Estep M (2014) Self-organizing natural intelligence: issues of knowing, meaning, and complexity. Springer-Verlag, Heidelberg
131. Evans J (2007) Hypothetical thinking: dual processes in reasoning and judgement. Essays in cognitive psychology. Psychology Press, New York, NY
132. Eveleens JL, Verhoef C (2010) The rise and fall of the chaos report figures. IEEE Softw 27:30–36
133. Fahle M, Poggio T (2002) Perceptual learning. MIT Press, Cambridge, MA
134. Fan J, Han F, Liu H (2014) Challenges of big data analysis. Natl Sci Rev 1:293–314
135. Feigenbaum EA, McCorduck P (1983) The fifth generation—Artificial intelligence and Japan's computer challenge to the world. Addison-Wesley, Reading, MA
136. Feistel R, Ebeling W (2011) Physics of self-organization and evolution. Wiley-VCH Verlag, Weinheim, Germany
137. Feldman JA (2006) From molecule to metaphor. A neural theory of language. The MIT Press, Cambridge, MA
138. Feng J, Jost J, Minping Q (eds) (2007) Network: from biology to theory. Springer, Berlin
139. Feynman R (1967) The character of physical law. MIT Press, Cambridge, MA
140. Firesmith D (2014) Common system and software testing pitfalls and ways to prevent and mitigate them: descriptions, symptoms, consequences, causes, and recommendations. Addison-Wesley Professional, Upper Saddle River, NJ
141. Firesmith D, Capell P, Hammons C, Latimer D, Merendino T, Falkenthal D (2008) The method framework for engineering system architectures. Auerbach Publications, Boca Raton, FL
142. Fitting M (1969) Intuitionistic logic model theory and forcing. Studies in logic and the foundations of mathematics. North-Holland, Amsterdam
143. Fitting M (2002) Types, tableaus, and Gödel's God. Trends in logic, Springer, Heidelberg
144. Fourman M, Scott D (1979) Sheaves and logic. In: Fourman M, Mulvey C, Scott D (eds) Applications of sheaves. Lecture notes in mathematics, vol 753. Springer, Heidelberg, pp 302–401
145. Fraenkel A, Bar-Hillel Y, Levy A (1973) Foundations of set theory. North-Holland, Amsterdam
146. Frankish K, Ramsey W (eds) (2012) The Cambridge handbook of cognitive science. Cambridge University Press, Cambridge
147. Frege G (1903) Grundgesetzen der Arithmetik, 2. Verlag von Hermann Pohle, Jena
148. Friesz TL, Bernstein D (2016) Foundations of network optimization and games. Springer, Heidelberg
149. Fu X, Small M, Chen G (2014) Propagation dynamics on complex networks. Methods and stability analysis. Models. Wiley, Chichester, UK
150. Gabbay DM, Kanamori A, Woods J (eds) (2012) Sets and extensions in the twentieth century. Handbook of the history of logic, vol 6. Elsevier, Amsterdam
151. Gallagher S, Schmicking D (eds) (2010) Handbook of phenomenology and cognitive science. Springer, Heidelberg
152. Gamma E, Helm R, Johnson R, Vlissides J (1994) Design patterns: elements of reusable object-oriented software. Addison-Wesley Professional, New York, NY

153. Gammage M (2012) It project failure: how did we end up here? http://www.processartisan.com/2012/10/29/
154. Gärdenfors P (2014) The geometry of meaning. Semantics based on conceptual spaces. The MIT Press, Cambridge, MA
155. Garling DJH (ed) (1987) A course in Galois theory. Cambridge University Press, New York, NY
156. Gauss CF (1831) Letter to Schumacher, 12 July 1831. Werke 8:215–218
157. Geach P, Black M (eds) (1960) Translations from the philosophical writings of Gottlob Frege. Blackwell, Oxford, UK
158. Gell-Mann M (1994) The quark and the jaguar—adventures in the simple and the complex. Brown and Co., London
159. Gentner D, Holyoak KJ, Kokinov BK (eds) (2001) The analogical mind. Perspectives from cognitive science. The MIT Press, Cambridge, MA
160. George ML, Maxey J, Rowlands D, Price M (2014) The lean six sigma pocket toolbook: a quick reference guide to 100 tools for improving quality and speed. McGraw-Hill Education
161. Gershenson C (ed) (2007) Design and control of self-organizing systems. CopIt ArXives, Mexico City, Boston, MA
162. Gillies D (1992) Revolutions in mathematics. Oxford Science Publications, The Calderon Press, Oxford University, Oxford
163. Givant SR, McKenzie RN (eds) (1986) The collected papers of Alfred Tarski, 4 vols. Birkhäuser, Basel
164. Glass R (2003) Facts and fallacies of software engineering. Addison-Wesley Professional, Upper Saddle River, NJ
165. Glivenko V (1929) Sur quelques points de la logique de M. Brouwer. Academie Royale de Belgique, Bulletins de la Classe des Sciences 15(5):183–188
166. Glock HJ (1996) A Wittgenstein dictionary. Wiley-Blackwell, Oxford, UK
167. Godel K (1931/2) Zur intuitionistischen Arithmetik und Zahlentheorie. In: Heft M (ed) Ergebnisse eines mathematischen Kolloquiums, vol 4. Wien, pp 34–38
168. Gödel K (1931) Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. I. Monatshefte für Mathematik und Physik 38:173–198
169. Goldin D, Smolka S, Wegner P (eds) (2006) Interactive computation: the new paradigm. Springer, Heidelberg
170. Gordon DM (2010) Ant encounters. Interaction networks and colony behavior. Princeton University Press, Princeton, NJ
171. Gordon E (ed) Integrative neuroscience. Bringing together biological, psychological and clinical models of the human brain. Harwood Academic Publishers, Amsterdam
172. Gorenstein D (2014) The classification of finite simple groups: Volume 1: groups of noncharacteristic 2 type. Springer, New York, NY
173. Gorenstein D, Lyons R, Solomon R (1994) The classification of the finite simple groups. American Mathematical Society, Providence, RI
174. Granström JG (2009) Treatise on intuitionistic type-theory. Springer, Heidelberg
175. Greenspan A (2008) Testimony of Alan Greenspan on the role of federal regulators in the financial crisis. http://www.law.du.edu/documents/corporate-governance/legislation/preliminary-transcript-crisis.pdf
176. Greenspan A (2013) The map and the territory: risk, human nature, and the future of forecasting. Penguin Press HC, New York, NY
177. Greenspan A (2014) Transcription of the map and the territory: risk, human nature, and the future of forecasting, American Enterprise Institute. https://www.aei.org/wp-content/uploads/2014/02/-the-map-and-the-territory_112315925967.pdf
178. Gros C (2008) Complex and adaptive dynamical systems. A primer. Springer, Heidelberg
179. Gross T, Sayama H (eds) Adaptive networks. Theory, models and applications. Springer, Heidelberg
180. Grossi D, Pigozzi G (2012) Introduction to judgment aggregation. In: Goranko V, Bezhanishvili N (eds) Lectures on logic and computation: ESSLLI 2010/2011 Lecture notes in computer science, vol 7388. Springer, Heidelberg, pp 160–209

181. Grossi D, Pigozzi G (2014) Judgment aggregation: a primer. Synthesis lectures on artificial intelligence and machine learning, vol 27. Morgan & Claypool, San Rafael, CA
182. Grossi D, Roy O, Huang H (eds) Logic, rationality, and interaction. In: Proceedings of the 4th international workshop, (LORI 2013), Hangzhou, China, 9–12 Oct 2013. Lecture notes in computer science, vol 8196. Springer, Heidelberg
183. Hager P, Halliday J (2006) Recovering informal learning wisdom. Judgement and community. Springer, Heidelberg
184. Haken H (2006) Information and self-organization a macroscopic approach to complex systems. Springer, Berlin
185. Halmos P (1968) Mathematics as a creative art. The American Scientist 56 (1968) 375–389. (lecture presented by Halmos in Edinburgh on 7 May 1973 and published in the Royal Society of Edinburgh Year Book 1973 (Session 1971–1972))
186. Hamblin C (1970) Fallacies. Methuen, London
187. Hamblin C (1987) Imperatives. Blackwell, Oxford, UK
188. Hardman D, Macchi L (2003) Thinking: psychological perspectives on reasoning, judgment and decision making. Wiley, New York, NY
189. Harnad S (1990) The symbol grounding problem. Physica D 42:335–346
190. Press Harvard Business School (2006) SWOT analysis I: looking outside for threats and opportunities. Harvard Business School Publishing Corporation, Boston, MA
191. Press Harvard Business School (2006) SWOT analysis II: looking inside for strengths and weaknesses. Harvard Business School Publishing Corporation, Boston, MA
192. Haskins C (2012) Incose systems engineering handbook version 3.2.2. A guide for life cycle processes and activities. International Council on Systems Engineering, Philadelphia, PA
193. Hastie T, Tibshirani R, Friedman JH (2001) The elements of statistical learning: data mining, inference, and prediction. Springer-Verlag, Heidelberg
194. Haugeland J (ed) (1997) Mind design II: philosophy, psychology, and artificial intelligence. A Bradford Book, Chester, NJ
195. Hausser R (2006) A computational model of natural language communication interpretation, inference, and production in database semantics. Springer, Heidelberg
196. Hawking S, Penrose WR (1996) The nature of space and time. Princeton University Press, Princeton, NJ
197. Hawkins JA, Gell-Mann M (eds) (1992) The evolution of human languages. Addison-Wesley, Redwood City, CA
198. Helbig H (2006) Knowledge representation and the semantics of natural language. Springer, Heidelberg
199. Helbing D (ed) (2012) Social self-organization agent-based simulations and experiments to study emergent social behavior. Springer, Heidelberg
200. Helbing D (2013) Globally networked risks and how to respond. Nature 497:51–59
201. Heller M (1990) The ontology of physical objects. Four dimensional hunks of matter. Cambridge University Press, Cambridge, UK. Cambridge Studies in Philosophy
202. Heyting A (1971) Intuitionism: an introduction. North-Holland, Amsterdam
203. Higgs D (1973) A category approach to Boolean-valued set theory. University of Waterloo, Waterloo
204. Highsmith J (2009) Agile project management: creating innovative products, 2nd edn. The agile software development series, Addison-Wesley Professional, Upper Saddle River, NJ
205. Hilbert D (1983) Über das Unendliche. Mathematische Annalen 95 (1926):161–190. (trans: Putnam E, In Massey GJ). In: Benacerraf P, Putnam H (eds) Philosophy of mathematics: selected readings, 2nd edn. Cambridge University Press, Cambridge, UK, pp 183–201
206. Hillis D (ed) (1985) The connection machine. MIT Press, Cambridge, MA
207. van der Hoek W, Holliday WH, fang Wang W (eds) Logic, rationality, and interaction. In: Proceedings of the 5th international workshop (LORI 2015), Taipei, Taiwan, 28–31 Oct 2015. Lecture notes in computer science, vol 9394. Springer, Heidelberg
208. Hoekstra AG, Kroc J, Sloot PM (eds) (2010) Simulating complex systems by cellular automata. Springer, Heidelberg

209. Holland JH (1986) Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine learning: an artificial intelligence approach, vol II. Morgan Kaufmann, San Mateo, CA, pp 593–623
210. Holland JH (2014) Signals and boundaries. Building blocks for complex adaptive systems. The MIT Press, Cambridge, MA
211. Höller J, Tsiatsis V, Mulligan C, Avesand SKS, Boyle D (2014) From machine-to-machine to the internet of things. Introduction to a new age of intelligence. Academic Press, Elsevier, Waltham, MA
212. Holme P, Saramäki J (eds) (2011) Temporal networks. Springer, Heidelberg
213. Hooker C (2011) Introduction to philosophy of complex systems part a: towards a framework for complex systems. In: Hooker C (ed) Handbook of the philosophy of science, vol 10, Philosophy of Complex Systems, Elsevier, Amsterdam, pp 3–90
214. Hooker C (2011) Introduction to philosophy of complex systems part b: Scientific paradigm + philosophy of science for complex systems: a first presentation c. 2009. In: Hooker C (ed) Handbook of the philosophy of science, vol 10. Philosophy of complex systems. Elsevier, Amsterdam pp 841–909
215. Howard P, Rubin J (1998) Consequences of the axiom of choice, mathematical surveys and monographs, vol 59. American Mathematical Society, Providence, RI
216. Huget M-P (ed) (2003) Communication in multiagent systems: agent communication languages and conversation policies. Springer, Heidelberg
217. Huhns MN, Singh MP (1998) Readings in agents. Morgan Kaufmann, San Mateo
218. Hüllermeier E (2007) Case-based approximate reasoning. Springer, Heidelberg
219. Imai M, Kaizen G (2012) A commonsense approach to a continuous improvement strategy, 2nd edn. McGraw-Hill Professional, New York, NY
220. Inuiguchi M, Hirano S, Tsumoto S (eds) (2003) Rough set theory and granular computing. Studies in fuzziness and soft computing, vol 125. Springer, Heidelberg
221. ISO/IEC/IEEE 42010 (2015) A conceptual model of architecture description of ISO/IEC/IEEE 42010. https://www.opengroup.org/togaf/
222. Jackendoff R (2002) Foundations of language. Brain, Meaning, Grammar. Oxfrod University Press, Oxford, UK
223. Jacobson C (2014) The importance of cyber-physical systems for industry. ERCIM News 94:4–4
224. Jagadish H, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C (2014) Big data and its technical challenges. Commun ACM 57:86–94
225. Jankowski A (1976) Formalizacja pewnych zależności między logikami (A formalization of some relationships between logics). Master's Thesis, Faculty of Mathematics, Informatics, and Mechanics, Univeristy of Warsaw (in Polish)
226. Jankowski A (1980) Embedding of logical structures. Forschungsinstitut Oberwolfach Tagensberich 18. Mathematische Logic 20.04 bis 26.04 18:3–5
227. Jankowski A (1980) On closure spaces of filters in complete lattices of sets. J Symbol Logic 45(2): 394–396 (Logic Colloquium 78, Mons)
228. Jankowski A (1980) Zanurzenia Struktur Logicznych, (Embedding of logical structures) Ph. D. Thesis 1978, vol. 10 seria B, February 1980. Institute of Mathematics Polish Academy of Sciences, Warsaw, 1980 (in Polish)
229. Jankowski A (1982) An alternative characterization of elementary logic. Bull Acad Polon Sci, Sér Sci Math 30:9–13
230. Jankowski A (1982) A characterization of the closed subsets of an <alpha, beta>-closure space using <alpha, beta>-base. Bull Acad Polon Sci, Sér Sci Math 30:1–8
231. Jankowski A (1983) Recursive closure spaces. J Symbol Logic 48(2):1220–1220 (Logic Colloquium 81, Marseille)
232. Jankowski A (1984) Absolute retracts and absolute extensors in the category of closure spaces. In: Guzicki W, Marek W, Pelc A, Rauszer C (eds) Proceedings of a conference held in September 1981 at Jadwisin, near Warsaw. Open Days in Model Theory and Set Theory. University of Leeds, Leeds, pp 135–143

233. Jankowski A (1984) Conjunctions in closure spaces. Studia Logica 43:341–351
234. Jankowski A (1984) Modifications of Scott's theorem. J Symbol Logic 48(2):693–694 (Logic Colloquium 82, Florence)
235. Jankowski A (1985) Disjunctions in closure spaces. Studia Logica 44:11–24
236. Jankowski A (1985) Galois structures. Studia Logica 44:109–124
237. Jankowski A (1985) Universality of the closure space of filters in the algebra of all subsets. Studia Logica 44:1–9
238. Jankowski A (1986) Retracts of the closure space of filters in the lattice of all subsets. Studia Logica 45:135–154
239. Jankowski A (1986) Some modifications of Scott's theorem on injective spaces. Studia Logica 45:155–166
240. Jankowski A (1996) Zamek w oceanie doświadczeń. Teleinfo 2 (in Polish)
241. Jankowski A (1996) Zamek z NIKąd. Teleinfo 4 (in Polish)
242. Jankowski A, Achinger J (1986) On decidable consequence operators. Studia Logica 45:415–424
243. Jankowski A, Michalewicz Z, Ras Z, Shoff D (1989) Issues on evolution programming. In: Janicki R, Koczkodaj WW (eds) Computing and information, North-Holland, Amsterdam pp 459–463
244. Jankowski A, Skowron A (2007) A Wistech paradigm for intelligent systems. Trans Rough Sets: J Subline 6:94–132 (LNCS, vol 4374. Springer, Heidelberg)
245. Jankowski A, Skowron A (2008) Logic for artificial intelligence: Rasiowa - Pawlak school perspective. In: Ehrenfeucht A, Marek VM, Srebrny M (eds) Andrzej Mostowski and foundational studies. Amsterdam, IOS Press, pp 106–143
246. Jankowski A, Skowron A (2017) Wisdom granular computing. In: Pedrycz et al. [394], pp 329–345
247. Jankowski A, Skowron A (2009) Wisdom technology: a rough-granular approach. In: Marciniak M, Mykowiecka A (eds) Aspects of natural language processing essays dedicated to Leonard Bolc on the occasion of His 75th birthday. Lecture notes in computer science, vol 5070. Springer, Heidelberg, pp 3–41
248. Jankowski A, Skowron A, Swiniarski R (2014) Interactive complex granules. Fundamenta Informaticae 133(2–3):181–196
249. Jankowski A, Skowron A, Swiniarski R (2013) Interactive complex granules. In: Szczuka M, Czaja L, Kacprzak M (eds) proceedings of the 22nd international workshop on concurrency, specification and programming (CS&P'2013), Warsaw, Poland, 25–27 Sept 2013, CEUR workshop proceedings, RWTH Aachen University, vol 1032, pp 206–218
250. Jankowski A, Skowron A, Swiniarski R (2014) Perspectives on uncertainty and risk in rough sets and interactive rough-granular computing. Fundamenta Informaticae 129:69–84
251. Jankowski A, Skowron A, Swiniarski RW (2013) Interactive computations: toward risk management in interactive intelligent systems. In: Maji P, Ghosh A, Murty MN, Ghosh K, Pal SK (eds) Proceedings of the 5th international conference on pattern recognition and machine intelligence (PReMI 2013), Kolkata, India, 10–14 Dec 2013. Lecture Notes in Computer Science, vol 8251. Springer, Heidleberg, pp 1–12
252. Jankowski A, Skowron A, Swiniarski RW (2013) Interactive rough-granular computing in wisdom technology. In: Yoshida T, Kou G, Skowron A, Cao J, Hacid H, Zhong N (eds) Proceedings of the A 9th international conference on active media technology (AMT 2013), Maebashi, Japan, 29–31 Oct 2013. Lecture notes in computer science, vol 8210. Springer, Heidelberg, pp 1–13
253. Jankowski A, Skowron A, Wasilewski P (2012) Risk management and interactive computational systems. J Adv Math Appl 1:61–73
254. Jankowski A, Zawadowski M (1985) Sheaves over Heyting lattices. Studia Logica 44:237–356
255. Liker Jeffrey (2004) The Toyota Way: 14 management principles from the world's greatest manufacturer. McGraw-Hill, New York, NY
256. Johnson J (2002) Build only the features you need (standish group). http://martinfowler.com/articles/xp2002.html

257. Johnson J (2016) My life is failure. The Standish Group, Boston, MA
258. Johnson-Laird PN (2008) How we reason. The MIT Press, Cambridge, MA
259. Johnstone PT (1977) Topos theory. Academic Press, New York, NY
260. Johnstone PT (2002) Sketches of an elephant. A topos theory compendium, vol 1. T. Woolmer, Oxford
261. Juran J, De-Feo J (2010) Juran's quality handbook: the complete guide to performance excellence. McGraw-Hill Education, New Jork
262. Kacprzyk J, Pedrycz W (2015) Springer handbook of computational intelligence. Springer, Heidelberg
263. Kahneman D (2011) Thinking. Fast and slow. Farrar, Straus and Giroux, New York, NY
264. Kaplan R, Anderson S (2009) What-if analysis and activity-based budgeting forecasting resource demands. In: Kaplan R, Anderson S (eds) Time-driven activity-based costing: a simpler and more powerful path to higher profits. Harvard Business Review Press (April 4, 2007), Boston, MA, pp 85–106. http://www.cwj24.cz/doc/Time-Driven-ABC.pdf
265. Kari L, Rozenberg G (2008) The many facets of natural computing. Commun ACM 51:72–83
266. Ishikawa Karoru (1988) What is total quality control? The Japanese Way. Prentice Hall, Upper Saddle River, NJ
267. Kawasaki S, Nguyen NB, Ho TB (2000) Hierarchical document clustering based on tolerance rough set model. In: Zighed DA, Komorowski HJ, Żytkow JM (eds) Proceedings of the 4th European conference on principles of data mining and knowledge discovery (PKDD 2000), Lyon, France, 13–16 Sept 2000. Lecture notes in computer science, vol 1910. Springer, Heidleberg, pp 458–463
268. Keefe R (2000) Theories of vagueness. Cambridge Studies in Philosophy, Cambridge, UK
269. Keefe R, Smith P (1997) Vagueness: a reader. MIT Press, Cambridge, MA
270. Kelly JE, Hamm S (2013) Smart machines. IBM's Watson and the era of cognitive computing. Columbia University Press, New York, NY
271. Kèpés F (ed) (2007) Biological networks. World Scientific, Singapore
272. Kiel F (2015) Return on character: the real reason leaders and their companies win. Harvard Business Review Press, Brighton, MA
273. Kleene SC (1936) General recursive functions of natural numbers. Mathematische Annalen 112:727–742
274. Kloesgen W, Żytkow J (eds) (2002) Handbook of knowledge discovery and data mining. Oxford University Press, Oxford
275. Knight C, Hurford MS-K JR (eds) The evolutionary emergence of language. social function and the origins of linguistic forms. Santa Fe Institute Studies on the Sciences of Complexity, Cambridge University Press, Cambridge, UK
276. Koehler DJ, Harvey N (2004) Blackwell handbook of judgment and decision making. Blackwell Publishing, Malden, MA
277. Kolmogorov AN (1925) On principe tertium non datur. Matematiceskij Sbornik 32:646–667
278. Kolmogorov AN (1957) On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition. Doklay Akademii Nauk USSR 14:953–956
279. Kossiakoff A, Sweet W, Seymour S, Biemer S (2011) Systems engineering principles and practice. Wiley-Interscience, Hoboken, NJ
280. Kotarski J (1993) Zamek na lodzie. Computer World PL 9 (in Polish, 1993)
281. Kowalkowski F, Laware G, Ulrich W, Poulin M, Connor MS (2015) Business and dynamic change: the arrival of business architecture. Future strategies. Lighthouse Point, FL
282. Kowalski R (1979) Logic for problem solving. Elsevier, North Holland, Amsterdam
283. Kowalski R (2003) A logic-based approach to conflict resolution. Report, Department of Computing, Imperial College (2003) 1–28. http://www.doc.ic.ac.uk/rak/papers/conflictresolution.pdf
284. Kowalski R (2011) Computational logic and human thinking: how to be artificially intelligent. Cambridge University Press, Cambridge, UK

285. KPMG Independent Project Monitoring (2014). http://www.doit.state.nm.us/docs/events/Independent%20Project%20Monitoring-%20SoNM%20EPMO%20PM%20Appreciation%20Event%20Nov%202014.pdf
286. Krasuski A, Jankowski A, Skowron A, Ślęzak D (2013) From sensory data to decision making: a perspective on supporting a fire commander. In: IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology, Atlanta, Georgia, USA, 17–20 Nov 2013. Workshop proceedings, pp 229–236
287. Kravtsov Y (ed) (1993) Limits of predictability. Springer, Heidelberg
288. Kriwaczek P (2010) Babylon: Mesopotamia and the birth of civilization. Atlantic Books, London, UK
289. Krohn W, Küppers G, Nowotny H (eds) (2008) Selforganization. Portrait of a scientific revolution. Kluwer, Springer, Dordrecht
290. Kuhn TS (1957) The Copernican revolution: planetary astronomy in the development of western thought. Harvard University Press, Cambridge, MA
291. Kuhn TS (1962) The structure of scientific revolutions. University of Chicago Press, Chicago, IL
292. Kuratowski K, Mostowski A (1976) Set theory, with an introduction to descriptive set theory. Polish Scientific Publishers PWN, Warszawa
293. Kyan M, Muneesawang P, Jarrah K, Guan L (2014) Unsupervised learning: a dynamic approach. IEEE press series on computational intelligence, Wiley-IEEE Press, Hoboken, New Jersey
294. Lacewing M (2015) Philosophy for A2: ethics and philosophy of mind. Routledge, New York, NY
295. Laird J (ed) (2012) The soar cognitive architecture. MIT Press, Cambridge, MA
296. Lambek JP (1986) Introduction to higher-order categorical logic. Cambridge studies in advanced mathematics, vol 7. Cambridge University Press, Cambridge, UK
297. Lamnabhi-Lagarrigue F, Di Benedetto MD, Schoitsch E (2014) Introduction to the special theme cyber-physical systems. Ercim News 94:6–7
298. Landau ID, Lozano R, M'Saad M, Karimi A (eds) Adaptive control. Algorithms, analysis and applications. Springer, Heidelberg
299. Landauer R (1984) Fundamental physical limitations of the computational process. Ann New York Acad Sci 426:161–170
300. Leibniz GW (2017) Dissertio de Arte Combinatoria. Leipzig, 1666
301. Leibniz GW (1982) New essays on human understanding (1705), (translated by Alfred Gideon Langley, 1896), (edited by Peter Remnant and Jonathan Bennett). Cambridge University Press, Cambridge UK
302. Leitner S, Wall F (eds) (2014) Artificial economics and self organization. Agent-based approaches to economics and social systems. Springer, Heidelberg
303. Lerner J, Wagner D, Zweig KA (eds) (1998) Algorithmics of large and complex networks. Design, analysis, and simulation. Springer, Heidelberg
304. Lewis M, Polvichai J, Sycara K, Scerri P (2011) Scaling-up human control for large UAV teams. In: Cooke N (ed) The human factors of remotely piloted vehicles. NY, Elsevier, New York, pp 237–250
305. Lewis R, Lewis A (1992) Independent verification and validation (New dimensions in engineering series). Wiley, New York, NY
306. Li M, Vitányi PM (2009) An introduction to Kolmogorov complexity and its applications. Texts in computer science. Springer, Heidelberg
307. Lin TY (1998) The discovery, analysis and representation of data dependencies in databases. In: Polkowski L, Skowron A (eds) Rough sets in knowledge discovery 1: methodology and applications. Studies in fuzziness and soft computing, vol 18. Physica-Verlag, Heidelberg, pp 107–121
308. Lin TY, Yao YY, Zadeh LA (eds) (2002) Data mining, rough sets and granular computing. Studies in fuzziness and soft computing, Vol 95. Physica-Verlag, Heidelberg

309. Lindström P (1966) First order predicate logic with generalized quantifiers. Theoria 32:166–195
310. Lindström P (1969) On extensions of elementary logic. Theoria 35:1–11
311. Liu J (2001) Autonomous agents and multi-agent systems: explorations in learning. Self-organization and adaptive computation. World Scientific Publishing, Singapore
312. Lloyd S (2006) Programming the universe: a quantum computer scientist takes on the cosmos. Alfred A. Knopf, New York, NY
313. Loritz D (1999) How the brain evolved language. Oxfrod University Press, Oxford, UK
314. Lü J, Yu X, Chen G, Yu W (eds) (2016) Complex systems and networks. dynamics, controls and applications. Springer, Heidelberg
315. Lu XB, Qin BZ (2011) Synchronization in complex networks. Nova Science Publishers, New York, NY
316. Luntley M (2003) Witgenstein: meaning and judgement. Wiley-Blackwell, Malden, MA
317. Mac Lane S, Birkhoff G (2010) Algebra. American Mathematical Soc., Providence, Rhode Island
318. Mac Lane S, Moerdijk I (1992) Sheaves in geometry and logic: a first introduction to topos theory. Science+Business Media Inc, New York, NY
319. MacKenzie IS (2013) Human-computer interaction: an empirical research perspective. Morgan Kaufmann, Burlington, MA
320. Maher SM (1902) Psychology: empirical and rational. Green and Co., New York, NY, Longmans
321. Mahony T (2012) Statement of work independent verification and validation (iv&v) services. http://www.google.pl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjyudrPx5XMAhVDCCwKHZA7BUcQFggqMAI&url=http
322. Maier M, Rechtin E (2009) The art of systems architecting. CRC Press LLC, Boca Raton, FL
323. Maimon O, Rokach L (eds) (2005) The data mining and knowledge discovery handbook. Springer, Heidelberg
324. Majer O, Pietarinen A, Tulenheimo T (eds) (2009) Games: unifying logic, language, and philosophy. Logic, epistemology, and the unity of science, vol 15. Springer, Heidelberg
325. Margolis E, Laurence S (eds) (2015) The conceptual mind. New directions in the study of concepts. The MIT Press, Cambridge, MA
326. Markov AA (1954) Theory of algorithms, Trudy Mat. Istituta imeni V.A. Steklova, vol 42. Izdatelstvo Akademii Nauk SSSR, Moscow
327. Marr Vision D (2010) A computational investigation into the human representation and processing of visual information. The MIT Press, Cambridge, MA
328. Martin WM (ed) (2006) Theories of judgment. Psychology, logic, phenomenology. Cambridge University Press, Cambridge, UK
329. Martin-Löf P (1984) Intuitionistic type theory. Bibliopolis, Napoli
330. Martin-Löf P (2006) 100 years of Zermelo's axiom of choice: what was the problem with it ? Comput J 49:345–350
331. Mazur S (1963) Computable analysis, Rozprawy Matematyczne, vol XXXIII, Państwowe Wydawnictwo Naukowe (PWN)
332. Mei S, Zarrabi N, Lees M, Sloot P (2015) Complex agent networks: An emerging approach for modeling complex systems. J Appl Soft Comput 37 Issue C:311–321. http://www.sciencedirect.com/science/article/pii/S1568494615005049
333. Mendel JM, Wu D (2010) Perceptual computing. Aiding people in making subjective judgements. IEEE press series on computational intelligence. Wiley, Hoboken, NJ
334. Menezes R, Evsukoff A, González MC (eds) (2013) Complex networks. Springer, Heidelberg
335. Meyn S (2008) Control techniques for complex networks. Cambridge University Press, New York, NY
336. Mieritz L (2012) Gartner survey shows why projects fail. http://thisiswhatgoodlookslike.com/2012/06/10/
337. Miikkulainen R, Bednar JA, Choe Y, Sirosh J (eds) (2005) Computational maps in the visual cortex. Science+Business Media Inc, New York, NY

338. Miller JH, Page SE (2007) Complex adaptive systems. An introduction to computational models of social life. Princeton University Press, Princeton, NJ
339. Minsky M (1974) A framework for representing knowledge, MIT, AI Laboratory Memo 306. https://dspace.mit.edu/handle/1721.1/6089
340. Minsky M (1985) Society of mind. Simon & Schuster, New York, NY
341. Minsky M (2006) Emotion machine: commonsense thinking, artificial intelligence, and the future of the mind. Simon & Schuster, New York, NY
342. Minsky M, Singh P, Sloman A (2004) The St. Thomas common sense symposium: designing architectures for human-level intelligence. AI Mag 25:113–124
343. Mishra RK, MaaB D, Zwierlein E (eds) (1993) On self-organization. An interdisciplinary search for a unifying principle. Springer, Heidelberg
344. Mitchel TM (1999) Machine learning. McGraw-Hill series in computer science, Boston, MA
345. Mitchell M (2006) Complex systems: network thinking. Artif Intell:1194–1212
346. Mitchell M (ed) (2009) Complexity: a guided tour. Oxford University Press, Oxford, UK
347. Mitchell R (2009) Y2k: The good, the bad and the crazy. Computerworld, vol 28
348. MüllerSchloer C, Schmeck H, Ungerer T (eds) (2011) Organic computing a paradigm shift for complex systems. Springer, Basel
349. Montgomery K (2005) Clinical judgement and the practice of medicine. Oxford University Press
350. Mooney R (2008) Learning to connect language and perception. In: Fox D, Gomes CP (eds) Proceedings of the 23rd AAAI conference on artificial intelligence (AAAI), Chicago, IL, 13–17 July 2008. AAAI Press, pp 1598–1601 (Senior Member Paper)
351. Moore EH (1910) Introduction to a form of general analysis. In: The New Haven mathematical colloquium, vol 2. Yale University Press, New Haven, pp 1–150
352. Moore GH (1982) The Warsaw School, widening applications, models of set theory (1918–1940). In: Zermelo's axiom of choice its origins, development, and influence. Dover Publications, Mineola, NY, pp 196–292
353. Mostowski A (1957) On a generalization of quantifiers. Fundamenta Mathematicae 44:12–36
354. Namatame A, Kaizouji T, Aruka Y (eds) (2006) The complex networks of economic interactions. Springer, Heidelberg
355. NASA (2016) Nasa iv&v facility. http://www.nasa.gov/centers/ivv/home/index.html
356. Aeronautics National, Administration Space (2007) NASA systems engineering handbook. NASA Headquarters, Washington, D.C
357. Newman MEJ (2006) The structure and function of complex networks. SIAM Rev 45:167–256
358. Ngo CL, Nguyen HS (2004) Hierarchical document clustering based on tolerance rough set model. In: Boulicaut J-F, Esposito F, Giannotti F, Pedreschi D (eds) Proceedings of the 8th European conference on principles of data mining and knowledge discovery (PKDD 2004), Pisa, Italy, 20–24 Sept 2004. Lecture notes in computer science, vol 3202. Springer, Heidelberg, pp 515–517
359. Nguyen HS (2017) Approximate boolean reasoning approach to rough sets and data mining. In: Ślęzak et al [472], pp 12–22. (plenary talk)
360. Nguyen HS (2006) Approximate boolean reasoning: foundations and applications in data mining. Trans Rough Sets: J Subline 5:344–523 (LNCS 4100; Springer, Heidelberg)
361. Nguyen HS, Jankowski A, Skowron A, Stepaniuk J, Szczuka M (2010) Discovery of process models from data and domain knowledge: a rough-granular approach. In: Yao JT (ed) Novel developments in granular computing: applications for advanced human reasoning and soft computation, PA, IGI Global, Hershey, pp 16–47
362. Nguyen HS, Nguyen SH (1999) Rough sets and association rule generation. Fundamenta Informaticae 40(4):383–405
363. Nguyen SH, Bazan J, Skowron A, Nguyen HS (2004) Layered learning for concept synthesis. Trans Rough Sets: J Subline 1:187–208 (LNCS 3100; Springer, Heidelberg)
364. Niazi MA, Hussain A (2013) Cognitive agent-based computing-I. A unified framework for modeling complex adaptive systems using agent-based & complex network-based methods. Springer, Heidelberg

365. Nicolis J (1986) Dynamics of hierarchical systems an evolutionary approach. Springer, Berlin
366. Noë A (2004) Action in perception. MIT Press, Cambridge, MA
367. Nolfi S, Mirolli M (eds) (2010) Evolution of communication and language in embodied agents. Springer, Heidelberg
368. Northrop L (2011) Does scale really matter?: ultra-large-scale systems seven years after the study. Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA. http://2013.icse-conferences.org/documents/publicity/ICSE_Keynote-Northrop.pdf
369. Northrop L, Feiler P, Gabriel R, Goodenough J, Lingerand R, Longstaff T, Kazman R, Klein M, Schmidt D, Sullivan K, Wallnau K (2006) Ultra-large-scale systems the software challenge of the future software engineering. Institute Carnegie Mellon, Pittsburg, PA
370. Łoś J, Suszko R (1958) Remarks on sentential logics. In: Béziau [49], pp 177–186
371. Ogden CK, Richards IA (1989) Study of the influence of language upon thought and of the science of symbolism. Kegan Paul, Trench, Trubner and Co., Ltd, London, 1923. (see also this book: With an Introduction by Postgate JP and Supplementary Essays by Malinowski B, Crookshank FG. Harcourt Brace Jovanovich Inc, New York (1989))
372. Omicini A, Ricci A, Viroli M (2017) The multidisciplinary patterns of interaction from sciences to computer science. In: Goldin et al. [169], pp 395–414
373. Open Group (2015) TOGAF®9 foundation study guide, 3rd edn. https://www.opengroup.org/togaf/
374. Ore O (1944) Galois connexions. Trans Am Math Soc 4:493–513
375. Ortiz CL Jr (2016) Why we need a physically embodied Turing test and what it might look like. AI Mag 37:55–62
376. Osterwalder A, Pigneur Y (2010) Business model generation: a handbook for visionaries, game changers, and challengers. Wiley, Hoboken, NJ
377. Pahl N, Richter A (2009) Swot analysis: Idea, methodology and a practical approach. GRIN Verlag GmbH, Münich
378. Pal SK, Polkowski L, Skowron A (eds) (2004) Rough-neural computing: techniques for computing with words. Series cognitive technologies, Springer, Heidelberg
379. Park K, W. W. (eds) The internet as a large-scale complex system. Santa Fe Institute Studies on the Sciences of Complexity, Cambridge University Press, Cambridge, UK
380. Pastor-Satorras R, Rubi M, Diaz-Guilera A (eds) (2003) Statistical mechanics of complex networks. Springer, Heidelberg
381. Pawlak Z (1981) Classification of objects by means of attributes, Reports, vol 429. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
382. Pawlak Z (1981) Information systems—theoretical foundations. Inf Syst 6:205–218
383. Pawlak Z (1981) Rough relations, Reports, vol 435. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
384. Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11:341–356
385. Pawlak Z (1983) Systemy Informacyjne. Podstawy Teoretyczne (Information Systems. Theoretical Foundations). Wydawnictwa Naukowo-Techniczne, Warsaw, Poland (in Polish). http://bcpw.bg.pw.edu.pl/dlibra/docmetadata?id=2327
386. Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. System theory, knowledge engineering and problem solving, vol 9. Kluwer Academic Publishers, Dordrecht, The Netherlands
387. Pawlak Z (1997) Vagueness—a rough set view. In: Mycielski J, Rozenberg G, Salomaa A (eds) Structures in logic and computer science. Lecture notes in computer science, vol 1261. Springer, Heidelberg, pp 106–117
388. Pawlak Z (1998) An inquiry into anatomy of conflicts. J Inf Sci 109:65–78
389. Pawlak Z, Skowron A (2007) Rough sets and boolean reasoning. Inf Sci 177(1):41–73
390. Pawlak Z, Skowron A (2007) Rough sets: Some extensions. Inf Sci 177(1):28–40
391. Pawlak Z, Skowron A (2007) Rudiments of rough sets. Inf Sci 177(1):3–27
392. Pearl J (2009) Causal inference in statistics: an overview. Stat Surv 3:96–146
393. Pedrycz W (2013) Granular computing analysis and design of intelligent systems. CRC Press, Taylor & Francis, Boca Raton, FL

394. Pedrycz W, Skowron S, Kreinovich V (eds) (2008) Handbook of granular computing. Wiley, Hoboken, NJ
395. Peirce C (2011) New elements of mathematics by Charles S. Peirce (Book 4). De Gruyter, Berlin
396. Pessoa L (2013) The cognitive-emotional brain. From interactions to integration. The MIT Press, Cambridge, MA
397. Pfau T (2010) The letter of judgment: practical reason in Aristotle, the stoics, and Rousseau. The Eighteenth Century 51:289–316
398. Pietarinen A-V (2006) Signs of logic peircean themes on the philosophy of language, games, and communication. Springer, Heidelberg
399. Plato (1925) Plato in Twelve Volumes. Harvard University Press, London. http://www.perseus.tufts.edu/hopper/searchresults?q=PLato
400. Poggio T, Smale S (2003) The mathematics of learning: dealing with data. Notices AMS 50(5):537–544
401. Polkowski L (ed) Approximate reasoning by parts. An introduction to rough mereology. Intelligent systems reference library, vol 20. Springer, Heidelberg
402. Polkowski L, Skowron A (1996) Rough mereology: a new paradigm for approximate reasoning. Int J Approx Reason 15(4):333–365
403. Polkowski L, Skowron A (2001) Rough mereological calculi of granules: a rough set approach to computation. Comput Intell: Int J 17(3):472–492
404. Pollak B (ed) (2006) Ultra-large-scale systems. The software challenge of the future. Software Engineering Institute. Carnegie Mellon University, Pittsburgh, PA
405. Portugali J (2000) Self-organization and the city. Springer, Berlin
406. Prokopenko M (ed) (2014) Guided self-organization: inception. Springer, Heidelberg
407. Pylyshyn ZW (2007) Things and places: how the mind connects with the world. The MIT Press, Cambridge, MA
408. Rahwan I, Simari GR (eds) (2009) Argumentation in artificial intelligence. Springer, Heidelberg
409. Rasiowa H (1974) An algebraic approach to non-classical logics. Studies in logic and the foundations of mathematics. Elsevier Science, Amsterdam
410. Rasiowa H, Sikorski R (1963) The mathematics of metamathematics. Polish Scientific Publishers, Warsaw
411. Reiter R (1978) On closed world data bases. In: Gallaire H, Minkers J (eds) Logic and data bases. Plenum Press, New York, NY, pp 55–76
412. Reiter R (1980) A logic for default reasoning. Artif Intell 13:81–132
413. Richter UM (ed) (2009) Controlled self-organisation. Using learning classifier systems. KIT Scientific Publishing, Karlsruher Institut für Technologie (KIT), Karlsruhe
414. Rissanen J (1978) Modeling by shortes data description. Automatica 14:465–471
415. Rissanen J (1985) Minimum-description-length principle. In: Kotz S, Johnson N (eds) Encyclopedia of statistical sciences. Wiley, New York, NY, pp 523–527
416. Robinson EJ (1985) Tales and poems of South India. T. Woolmer, London
417. Ronzitti G (ed) (2011) Vagueness: a guide, logic, epistemology, and the unity of science series, vol 19. Springer, Heidelberg
418. Rothe J (ed) (2016) Economics and computation. Springer, Heidelberg
419. Roy A, Pal SK (2003) Fuzzy discretization of feature space for a rough set classifier. Pattern Recogn Lett 24(6):895–902
420. Rozenberg G, Bäck T, Kok J (eds) (2012) Handbook of natural computing. Springer, Hidelberg
421. Rusnock P, George R (eds) (2014) Bernard Bolzano: theory of science 4. Oxford University Press, Oxford
422. Russell SJ, Norvig P (2010) Artificial intelligence. A modern approach. Pearson Education Inc, Upper Saddle River, NJ
423. Russo N, Antonchenko VY, Kryachko E (eds) (2008) Self-organization of molecular systems. From molecules and clusters to nanotubes and proteins. Springer, Heidelberg

424. Sage A, Rouse W (2014) Handbook of systems engineering and management. Wiley-Interscience, Hoboken, NJ
425. Saha S, Murthy C, Pal S (2010) Application of rough ensemble classifier to web services categorization and focused crawling. Web Intell Agent Syst 8(2):181–202
426. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18:613–620
427. van der Schaar M (ed) (2013) Judgement and the epistemic foundation of logic. Springer, Heidelberg
428. Scott DS (1974) Completeness and axiomatizability in many-valued logic. In: Béziau [49], pp 281–299
429. Searle J (1984) Minds, brains and science (1984 Reith Lectures). Harvard University Press, Cambridge, MA
430. Segel LA, Cohen IR (eds) (2001) Design principles for the immune system and other distributed autonomous systems. Oxford University Press, New York, NY
431. Seising R, Trillas E, Moraga C, Termini S (eds) (2013) On fuzziness. A homage to Lotfi A. Zadeh, vols 1& 2. Studies in fuzziness and soft computing vols 298 & 299. Springer, Heidelberg, pp 298–299
432. Shen H-W (2013) Community structure of complex networks. Springer, Heidelberg
433. Shevchenko P (ed) (2011) Modelling operational risk using bayesian inference. Springer, Heidelberg
434. Shklarsh A, Ariel G, Schneidman E, Ben-Jacob E (2011) Smart swarms of bacteria-inspired agents with performance adaptable Interactions. PLoS Comput Biol 7:1–11. [e1002177, doi:10.1371/journal.pcbi.1002177]
435. Shoham Y, Brown KL (2009) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, New York, NY
436. Siegelmann HT (1995) Computation beyond the Turing limit. Science 268:545–548
437. Siegelmann HT (1999) Neural networks and analog computation. Beyond the Turing limit. Springer Science & Business Media, New York, NY
438. Siegelmann HT (2013) Review Turing on super-Turing and adaptivity. Progr Biophys Mol Biol 113:117–126
439. Silva TC, Zhao L (2016) Machine learning in complex networks. Springer, Heidelberg
440. Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106:467–482
441. Simovici D, Djeraba C (2014) Mathematical tools for data mining: set theory, partial orders, combinatorics. Springer, Heidelberg
442. Skowron A (1993) Boolean reasoning for decision rules generation. In: Komorowski J, Raś ZW (eds) Proceedings of ISMIS'1993, Trondheim, Norway, 15–18 June 1993. Lecture notes in artificial intelligence, vol 689. Springer, Heidelberg, pp 295–305
443. Skowron A (2005) Rough sets and vague concepts. Fundamenta Informaticae 64(1–4):417–431
444. Skowron A, Jankowski A (2016) Interactive computations: toward risk management in interactive intelligent systems. Nat Comput 15(3):465–476. doi:(10.1007/s11047-015-9486-5)
445. Skowron A, Jankowski A (2016) Toward W2T foundations: Interactive granular computing and adaptive judgement. In: Zhong N, Ma J, Liu J, Huang R, Tao X (eds) Wisdom Web of Things (W2T). Springer, Heidelberg, pp 47–71
446. Skowron A, Jankowski A, Dutta S (2015) Toward problem solving support based on big data and domain knowledge: interactive granular computing and adaptive judgement. In: Japkowicz N, Stefanowski J (eds) Big data analysis: new algorithms for a new society, Series big data, vol 16. Springer, Heidelberg, pp 44–90
447. Skowron A, Jankowski A, Swiniarski RW (2013) 30 years of rough sets and future perspectives. In: Ciucci D, Inuiguchi M, Yao Y, Ślęzak D, Wang G (eds) Proceedings of the 14th international conference on rough sets, fuzzy sets, data mining, and granular computing (RSFDGrC 2013), Halifax, Canada, 11–14 Oct 2013. Lecture notes in computer science, vol 8170. Springer, Heidelberg, pp 1–13

448. Skowron A, Jankowski A, Wasilewski P (2012) Interactive computational systems. In: Popowa-Zeugmann L (ed) Proceedings of the workshop on concurrency, specification, and programming (CS&P 2012), Humboldt University, Berlin, 26–28 Sept 2012, Humboldt-Universität, Berlin, Informatik-Bericht, vol 225, pp 358–369
449. Skowron A, Nguyen HS (1999) Boolean reasoning scheme with some applications in data mining. In: Proceedings of the 3rd European conference on principles and practice of knowledge discovery in databases. Lecture notes in computer science, vol 1704. Springer, Berlin, pp 107–115
450. Skowron A, Nguyen HS (2017) Rough sets: from rudiments to challenges. In: Skowron and Suraj [463], pp 75–173
451. Skowron A, Peters J (2003) Rough sets: trends and challenges. In: Proceedings of the 9th international conference on rough sets, fuzzy sets, data mining, and granular computing (RSFDGrC'2003), Chongqing, China, 26–29 May 2003. In: Guoyin Wang G, Liu Q, Yao YY and Skowron A (eds) Lecture notes in computer science, vol. 2639. Springer, Heidelberg pp 25–34 (plenary talk)
452. Skowron A, Peters J, Stepaniuk J, Jankowski A (2008) Optimization in discovery of compound granules. Fundamenta Informaticae 14(3):367–285
453. Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. In: Słowiński R (ed) Intelligent decision support - handbook of applications and advances of the rough sets theory. Series system theory, knowledge engineering and problem solving, vol. 11. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 331–362
454. Skowron A, Stepaniuk J (1994) Generalized approximation spaces. In: The Third International workshop on rough sets and soft computing proceedings (RSSC'1994), San Jose, California, USA, 10–12 Nov 1994, pp 156–163
455. Skowron A, Stepaniuk J (2017) Information granules and rough-neural computing. In: Pal et al [378], pp 43–84
456. Skowron A, Stepaniuk J (2005) Hierarchical modelling in searching for complex patterns: constrained sums of information systems. J Exp Theoret Artif Intell 17:83–102
457. Skowron A, Stepaniuk J (2010) Approximation spaces in rough-granular computing. Fundamenta Informaticae 100 (1–4) 141–157
458. Skowron A, Stepaniuk J (2011) Data driven approximate reasoning about changes. In: Szczuka M, Czaja L, Skowron A, Kacprzak M (eds) Concurrency, specification and programming (CS&P'2011), Pultusk, 28–30 Sept 2011. Warsaw, Białystok University of Technology, Humboldt University, Warsaw University, pp 477–486
459. Skowron A, Stepaniuk J, Jankowski A, Bazan J (2012) Rough derivatives as dynamic granules in rough granular calculus. In: Greco S, Bouchon-Meunier B, Coletti G, Fedrizzi M, Matarazzo B, Yager RR (eds) Advances on computational intelligence - proceedings of the 14th international conference on information processing and management of uncertainty in knowledge-based systems (IPMU 2012), Catania, Italy, 9–13 July 2012. Part I. Communications in computer and information science, vol 297. Springer, Heidelberg, pp 321–330
460. Skowron A, Stepaniuk J, Jankowski A, Bazan JG, Swiniarski R (2012) Rough set based reasoning about changes. Fundamenta Informaticae 119(3–4):421–437
461. Skowron A, Suraj Z (1993) A rough set approach to real-time state identification. Bull EATCS 50:264–275
462. Skowron A, Suraj Z (1995) Discovery of concurrent data models from experimental tables: a rough set approach. In: Proceedings of the first international conference on knowledge discovery and data mining (KDD 1995), Montreal, Canada, 20–21 Aug 1995. AAAI Press, Menlo Park, CA, 1995, pp 288–293
463. Skowron A, Suraj Z (eds) Rough sets and intelligent systems. Professor Zdzislaw Pawlak in Memoriam, Series Intelligent systems reference library, vol 42–43. Springer, Heidelberg
464. Skowron A, Szczuka M (2009) Toward interactive computations: a rough-granular approach. In: Koronacki J, Raś Z, Wierzchoń S, Kacprzyk J (eds) Advances in machine learning II: dedicated to the memory of professor Ryszard S. Michalski. Studies in computational intelligence, vol 263. Springer, Heidelberg, pp 23–42

465. Skowron A, Wasilewski P (2010) Information systems in modeling interactive computations on granules. In: Szczuka M, Kryszkiewicz M, Ramanna S, Jensen R, Hu Q (eds) Proceedings of RSCTC 2010. Lecture notes in computer science, vol 6086. Springer, Heidelberg, pp 730–739
466. Skowron A, Wasilewski P (2010) An introduction to perception based computing. In: Kim T-H, Lee Y-H, Kang BH, Ślęzak D (eds) Proceedings of FGIT 2010. Lecture notes in computer science, vol 6485. Springer, Heidelberg, pp 12–25
467. Skowron A, Wasilewski P (2011) Information systems in modeling interactive computations on granules. Theoret Comput Sci 412(42):5939–5959
468. Skowron A, Wasilewski P (2011) Toward interactive rough-granular computing. Control Cybern 40(2):1–23
469. Skowron A, Wasilewski P (2012) Interactive information systems: toward perception based computing. Theoret Comput Sci 454:240–260
470. Ślęzak D, Wróblewski J (1999) Classification algorithms based on linear combinations of features. In: Zytkow J, Rauch J (eds) Third European conference on principles and practice of knowledge discovery in databases (PKDD'1999). Lecture notes in artificial intelligence, vol 1704. Springer, Heidelberg, pp 548–553
471. Ślęzak D, Wasilewski P (2008) Foundations of rough sets from vagueness perspective. In: Hassanien AE, Suraj Z, Ślęzak D, Lingras P (eds) Rough computing: theories. PA, Technologies and Applications, IGI Global, Hershey, pp 1–37
472. Ślęzak D, Yao JT, Peters JF, Ziarko W, Hu X (eds) Proceedings of the 10th international conference on rough sets, fuzzy sets, data mining, and granular computing (RSFDGrC'2005), Regina, Canada, August 31–September 3, 2005, Part II. Lecture notes in artificial intelligence, vol 3642. Springer, Heidelberg
473. Sliger M, Broderick S (2008) The software project manager's bridge to agility. The Agile software development series. Addison-Wesley Professional, Upper Saddle River, NJ
474. Slomson B, Bell J (2006) Models and ultraproducts: an introduction. Dover Publications, Mineola, NY
475. Smith JA (2007) On the soul by Aristotle (trans). The University of Adelaide, Adelaide. http://ebooks.adelaide.edu.au/a/aristotle/
476. Sobek DK II, Smalley A (2008) Understanding A3 thinking: a critical component of Toyota's PDCA management system. Productivity Press, Boca Raton, London, New York
477. Sowa J (2013) The goal of language understanding. VivoMindIntelligence Inc, Rockville, MD. http://www.jfsowa.com/talks/goal.pdf
478. Sporns O, Chialvo DR, Kaiser M, Hilgetag CC (2004) Organization, development and function of complex brain networks. Trends Cognit Sci 8:418–425
479. Stacey RD, Griffin D, Shaw P (2000) Complexity and management. Fad or radical challenge to systems thinking? Complexity and emergence in organizations, Rutledge. Taylor & Francis, London
480. Staddon JER (2001) Adaptive dynamics. The theoretical analysis of behavior. Columbia University Press, New York, NY
481. Steels L (2015) The talking heads experiment: origins of words and meanings (Computational Models of Language Evolution). Language Science Press, Berlin
482. Steels L, Hild M (eds) (2012) Language grounding in robots. Springer, Heidelberg
483. Stefanoiu D, Borne P, Popescu D, Filip FG, Kamel A (2014) Optimization in engineering sciences: approximate and metaheuristic methods. Wiley & Sons, Hoboken, NJ
484. Sternberg R (2003) Wisdom, intelligence, and creativity synthesized. Cambridge University Press, Cambridge, UK
485. Strogatz SH (2001) Exploring complex networks. Nature 410:268–276
486. Summerfield M (2013) Python in practice: create better programs using concurrency, libraries, and patterns (Developer's Library). Addison-Wesley Professional, Upper Saddle River, NJ
487. Sun R (2001) Duality of the mind: a bottom-up approach toward cognition. Lawrence Erlbaum Associate, Mahwah, NJ

488. Sun R (ed) (2006) Cognition and multi-agent interaction. From cognitive modeling to social simulation. Cambridge University Press, Cambridge, UK
489. Sutherland J, Schwabe K (2012) 30 Days: how agile managers beat the odds, delight their customers, and leave competitors in the dust. Wiley, New York, NY
490. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. The MIT Press, Cambridge, MA
491. Sycara K (1998, Summer) Multiagent systems. AI Mag:79–92
492. SYMPOSIUM-ON-FOUNDATIONS-OF-MATHEMATICS. Warsaw, 1959. Infinitistic methods. In: Proceedings of the symposium on foundations of mathematics, Warsaw, 2–9 Sept 1959
493. Székely GJ (2011) Paradoxes in probability theory and mathematical statistics, vol 15. Springer, Berlin
494. Tabues GA (2014) The rise and fall of thinking machine. Inc., Magazine. www.inc.com/magazine/19950915/2622.html. Accessed 26 June 2014
495. Tamir DE, Rishe ND, Kandel A (eds) (2015) Fifty years of fuzzy logic and its applications. Studies in fuzziness and soft computing, vol 326. Springer, Heidelberg
496. Tarski A (1928) Remarks on fundamental concepts of the methodology of mathematics. In: Béziau [49], pp 67–70
497. Tarski A (1933) Pojecie prawdy w jezykach nauk dedukcyjnych (The concept of truth in deductive science). Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizycznych 34:1–114. (in Polish); English Transl. in [163]
498. Tarski A (1938) Der Aussagenkalkul und die Topologie. Fundamenta Mathematicae 31:103–134
499. Tarski A (1983) Logic, semantics, metamathematics (trans: Woodger JH). Oxford University Press, Oxford, UK
500. Techopedia (2016) Independent verification and validation (iv&v). https://www.techopedia.com/definition/24836/
501. Telegársky R (1987) Topological games: on the 50th anniversary of the Banach-Mazur game. Rocky Mount J Math 17:227–276
502. Terzis G, Arp R (eds) (2013) Information and living systems. Philosophical and scientific perspectives. The MIT Press, Cambridge, MA
503. Thai MT, Pardalos PM (eds) (2012) Handbook of optimization in complex networks. Springer, Heidelberg
504. Thiele LP (2010) The heart of judgment: practical wisdom, neuroscience, and narrative. Cambridge University Press, Cambridge, UK
505. Toffoli T (1982) Physics and computation. lnt J Theoret Phys 21:165–175
506. Tomforde S, Müller-Schloer C (2014) Incremental design of adaptive systems. J Ambient Intell Smart Environ 6(2):179–198
507. Troelstra AS, van Dalen D (1988) Constructivism in mathematics. An introduction, vols 1 and 2. Elsevier, Amsterdam
508. Turing A (1936) On computable numbers, with an application to the Entscheidungsproblem. Proc Lond Math Soc 42:230–265
509. Turing A (1939) Systems of logic based on ordinals. PhD thesis, Princeton University, Princeton, NJ
510. Turing A (1950) Computing machinery and intelligence. Mind LIX:433–460
511. Łukasiewicz J (1920) O logice tójwartościowej (on three-valued logic). Ruch Filozoficzny 5:170–171
512. Łukasiewicz J (1952) On the intuitionistic theory of deduction. Koninklijke Nederlandse Akademie van Wetenschappen 55:202–212
513. Łukasiewicz J (1970) Selected works (Studies in logic and the foundations of mathematics). North-Holland, Amsterdam
514. Ulam S, Bednarek A (1990) On the theory of relational structures and schemata for parallel computation. In: Analogies between analogies: the mathematical reports of S.M. Ulam and his Los Alamos Collaborators, pp 477–508

515. Ulam S, Bednarek A, Ulam F (1990) Analogies between analogies: the mathematical reports of S.M. Ulam and His Los Alamos Collaborators. University of California Press, Oakland, CA
516. Ulrich W (2002) Legacy systems: transformation strategies. Prentice Hall, Upper Saddle River, NJ
517. Ulrich W, McWhorte N (2010) Business architecture: the art and practice of business transformation. Wiley, Hoboken, NJ
518. Ulrich W, Newcomb P (2010) Information systems transformation: architecture-driven modernization case studies. Morgan Kaufmann, New York, NY
519. Urbanowicz R, Moore JH (2009) Learning classifier systems: a complete introduction, review, and roadmap. J Artif Evol Appl 2009:1–25
520. Väänänen J (1969) Lindström Theorem. In: éziau B [49], pp 231–236
521. Valiant L (2013) Probably approximately correct. nature's algorithms for learning and prospering in a complex world. Basic Books, a member of the Perseus books group, New York, NY
522. Valiant L (2017) Research interests. www.people.seas.harvard.edu/valiant/researchinterests.htm
523. van Atten M, Boldini P, Bourdeau M, Heinzmann G (eds) (2008) One hundred years of intuitionism (1907–2007) The Cerisy Conference. Birkhäuser, Providence, RI
524. van Dalen D (ed) (1981) Brouwer's Cambridge Lectures on Intuitionism (1951). Cambridge University Press, Cambridge, UK
525. van Dalen D (2017) Another look at Brouwer's dissertation. In: van Atten et al [523], pp 3–21
526. van der Aalst WMP (ed) (2011) Process mining discovery. Conformance and enhancement of business processes. Springer, Heidelberg
527. Vapnik V (1998) Statistical learning theory. Wiley, New York, NY
528. Wagon S (1993) The Banach-Tarski Paradox. Cambridge University Press, Cambridge, UK
529. Walton DN, Krabbe ECW (eds) (1995) Commitment in dialogue: basic concepts of interpersonal reasoning. State University of New York Press, Albany
530. Wasson C (2005) System analysis, design, and development: concepts, principles, and practices. Wiley-Interscience, Hoboken, NJ
531. Wasson C (2009) Software engineering best practices: lessons from successful projects in the top companies. McGraw-Hill Osborne Media, New York, NY
532. Weiss G (ed) (2013) Multiagent systems. MIT Press, Cambridge, MA
533. Wikipedia (2015) Sensitivity analysis. https://en.wikipedia.org/wiki/Sensitivity_analysis
534. Wikipedia (2015). What if chart. https://en.wikipedia.org/wiki/What_if_chart
535. Wikipedia (2015). Zachman framework. http://en.wikipedia.org/wiki/Zachman_Framework
536. WikiPedia (2016) Independent software verification and validation. https://en.wikipedia.org/wiki/
537. Wilson RA, Keil F (eds) (1999) The MIT encyclopedia of the cognitive sciences. The MIT Press, Cambridge, MA
538. WITAS (2017) Project witas homepage. http://www.ida.liu.se/ext/witas/eng.html
539. Wittgenstein L (1967) Philosophical Investigations. (trans: Anscombe GEM (1953)), 3rd edn. Blackwell, Oxford, UK
540. Wittgenstein L (1983) Remarks on the foundations of mathematics. The MIT Press, Cambridge, MA
541. Wojna A (2005) Analogy based reasoning in classifier construction. In: Trans Rough Sets IV: J Subline, Springer, Heidelberg, pp 277–374
542. Wolski M (2017) Science and semantics: a note on vagueness. In: Skowron and Suraj [463], pp 623–643
543. Yang A, Shan Y (2008) Intelligent complex adaptive systems. IGI Global, Herehey, PA
544. Yang X (2010) Nature-inspired metaheuristic algorithms. Luniver Press, Frome, UK
545. Yourdon E (2003) Death March, 2nd edn. Prentice Hall, Upper Saddle River, NJ
546. Zadeh L (1997) Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst 90:111–127

547. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
548. Zadeh LA (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans Syst Man Cybern SMC-3:28–44
549. Zadeh LA (1979) Fuzzy sets and information granularity. In: Advances in fuzzy set theory and applications. North-Holland, Amsterdam, pp 3–18
550. Zadeh LA (1999) From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. IEEE Trans Circuits Syst 45:105–119
551. Zadeh LA (2001) A new direction in AI: toward a computational theory of perceptions. AI Mag 22(1):73–84
552. Zadeh LA (2017) Foreword. In: Pal et al [378], pp IX-XI
553. Zadeh LA (2006) Generalized theory of uncertainty (GTU)-principal concepts and ideas. Comput Stat Data Anal 51:15–46
554. Zadeh LA (ed) (2012) Computing with Words: principal concepts and ideas. Studies in fuzziness and soft computing, vol 277. Springer, Heidelberg
555. Zhang W (ed) (2008) Self-organization. Theories and methods. Nova Science Publishers, New York, NY
556. Zhong N, Ma J, Huang R, Liu J, Yao Y, Chen YZJ (2013) Research challenges and perspectives on Wisdom Web of Things (W2T). J Supercomput 64:862–882
557. Zhong N, Ma J, Liu J, Huang R, Tao X (eds) (2016) Wisdom web of things (W2T). Springer, Heidelberg
558. Zuse K (1969) Rechnender Raum. Friedrich Vieweg & Sohn, Braunschweig (English transl: Calculating Space, MIT Technical Translation AZT-70-164-GEMIT, Cambridge, MA, 1970)

Index

A. Jankowski, *Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective*, Lecture Notes in Networks and Systems 17, DOI 10.1007/978-3-319-57627-5