Modeling Knowledge: Model-Based Decision Support and Soft Computations*

Marek Makowski[†] Andrzej P. Wierzbicki[‡]

Draft of Jun-05-2003, 09:21

Abstract

This chapter provides an overview of model-based support for modern decision making. It starts with discussing basic elements of decision making process, including characteristics of complex decision problems, concepts of rationality, and various requirements for model-based support at different stages of decision making process. Then the characteristics of models, and of modeling processes aimed at decision-making support for complex problems are presented. In this part guidelines for model specification and instantiation are illustrated by an actual complex model. This is followed by a discussion of modern methods of model analysis, which include a more detailed discussion of reference point optimization methods, and an outline of methods for sensitivity analysis, and of softly constrained inverse simulation. Finally, an overview of architecture of model-based decision support system is presented.

Keywords: decision making, decision support systems, fuzzy sets, goal programming, intuition, knowledge, mathematical modeling, model analysis, model specification, multi-objective optimization, Pareto-optimality, robustness, sensitivity analysis, simulation, soft computing.

Contents

*This is a draft version of the chapter in: *Applied Decision Support with Soft Computing*, X. Yu and J. Kacprzyk, Eds., pages 3–60, Springer-Verlag, Berlin, 2003, ISBN 3-540-02491-3.

[†]International Institute for Applied Systems Analysis, Schlossplatz 1, A-2361 Laxenburg, Austria, e-mail: marek@iiasa.ac.at, http://www.iiasa.ac.at/~marek

[‡]National Institute of Telecommunication, Szachowa 1, 04-894 Warsaw, Poland, e-mail: a.wierzbicki@itl.waw.pl, http://www.itl.waw.pl

1 Introduction

Rational decision making is becoming increasingly difficult despite fast development of methodology for decision support and even faster development of computing hardware, networks and software. At least three commonly known observations support this statement:

- First, complexity of problems for which decisions are made grows even faster.
- Second, the amount of knowledge potentially applicable to rational decision making is also growing very fast. However, this knowledge is available in diversified forms (encoded in analytical models, stored in large and heterogeneous data sets, or as tacit knowledge of practitioners and scientists).
- Third, knowledge and experiences related to rational decision making develop rapidly but in diverse directions, therefore integration of various methodologies and tools is practically impossible.

Most of decision problems are no longer well structured problems that are easy to be solved by intuition or experience supported by relatively simple calculations. Even such types of problems that used to be easy to be defined and solved, are now much more complex because of the globalization of the economy, and a much greater awareness of its linkages with various environmental, social and political issues. Therefore, modern Decision Makers (DM) typically need to integrate quickly and reliably knowledge from these various areas of science and practice in order to use it as one element of the Decision Making Process (DMP). Unfortunately, the culture, language and tools developed for knowledge representations in key application areas (such as economy, engineering, finance, environment management, social and political sciences) are very diversified. This observation is known to everybody who has ever participated in a teamwork with researchers and practitioners having background in different areas. Given the great diversity of knowledge representations in various disciplines, and the fast growing amount of knowledge in most areas, the need for efficient knowledge integration for decision support becomes a challenge that deserves to be addressed.

Good DMs make decisions by combining own knowledge, experience, and intuition with that available from other sources. Experienced DMs have a fairly stable set of ways for thinking, evaluating, judging and making decisions; this is often called a Habitual Domain (HD), defined by Yu (1990). HD is specific for each person, and it differs often substantially, even for persons with similar cultural and professional background, experience and duties. Moreover, whenever a complex problem is considered, then no one individual system of management (represented by one person) is sufficient. Therefore, typically a DMP involves more than one person, its participants often have very different HDs, and it is the integration of their ideas that creates a powerful force for change and improvement, which typically drive DMPs for complex problems.

For computerized decision support, the key question is how to represent knowledge about the decision situation – including external, objective knowledge about the underlying physical, biological, technical, environmental etc. aspects, and the subjective, individual knowledge and preferences of decision makers. The experiences in applications of decision support have shown that these two types of knowledge should be expressed separately, and then used in a combined way.

There are various approaches even to the question how to express the external knowledge. One approach is to deny the existence of any external knowledge except that contained in relevant data sets and in logic; this perspective of knowledge engineering leads to so-called *data-based decision support*. Another approach is to accept relevant knowledge coming from selected disciplines of science and express it in the form of computerized mathematical models; this perspective leads to so-called *model-based decision support*. The latter developed historically from the tradition of various disciplines, including engineering design and operations research (OR).

The OR methods were developed and first applied to well structured decision problems, where optimization focused approaches provided solutions corresponding to optimization of a precisely defined goal in a precisely formulated mathematical programming problem that can adequately represent only a well structured problem. Due to the successes of early applications and limitations of computing power, OR for relatively long time was focused on single-criterion optimization approaches, which has limited application to actual decision-making support for complex problems. However, the assumption made for application of the classical OR methods are not adequate for most of actual support for decision-making related to complex problems, because of at least the following reasons:

- Actual complex problems can hardly be represented by only one precisely defined mathematical model.
- Already about 50 years ago Simon (1958) has pointed out that actual decisions, particularly in large organizations, are optimized only until a satisfying (in technical terms called *satisficing*) level of key criteria is reached; at this point instead of a further optimization other attributes of decisions are considered.
- DMs clearly prefer to be sure that their sovereignty in making decisions (for which he is responsible) is not jeopardized by developers of the model used for decision support, nor by a computer used for its analysis.

The limitations of the traditional OR approaches have been discussed e.g. by Ackoff (1967), Ackoff (1979), and Chapman (1988). Motivated by the needs for model-based decision-making support for complex problems, various scientific communities have over last several decades developed diversified methodologies and tools (often referred to as *modeling paradigms*) aimed at providing adequate model-based support to DMPs for such problems.

A scientific paradigm, as defined by Kuhn (1970) and by Hloyningen-Huene (1993) embodies the consensus of a scientific community on an approach to a problem (or to a class of problems), and consists of the theories, laws, rules, models, concepts, and definitions that are generally accepted in science and practice. Due to the unquestionable success of modeling in problem solving, various modeling paradigms have been intensively developing over last few decades. The following publications provide a small sample of diverse successful attempts to develop modeling methods and apply to various Decision Support Systems (DSS) which over the years have resulted in recognized modeling paradigms: (Zadeh, 1965; Charnes and Cooper, 1967; Keeney and Raiffa, 1976; Wierzbicki, 1977; Zimmermann, 1978; Wierzbicki, 1982; Maclean, 1985; Sawaragi, Nakayama and Tanino, 1985; Tversky and Kahneman, 1985; Yu, 1985; Steuer, 1986; Rapoport, 1989; Stewart, 1992; Wierzbicki, 1992c; Sakawa, 1993; Radermacher, 1994; Bosc and Kacprzyk, 1995; Raiffa, 1997; Zadeh and Kacprzyk, 1999; Wierzbicki, Makowski and Wessels, 2000; Wierzbicki and Makowski, 2000; Makowski, 2001; Ruan, Kacprzyk and Fedrizzi, 2001; Carlsson and Fullér, 2002; Fink, 2002; Liu, 2002; Makowski and Wierzbicki, 2003).

All these developments have been rational from the point of view of providing more efficient solutions for specific types of models or elements of modeling process. However, efficiency requirements in developing specific methodologies and tools for each modeling paradigm resulted in paradigm-specific problem representations, and corresponding modeling tools for model specification, analysis, documentation, and maintenance. As a consequence of this long-term development, it has become increasingly difficult to apply all pertinent paradigms to a problem at hand, because these resources are fragmented and using more than one paradigm for a problem at hand is expensive and time consuming in practice. Models have become complex and/or large, therefore the development and use of a model with even one specific paradigm is a costly and time-consuming process. Moreover, incompatible model representations used by different paradigms imply that resources used for modeling with one paradigm can hardly be reused when another paradigm is applied to the same problem. Therefore multi-paradigm modeling support is currently one of the most challenging problems faced not only by the OR community, but by a broader audience of computerized decision and design support or even generally scientific community. Various modeling paradigms could be used much more broadly, if scientific communities will get a better understanding of methodologies developed by other communities, and if modeling resources (model specifications, data, modeling tools) will be available in a uniform representation.

This chapter gives an overview of model-based decision making support methods, which are applicable not only to soft computing, but also to other modeling paradigms. Therefore it provides a context in which advanced soft computing methods can be used within multi-paradigm modeling, which in turn will make them available to a much larger scope of applications than that presented in this book.

The chapter is organized as follows. Section 2 discusses the issues of modern decision making, including the characteristics of complex problems requiring decision support, concepts of rationality, and the role of intuition in decision making, as well as the dynamics of decision processes. Section 3 presents an overview of key modeling paradigms relevant to the development of models aimed at decision-making support. Section 4 concentrates on basic concepts related to the architecture and implementation of model-based decision support systems.

2 Modern Decision Making

By a *decision maker* we could understand anybody who makes decisions or takes a part in a decision making process – but everyone makes daily many decisions. Thus, we limit the concept of a decision maker to those persons who are aware of importance of decisions made by them or at least reflect how these decisions are made. Nevertheless, this concept is very rich in meaning. We can distinguish many types of decision makers, such as a *top-level*, economic or political *decision maker*, or an *analyst* preparing the analysis of a given decision situation, or generally a *user* of a decision support system. We can also distinguish various types of decisions: operational, tactical, strategic, creative, *etc.*

But what is a *modern decision maker?* Is a person modern because of using computerized help, such as decision support systems or computer network communications? Is a person modern because of understanding computerized modeling and using its various means - logical, graph-theoretical, operational research type? Certainly, these are some dimensions of being modern. However, there is also a richer meaning of being modern: *a person is modern because of using basic concepts that are adequate for understanding the contemporary, complex world*. This meaning is actually related to creativity and thus adaptability – two features that are especially important now, because the contemporary world is not only complex, but also changing fast.

2.1 Changing Civilization Eras

We live in a period of fundamental technological and social change. The reasons of this change are variously described, but one of the most widely accepted is the hypothesis of **the turn of cultural eras** – from the era of *industrial civilization* to the *information civilization era.* The information civilization society is related to many more specific terms, such as *service society, information society, networked society, knowledge based economy,* etc. These concepts are variously though similarly defined; for example, information society and knowledge based economy can be jointly defined as society and economy in which information and knowledge become essential productive factors. Other definitions put more stress on technological developments, such as the emergence of the *global information infrastructure.* However, we maintain that the most important is the change of basic concepts.

A cultural era is indeed defined by a system of basic concepts which are used for perceiving and understanding the world. Such a system must be widely accepted, embedded in the culture and education of the era; it constitutes a *cultural platform* of the era. The basic concepts that constitute the cultural platform of industrial civilization were developed much earlier than the beginnings of this era, usually dated at the 18-th century and coinciding with the Enlightenment era. These concepts were actually based on the discoveries of Gutenberg, Copernicus, Galileo, Newton, Kepler and others that helped to perceive the universe as a giant machine, to construct smaller machines and industrialize the world. Thus, the cultural platform of industrial civilization is *mechanistic* and *reductionist*: the analysis of the universe or a society is reduced to the analysis of their parts and relations between these parts. Most of our thinking about the world is still based on this platform.

The cultural platform of informational civilization is being formed now, though it uses concepts developed during over hundred years. It started with the theories of Maxwell and Einstein, challenging the mechanistic perception of the universe, proposing fundamentally different, relativistic interpretations. Later, many developments of telecommunication and control science contributed to the understanding of the role of information in complex systems, including the fundamental concept of feedback which changed the classical understanding of the relation of cause and effect. Finally, the development of computers and computer science resulted in diverse understanding of information processing.

The new cultural platform is not fully formed yet; some essential concepts – such as the deterministic theory of chaos – are fully known only to a handful of specialists. The full formation of the new cultural platform, its acceptance and spread by our educational systems might need yet several generations. There is no doubt today, however, that the mechanistic vision of the world of industrial civilization era will be replaced by a new vision. Some computer scientists propose to perceive the world as a giant computer; however, this is a simplistic vision which does not take into account relativism, feedback, chaos. More probable is that the new vision will have a systemic character and will include a deeper understanding of chaos.

Thus we cannot say that the information civilization has already arrived, though it has started. The date of its beginning might be counted at the discovery of the TCP-IP protocol in computer networks just before 1980; but full information civilization has not arrived with this date, similarly as industrial civilization did not arrive around 1760 with the improvement of steam engine, it only started then. There exist today information technologies that will fundamentally change the society even as we know it today. These are, for example, microprocessors with computing power comparable to that of supercomputers a few years ago, which will become cheap and generally used in the next few years; teleconferencing programs that make it possible to construct sophisticated videophones, available in every home; the explosive development of computer networking; computer integrated design, manufacturing and management; widespread use of computer intelligence, called today the *vision of ambient intelligence,* implying also widespread use of computerized decision support systems; and many others. But in order to use such systems effectively, we should be aware of basic concepts concerning decision making, used when developing such systems.

2.2 Concepts and Creative Processes

Language, logic, mathematics are just instruments of description, discussion, formalization of our deeper thoughts that might occur on a nonverbal level, in terms of images and basic concepts. It is obviously better for communication that basic concepts are well understood and have a well-defined meaning.

On the other hand, *a concept is the more valuable the richer it is*, in a synergetic sense; and the number and value of associated images related to a concept depend on the level of expertise. We shall discuss here mainly such concepts that are essential for modern computerized decision support.

For modeling external knowledge about decision situations, especially important is the concept of *cause and effect*; but it is understood still by most people, even philosophers, somewhat mechanistic and in terms of *binary logic*, while we know today that there are situations in which *fuzzy, rough or other multi-valued logic* is more adequate, and that this concept is even more fundamentally changed if we include also *feedback* mechanisms. The concept of feedback is essential for a modern, systemic understanding of the world – but it is differently perceived by an economist who has often been taught only the basic definition of this concept, differently by a control theory specialist who spent a life investigating various feedback effects, and differently by a specialist in the *theory of chaos*, who would say *feedback with nonlinear dynamics might produce chaos in a deterministic system, which will change the way we should think about the indeterminism of the universe.*

Indeterminism is deeply related with the concept of *uncertainty,* essential for decision theory. If we could be sure about consequences of every planned action, decision making would be much simpler. In fact, decision theory developed together with statistics, with probabilistic descriptions of uncertain results of decisions. However, today we know that probabilistic description of uncertainty is only one of such possible representations and is not always acceptable. It is a very good tool of describing uncertainty if we have access to large amounts of data from which probability distributions can be empirically derived or at least the parameters of some assumed (e.g. normal) distributions experimentally tested. If the data are more scarce, other ways of expressing uncertainty might be more adequate. One is suggested by the *rough set theory* of Pawlak (1991): classify existing data into three logical categories – *true*, *false,* and *may be,* then assess how large is the last, uncertain category. Another, actually older, is to use not trinary, but infinite-valued logic as suggested by *fuzzy set theory* of Zadeh (1965); this is particularly appropriate if the uncertainty is expressed by people, by linguistic descriptions instead of numerical data.

Another basic concept is that of *complexity.* It has a rather precise meaning in computer science. It gives then an approximation of the number of basic computations needed to solve a finite computational problem, in the worst possible case and in the sense of the character of the dependence of this number on the dimensions (number of variables) of the model of the problem. For very simple problems (which are rather rare) this dependence is linear; for simple problems, polynomial. Truly complex problems have exponential or combinatorial dependence of computations on the dimension of the model. This means that while we can solve, say in several hours of computing time, a problem of dimension n, solving the same type of problem with dimension $n+1$ can take several years. Complexity theory is actually much more complex than shortly described here: sometimes we cannot estimate the worst case complexity, sometimes we can only prove that the complexity is no less than some estimation, etc. However, these are technical details; the essential issue is that most truly difficult computational problems have at least non-polynomial (exponential or more) complexity. And this concerns only finite problems i.e. problems solvable in finite time. There are many problems – e.g. nonlinear programming problems – that can be solved only approximately, their exact solution would require infinite time. But again, complexity of solving them with prescribed accuracy is very often non-polynomial.

This describes only the issue of computational complexity. The issue of complexity of real life problems would have to take into account the mapping of the real life into computerized models. We shall show below that there is no precise way to evaluate such mapping; hence complexity of real life problems must have only an approximative, heuristic meaning. It must take into account that computerized models are very often composed from many submodels; that each of the submodels might have parameters determined with statistical procedures using available data; that a composition of submodels does not behave as a simple sum of them; that computational errors in complex models not only accumulate, but also might explode; that feedbacks between submodels not only increase computational complexity, but also might result in a chaotic behavior of the entire model; etc. To deal with all these problems, we must follow Einstein advice: *keep things as simple as possible, but not too simple.*

We mentioned that computerized models are used in decision support to represent external knowledge about a decision situation. We also distinguished knowledge from information. But what is *knowledge?* Typical definitions distinguish between at least two types of knowledge. One is *personal knowledge* – what a person actually knows and can apply usefully – which might be learned by education or derived from experience; the latter is often difficult to be described and communicated by words and is called in such case *tacit knowledge.* Another is *social knowledge* – entire body of knowledge established by various sciences including hard sciences, but also social sciences and humanities, recorded in books and by other means, transmitted by educational institutions. Selected part of social knowledge, pertinent for a given decision situation, we describe here as external knowledge – in opposition to the personal and tacit knowledge of the decision maker. But there are also various ways of representing knowledge in a computer. Classical definitions of knowledge engineering in computer science define knowledge as patterns that can be learned from data. However, this definition assumes in a sense that the only source of knowledge are pertinent data, and such an assumption is certainly too narrow for applications in decision support. Much better is to define knowledge as any model, either deduced from data or from social knowledge; patterns are only a type of logical models and we use very often other, analytical types of models to represent computerized knowledge.

We see that a precise agreement on a single meaning of a concept is often not possible. Moreover, such an agreement would not necessarily guard us against hidden assumptions in its accepted definition. The philosophy of science (see *e.g.* (Kuhn, 1964; Vollmer, 1984)) stresses that our *mesocosmic experience* results in the danger of such hidden assumptions; special techniques are used – *e.g.* that of thought experiments – to clarify such presuppositions. While the clarity and a unique meaning of a concept are needed for communication, there might be other aspects of cognitive processes for which the *richness of meaning is more valuable*; one of such aspects is *creativity*.

Any scientist with some experience in creative processes – such that require not only mere search and logical or experimental testing, but also result in new cognitive insights – would acknowledge the basic role of the richness of concepts related to her/his field of study. We often perceive uneasiness about an aspect of a scientific problem; then we try to look at it from various angles, through diverse images related to our basic concepts; after some time, *a moment of cognitive enlightenment, a heureka effect* comes. The mechanism of this effect relates to subconsciousness and intuition.

Similar creative processes occur also in other fields – in art, politics, even managerial decisions. It obviously helps to study your subject thoroughly and to discuss it with others – we are forced then to at least partly formulate our thoughts, and even casual remarks of others might trigger associations; it also helps to relax after some time of study, forget the problem for a while – an intuitive answer might need such a gestation period. It might also be helpful to extend your interests beyond the strict boundaries of your discipline or even embark on some interdisciplinary research; it pays to travel and be exposed to cross-cultural influences. How soon you perceive a heureka effect and how good your answer is, depends on many aspects: on the level of expertise, on the thoroughness of study, on cross-fertilization through discussions, interdisciplinary and cross-cultural contacts – and on personal creativity or intuitive capabilities, admittedly hard to define.

Thus, *creative and cognitive processes are relying on the richness of images and concepts* – or also on the richness, even redundancy, of information and knowledge; *they often occur* on a deep level of nonverbal associations, often *subconsciously or intuitively*. Intuition is usually (probably for historical reasons) considered as an opposite concept to rationality. We shall discuss first the concept of rational decisions, then show that it is possible to rationally define and analyze intuition.

2.3 Concepts of Rationality

During the last century, joint developments in statistics and economics resulted in an increasing understanding how economic markets function. This was also related to the study how economic decisions are made and resulted in the development of *decision theory.* Actually, decision theory includes various branches considered today as separate disciplines: mathematical models of analyzing conflicts and cooperation in game theory, general models of decision preferences in value theory, graph theory and theory of relations as applied to decision making, etc. However, all these branches rely on the same concept of rationality, resulting from the study of economic decisions.

In decision theory, *a decision maker is considered rational if she/he chooses the most preferred decision*, defined usually as *maximizing her/his utility or value function*. Actually, such a definition is slightly oversimplified; but the assumption of choice based on preference, on the maximization of a utility or value function has been very productive in the developments of economic theory and mathematical game and decision theory. It has a long research tradition and has led to such an expansion of mathematical economics, game and decision theory that no single book today would even list nor summarize the existing literature of the subject. A *paradigm* in the sense of Kuhn (1970) has been formed around this assumption, with all its features – including also a psychological commitment to this assumption, reinforced by the sociological aspects of a field of science.

There were also many schools of criticism of this central assumption of decision theory, usually dependent on a certain disciplinary or cultural perspective; we shall call such a school a *framework* of perceiving rationality of decisions.

The most notable is the framework coming from managerial and microeconomic perspective (consistent also with an engineering type of behavior). This is the theory of *satisficing behavior* developed originally by Simon (1955), Simon (1957). According to Simon, individual decisions are not optimized because it is too difficult (particularly in situations with uncertainty), because the access to information is limited, and because an individual optimization might lead to unnecessary conflicts in non-market situations with a limited number of actors (as in management). Instead of optimizing, an individual decision maker forms her/his own *aspirations*. They are formed adaptively, through learning processes that play an important role in this theory. The central point of the theory of satisficing behavior is that the decision maker stops searching and is satisfied with a given decision, if the aspiration levels are attained.

The satisficing framework of rationality was further developed through intensive studies, including empirical tests; finally, it was assimilated also in the utility maximization paradigm under the name of *bounded rationality* (this illustrates a self-reinforcing aspect of the paradigm: while believing that a perfectly rational individual should optimize utility, it can be admitted that most individuals are less than perfect). Related to the satisficing framework but distinct by stressing a big organization perspective (in corporate or military planning, also in centrally planned economies) is the framework of *goal-oriented behavior*, discussed *e.g.* by Glushkov (1972), Pospelov and Irikov (1976) independently made operational by Charnes and Cooper (1977) and many other researchers as the *goal-programming method* of multi-objective decision support.

From the perspective of computerized decision support in multi-objective situations, a methodology of *reference point optimization* was later developed – see *e.g.* (Wierzbicki, 1980; Wierzbicki, 1986; Lewandowski and Wierzbicki, 1989; Wierzbicki, 1992c). This methodology uses both optimization and adaptive formation of aspirations, but rather as instruments of interaction between the decision maker and the computer than as assumptions concerning human behavior. While this methodology generalizes the goal-programming method and can support decision makers which adhere to various rationality frameworks (*e.g.* such that believe either in satisficing or goal-oriented or utility-maximizing behavior), it also stresses the dynamics of a decision process and adaptive learning.

There were also many other schools of criticism of the assumption of utility maximization, *e.g.* coming from the psychologist's perspective, such as the *theory of regret* by Kahneman and Tversky (1982). Another important avenue of criticism of individual utility maximization resulted from *the evolutionary perspective*, with the introduction of the concept of *evolution of cooperation*, see Axelrod (1984), Rapoport (1989). This approach shows that evolutionary interests of a population – and also individual interests of members of this population – might be better served by adopting certain individual decision rules that take into account cooperation principles than by pursuing the maximization of purely individual interests (no matter whether short- or long-term). The concept of evolution of cooperation does contradict more extreme interpretations of the paradigm of individual utility maximization (that it is better if everyone pursues her/his own individual interests) and contributes to a better understanding of emerging issues that require common action, such as the importance of preserving clean environment or, more generally, preserving our planet for the survival of the human race.

These were the schools of criticism of the simplified meaning of the term *rationality* related to decision analysis; however, this term is interpreted yet differently in various scientific disciplines. A possibly broadest sense of rationality or validity of scientific statements is implied by the philosophy of science, see *e.g.* (Popper, 1959; Popper, 1975; Popper, 1983). Even in this field, there are diverse interpretations of this concept: *falsificationism* by Popper, *historicism of changing paradigms* by Kuhn, *competition of scientific programs* by Lakatos, *etc.*, see (Hacking, 1964) – versus *evolutionary epistemology* as initiated by Lorentz (1965), see (Wuketits, 1984a). From the more normative, Popperian perspective, rationality of a statement concerning the real world implies its empirical testability and falsifiability (at least in principle); if a statement relates rather to a method of communication than to the properties of the real world itself (as, for example, a mathematical theorem) then it must be logical and consistent with the assumptions of the method.

When seen from this perspective, the term *rationality* in decision theory has been used rather unfortunately. Interpreted as a mathematical statement, the assumption of utility maximization is logical and consistent with many other developments of decision theory. However, interpreted as a description that real *rational* decisions are always made in the best perceived interests of the decision maker, this assumption is itself not rational in the normative, Popperian sense of the theory of science. Popper uses precisely this as an example of an unrefutable theory. If any other concerns – honor, altruism, *etc.* – can be subsumed into the individual utility, then the assumption of best perceived individual interests cannot be falsified in principle and expresses rather an ideology than science – at least, according to Popper, while Kuhn (1970) argues that certain components of ideology are always present in a paradigm of normal science. Thus, we will use sometimes the concept of rationality of a decision maker in its paradigmatic sense, but with an adjective; we shall call it *economic rationality* to distinguish it from other possible perspectives.

We shall use the general term *rational decision* in a possibly broadest sense: while stipulating that the specific meaning of this term might depend on professional and cultural background (see *e.g.* Grauer, Thompson and Wierzbicki (1985) or Yu (1990)), we shall require that a rational explanation of the way decisions are made should itself be – at least in principle – empirically refutable. In this sense, we can speak also about *procedural rationality* in the sense that we can empirically investigate the process of decision making and analyze whether it is reasonable. But before examining decision making processes, we must outline the role of intuition in decision making.

2.4 A Rational Definition of Intuition

The experiences in political decision making result in the following pattern. *The higher the level and experience of a decision maker, the less inclined is she/he to trust in various tools and methods of decision analysis and support.* When asked to explain such an attitude, they point to tacit knowledge, personal experience and intuition. A clear conclusion is that decision scientists must make more efforts to better understand the dichotomy of rational versus intuitive decisions and the role of intuition in decision making processes.

In everyday language, we tend to use the word *intuitive* with some connotation of *irrational*. This is probably due to Bergson (1903), who attached a great importance to intuition but interpreted it as a somehow mystic force which by definition could not be an object of research by rational means of inquiry. However, almost a hundred years of research in various fields of science motivates today a reversal of this interpretation. We shall show that *intuition*, when appropriately defined, *might be a subject of rational inquiry*.

We first turn to an avenue of criticism of the mainstream decision theory which claimed that individual decisions are too complicated to be explained by *analytical* theory, that decisions are rather made in a *deliberative* way. This criticism relates to *general systems theory* (Bertalanffy, 1968) or so-called *soft systems approach*, stresses the role of *synergy* – that the whole is bigger than the sum of its parts – as well as the role of *heureka* (after Archimedes) or *aha effect* – an enlightenment moment in a cognitive process. Thus, soft or deliberative decision making consists in trying to perceive the whole picture, to observe it from various angles, and finding the right decision by expert intuition.

This avenue of criticism is also presented in the book *Mind over Machine* by Dreyfus and Dreyfus (1986) that presents a well documented thesis as follows: *the way decisions are made changes with the increasing level of expertise* of the decision maker. While a beginner, novice or apprentice needs analysis to support her/his decisions, specialists, experts and master experts either make decisions instantly, intuitively, or rely increasingly more on deliberation.

Another reason to analyze the concept of intuition comes from the research on hemispheric asymmetry of the brain, see e.g. Springer and Deutsch (1981). All this research indicates that the typical left hemisphere is somehow related to *verbal, sequential, temporal, digital, logical, analytic, and rational* thought activity, while the right one is rather related to *nonverbal, visuo-spatial, simultaneous, analog and intuitive* thought activity. Best substantiated by experimental results is the distinction between verbal and visuo-spatial processing in distinct hemispheres; further dichotomies are more speculative. Note that in the dichotomies presented above after Springer and Deutsch (1981) rationality is still rather traditionally opposed to intuition.

Nevertheless, we need a definition of intuition that is rational in the sense of rationality of scientific theories: it should be testable experimentally. Such a definition can be obtained from two premises. One is the knowledge derived from modern telecommunications and complexity theory: transmitting images (video signals, typically 2 MHz) requires 10^2 times more volume (broadband) than transmitting speech (audio signals, typically 20 kHz). Since the complexity of processing such signals is certainly not linear, at least quadratic, processing images requires circa 10^4 times more capacity than processing words.

Another is a thought experiment: what happened to human mind at the moment of the discovery of language? *The development of language was a powerful evolutionary shortcut in the history of humanity*: it was easier to think in words and the communication with other individuals not only helped evolutionary in dealing with the environment, but also opened all development of cultural achievements of the race.

However, this shortcut left powerful parts of our brain utilized only subconsciously. If we acknowledge the essential difference of difficulty between processing images and processing words, we can postulate that *our mind works also on a deeper layer of nonverbal image processing; this task employs a large part of the mind's processing potential, is sometimes conscious but often unconscious, and uses rather multi-valued than binary logic*. Our language can only try to approximate this deeper level by words and by a simpler, more decisive binary logic; but words will always have multiple meanings.

Based on such premises, we can *define intuitive decisions as quasi-conscious and subcon-*

scious[1] *information processing, leading to an action,* utilizing aggregated experience and training and performed (most probably) by a specialized part of the human mind.

Note that this definition includes intuitive decisions such that have conscious components but take advantage of the parallel processing by the parts of our mind specialized in subconscious or quasi-conscious operations. Thus, *every day our mind makes many intuitive operational decisions.* The role of training in making these decisions well is fully understood and institutionalized in human society through our education systems. All training in repetitive activities is actually aimed at delegating them to the quasi-conscious level, automating them – by establishing shortened or more easily used connections between synapses, some specialized *roads in the brain.*

However, this definition includes also strategic and creatively intuitive decision making and makes it possible to develop empirical studies of intuitive decisions. While an empirical study of repetitive, operational intuitive decisions might be simpler (we can ask subjects of experiments not to use conscious parts of their minds, we can make sure that they do not use them by turning their attention to other matters or, as done by Dreyfuses, by saturating the conscious part of the mind), such a study of strategic and creatively intuitive decisions will be certainly more valuable. Unfortunately, not many such studies have been concluded yet, see (Wierzbicki, 1993; Wierzbicki and Wessels, 2000).

2.5 Dynamics and Phases of Decision Processes

The above discussions lead us to the following questions: *if most decisions are made intuitively, how can we support them using computerized information and models?* In particular, how can we better understand and support creative or strategic intuitive decision making?

The complexity and the speed of change of information in the modern world make the use of computerized decision support self-evident. However, the reflection on the role of intuition changes the focus of decision support. We can use here the analogy to training in repetitive decision activities and automating them: decision support is aimed at automation of decisions, the issue is only to what extent, at which stages of a decision process and which parts of such a process will be, after all decision support, reserved for human decision makers.

The classical decision theory concentrated almost exclusively on automating the actual choice. In his criticism of the mainstream theory, Simon (1957) defined the following *essential phases of an analytical decision process*:

- **Intelligence** - information and data gathering;
- **Design** - selecting or constructing a model of the decision situation;
- **Choice** - examining decision options and choosing between them.

Later, see *e.g.* (Lewandowski and Wierzbicki, 1989), another essential phase of

- **Implementation**

was added. These phases could be subdivided into more detailed sub-phases. However, such a subdivision of a decision process puts the emphasis on analytical decision making. For the case of creative or strategic, intuitive decision processes, the following subdivision of their phases was proposed in (Wierzbicki, 1992b):

[1] Our mind can work quasi-consciously when we want to do something but leave the details to subconsciousness, or subconsciously when we do not know that we want to do something and are not aware of doing this, but we can become aware a posteriori of the results of subconscious work of our mind.

- **Recognition**;
- **Deliberation or analysis**;
- **Gestation and enlightenment**;[2]
- **Rationalization**;[3]
- **Implementation**.

An essential question is: how to support various phases of an intuitive decision process, as in an analytical decision support? Or, if we cannot support them directly, could we support them somehow indirectly during an analytical decision process?

It is clear that *we must concentrate more on the phases of intelligence and of design and analysis, preceding the phase of choice*. During a decision support process, there must be enough time and opportunities for learning and thus enriching intuition, helping in gestation and possible enlightenment – all this well before the phase of choice or rationalization.

How should these conclusions influence the way of constructing or using decision support systems? Any decision process is actually multi-objective – particularly if it includes elements of intuition and should be creative: it should aim at novel solutions. If the process is supported analytically by computerized information processing and models, the information not only should be presented to the decision maker in graphic terms, but it should also be rich, multi-dimensional; we do not know a priori on which piece of information her/his subconsciousness will focus. In the organization of decision support, we should avoid an early aggregation of objectives into a utility or value function, avoid even pair-wise comparison and other known techniques related to the paradigm of utility maximization, nor should we require consistency from the decision maker – since if she/he knew from the beginning what she/he precisely wants, the solution could not be creative. The support should be rather concentrated on helping to generate new options, even to formulate new problems, on organizing the dynamics of the decision process in such a way that it leaves enough time for gestation.

However, if we concentrate on multi-objective decision theory and optimization, we can use the tools of this field in order to support the intuition of an analyst rather than to replace his intuitive power of choice by a computerized system. This is achieved by applying such tools to the earlier phases of problem formulation and analysis and by *treating optimization as a flexible tool of analysis, not as the goal*. In particular, optimization can be used as a tool of *multi-objective model simulation* which takes into account hard and soft constraints, includes inverse simulation problems (in which model output, not model input is specified by the analyst) and various aspects of multi-objective model analysis. This provides a sufficiently flexible toolbox to help in treating computerized models as a kind of virtual reality, in which the attainability of various aspirations and the consequences of various approaches to the decision problem are analyzed.

2.6 The Decision Maker of the Future

We observed that decision support systems are used today mostly by analysts or modelers. Although this might change in generations to come, we should not expect this change too soon.

[2] This is an extremely important phase – we must have time for forgetting the problem in order to let our subconsciousness work on it. The expected heureka effect might come but not be consciously noticed; for example, after a night sleep it is simply easier to generate new ideas (which is one of the reasons why group decision sessions are more effective if they last at least two days).

[3] For example, when writing a paper, we obviously rationalize our deeper thoughts – and sometimes also advertise them. Note the similarity of this phase to the classical phase of choice.

Note that a high-level political decision maker is elected because of her/his ability and quality of intuitive generalizations in decision making – thus, she/he will not use computerized decision support unless she/he grows up in a culture where such tools are used everyday and by everybody.

On the other hand, the volume and the speed of change of information processing will grow together with the maturing of the information society. The education systems take these facts into account already today. With the development of information society, the role of education will also substantially change. As the information processing becomes one of the essential productive factors, the access to information and the ability to process it with modern means will become one of the main social issues. Thus, the access to good education, adequate for the challenges of information society, will be essential for a success in life. However, education will at the same time concentrate less on preparing for a designated profession or position in life, and more on providing general abilities and teaching a system of concepts adequate for understanding the new civilization era of information society. All this indicates that computerized decision support will be used much broader in the future than today.

What characteristics, then, will or should a future decision maker have? Evidently, she/he must be able to use various computerized information and model processing tools. However, even more important is being more informed in the sense of basic concepts that allow to understand the coming cultural era. Finally, in decision making, she/he should know in what proportions to blend computerized, analytical decision support and her/his intuition – or, in other words, how to use decision support in order to enlarge her/his *habitual domain*, see (Yu, 1995).

3 Modeling for Decision-making Support

In previous sections we have discussed characteristics of modern decision makers, of decision problems, and of decision making processes, as well as the role of mathematical models in representing knowledge relevant also to decision making processes and thus supporting making better decisions. In this section we outline the characteristics of models, and of modeling processes that pertain to modeling activities aimed at decision-making support for complex problems.

One needs to stress that such characteristics are different from those of models used in introductory text books that focus on providing basic knowledge on mathematical modeling. Small and simple models are typically used for educational purposes to illustrate basic concepts, methods, and techniques. However, adequate model-based support for complex problems typically requires also complex and/or large models, which in turn require also more complex modeling methodology and tools than those adequate for the toy-type models. Therefore, in this section we focus on modeling issues that are relevant for model-based support of decision-making for complex problems.

3.1 Context

Generally speaking, mathematical modeling is the process of creating a model, which is an abstract representation of a system (or a phenomenon, or a problem) developed for gaining a better understanding of the modeled object. A purpose must be defined for any modeling project because modeling even a small part of reality and using even a part of accumulated knowledge would inevitably result in a model far too complex to be implemented using justifiable resources. Consider e.g., modeling a cup of coffee. Very different models are suitable for studying various aspects, e.g., how something (sugar, cream) is dissolved in a content of the cup, or in what conditions the cup might break from thermal stresses, or what shape of the cup is most suitable for use in aircraft, or how a cup of coffee enhances productivity of an individual or of a group. An attempt to develop a model that covers all these aspects, and represents all accumulated knowledge on even such a simple topic like a cup of coffee would hardly be rational.

A mathematical model describes the modeled object by means of variables, which are abstract representations of these elements of the object that the users of the model want to study. Many commonly used models can be classified as Algebraic Models (AM).[4] AMs are widely used for supporting decision-making[5] in many areas of science and industry for predicting the behavior of a system under particular circumstances, when it is undesirable or impossible to experiment with the system itself. The understanding of the system gained through a comprehensive examination of its model can greatly help in finding decisions (controls) whose implementations will result is a desired behavior of the system. Thus AMs support finding better solutions to real problems than those that could be found without model-based problem analysis.

AMs have many common analytical features and are used whenever decisions require analyses of a large amount of data and logical relations, in a wide range of application domains including (but not restricted to) planning problems in environmental systems analysis, telecommunication, logistics, transportation, finance, marketing, production, distribution, as well as in science, research and education. The systems that are modeled have very different characteristics (including the nature of the underlying physical and/or economical processes, their complexity, size, types of relations between variables). There is also a great variation in the use of models, which depends on various factors (like the decision making process, the background and experience of model users, available data, resources assigned for modeling activities). However, modeling activities have many similarities also when the modeled systems are very different.

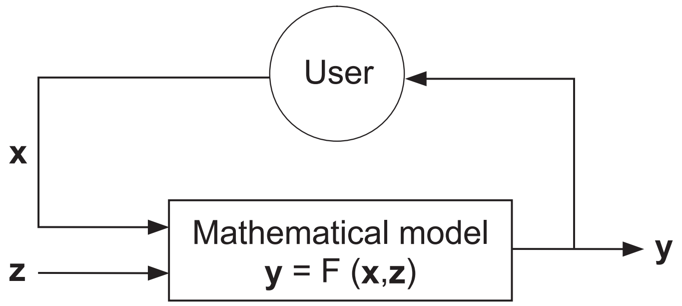

Figure 1: A mathematical model represents relations between decisions (inputs) $\boldsymbol{x}$, external decisions (inputs not controlled by the user) $\boldsymbol{z}$, and consequences (outcomes) $\boldsymbol{y}$.

The concepts essential for explanation of model-based decision support are illustrated in Figure 1. The basic function of a DSS is to support the user in finding values of his/her decision variables, which will result in a behavior of the system that fits best to the preferences of the user. A model-based DSS is typically developed using the following concepts:

[4]Following the Oxford Dictionary and a common understanding, we use the term *Algebraic Model (AM)* for a set of relations (such as equations or inequalities) between quantitative inputs (decisions) and outputs (performance indices) that measure consequences of implementation of decisions.

[5]The term *decision support* is typically associated with management and policy-making but in practice similar model-based activities are being performed also in almost all fields of industry and research.

- decisions (inputs) $\boldsymbol{x}$, which are controlled by the user,
- external decisions (inputs) $\boldsymbol{z}$, which are not controlled by the user,
- outcomes (outputs) $\boldsymbol{y}$, used for measuring the consequences of implementation of inputs,
- mathematical model F, which represents the relations between decisions $\boldsymbol{x}$ and $\boldsymbol{z}$, and outcomes $\boldsymbol{y}$, which can be symbolically represented by: $\boldsymbol{y} = F(\boldsymbol{x}, \boldsymbol{z})$.

Before discussing these basic concepts in the following subsections we will introduce an example of a complex model, which will be used to illustrate not only these concepts but also several other issues.

3.2 RAINS Model

In order to be able to later present various modeling problems we outline now an illustrative example of the RAINS model used for supporting international negotiations aimed at improving the European air quality, and described in more detail by Amann and Makowski (2000).

RAINS provides a consistent framework for the analysis of emission reduction strategies, focusing on acidification, eutrophication, and tropospheric ozone. RAINS comprises modules for emission generation (with databases on current and future economic activities, energy consumption levels, fuel characteristics, etc.), for emission control options and costs, for atmospheric dispersion of pollutants, and for environmental sensitivities (i.e., databases on critical loads). In order to create a consistent and comprehensive picture of the options for simultaneously addressing the three environmental problems (acidification, eutrophication, and tropospheric ozone), the model considers emissions of SO_2, NO_x, ammonia (NH_3), and VOC (volatile organic compounds). The quality of air is assessed by several indicators computed at a few hundred grids, and their values depend on locations and amounts of emissions of various pollutants. Hence the decision (input) variables are emissions (represented in this simplified presentation by a vector $\boldsymbol{x}$). The vector of output variables $\boldsymbol{y}$ includes subvectors of various air quality indicators, each composed of vectors representing values of such indicators at each of the grids into which Europe is divided for the purpose of air quality assessment.

The simplest model of air quality can be defined by $\boldsymbol{y} = F(\boldsymbol{x})$, where an operator $F(\cdot)$ represents relations between emissions $\boldsymbol{x}$ and air quality indicators $\boldsymbol{y}$. In the actual implementation these relations are defined by a collection of several thousands equations. Depending on the type of pollution, components of $F(\cdot)$ have different forms (e.g. linear for sulphur oxides and non-linear for trophosperic ozone). Such a model can be used for computing air quality corresponding to given emissions. However, it is not very useful for finding cost-effective strategies to improve air quality. For this purpose one needs to develop another model, which relates emission levels to costs of reducing them, and then to combine/couple both models. Such a combined model can be used for finding minimum-cost strategies for achieving air quality that conforms to a given standard, or for maximizing an overall index of air quality for given cost limits, or for examining various trade-offs between costs and air quality. The combined model contains more outcome variables, namely those representing costs of using various technologies that result in corresponding emission levels. Separate models need to be developed for different types of emissions, with corresponding air quality indices, and it is rational to re-combine them into one model, because some measures for decreasing emissions impact on more than one type of emission. Moreover, air quality should be assessed for several types of indices jointly, hence all relevant types of pollution should also be considered jointly.

Why not start with the "full" model instead of gradually developing one? The gradual approach has evolved in practice. First, a relatively simple (linear, with few hundred variables) model was developed, and used in practice. Then various needed submodels were developed, validated separately, and combined in larger models, until a multi-pollutant model was obtained, a complex (non-linear, non-convex with about 30,000 variables) model equipped with several methods and tools for its analysis. These submodels have been developed by several collaborating institutions, which in turn use other models to compute the data needed for specification and validation of the developed models. Such a bottom-up approach is typical for development of complex models, which are usually composed of submodels developed and validated separately.

3.3 Multi-paradigm Modeling

The RAINS model is an example of modeling a real-life complex problem, but it is by no means unique. Similar approaches are applied in all fields of research and policy-making, when models are used for solving complex problems. Over time the complexity of the problems being solved has grown, and therefore the requirements of users have also grown substantially. In order to meet such requirements, the modeling technology faces new challenges: models are not only becoming more and more complex, their development typically requires results from other models, many of them using huge amount of data; in many cases a model is composed of submodels developed by various teams of specialists in different disciplines. Therefore, it is practically impossible to rationally select one modeling paradigm for a best model representation of a complex problem. On the contrary, one needs to experiment with different modeling paradigms to finally find the most appropriate one. Moreover, best paradigms are often different for different submodels.

The development of diversified modeling paradigms has been to a great extent driven by the needs of supporting decision-making for problems for which existing methods have not provided adequate support. In order to address these needs diverse types of models were developed to possibly best represent various problems by a selected type of model. These types of models are characterized by diverse types of variables and relations between them, e.g. *static, dynamic, continuous, discrete, deterministic, stochastic, set-membership, fuzzy, softly constrained, etc.* Moreover, various methods of model analysis (e.g. simulation, optimization, softly constrained inverse simulation, multicriteria model analysis) have been developed to best support various types of modeling tasks for different purposes and/or users. Finally, due to the growing complexity of various computational tasks, so-called *solvers* (dedicated software for performing such tasks) have become more and more specialized, even for what was originally the same type of mathematical programming problem. Each modeling paradigm embodies a lot of accumulated knowledge, expertise, methodology, and modeling tools specialized for solving many of the problems belonging to each modeling paradigm.

Multiparadigm modeling, defined as an efficient application of all pertinent modeling paradigms, is one of the key issues of modeling complex problems. In some situations it is possible to use a more general paradigm, which “includes” a simpler paradigm, e.g. building a non-linear model that contains a linear part.[6] Typically, instead of using a more general paradigm (one can formally treat a linear model as a non-linear one), it is rational to use a *unifying paradigm*, e.g. supporting non-linear models with a (possibly large) linear part. However, in other situations it is not practicable to unify different paradigms. In such situations one needs to *switch paradigms*. The substantial difference between switching (within a properly organized modeling process) paradigms, and applying different paradigms “independently” consists of appropriate handling of these elements of modeling process, which are common for different paradigms, and to support comparative analyses of results obtained with the help of applied paradigms.

[6]However, even in such a case it is typically rational to use non-linear and linear paradigms to respective parts of such a model.

While various modeling paradigms may differ substantially, most of them are based on a number of common concepts and elements, which are briefly discussed below.

3.3.1 Decision Variables

In model-based decision support it is assumed that decisions have quantitative characters and therefore can be represented by a set of the model variables, hereafter referred to as decisions $\boldsymbol{x} \in E_x$, where E_x denotes a space of decisions. In a trivial case $x \in R$, which denotes that a decision is represented by a real number. However, in most cases $\boldsymbol{x}$ is a vector composed of various types of variables. For larger problems the components of $\boldsymbol{x}$ are aggregated into various types of algebraic structures.[7]

In the RAINS model the vector of emission levels $\boldsymbol{x}$ is composed of subvectors $\boldsymbol{x_i}$, where the index i corresponds to countries. Each of $\boldsymbol{x_i}$ is in turn a vector composed of vectors $\boldsymbol{x_{i,j}}$, where the index j corresponds to types of pollution. Each of $\boldsymbol{x_{i,j}}$ is also a vector composed of vectors $\boldsymbol{x_{i,j,s}}$, where the index s corresponds to industrial and consumption sectors composed of activities that emit air pollution. Each of vectors $\boldsymbol{x_{i,j,s}}$ is composed of variables representing emission levels associated with various technologies that belong to s-th sector of i-th country, and emits pollution of type j. Hence, $\boldsymbol{x}$ is actually a four-dimensional matrix, whose dimensions correspond to countries, pollution type, sectors, and technologies, respectively. Additionally, optional decision variables are considered for scenarios that allow limited violations of air quality targets. For such scenarios variables corresponding to each type of considered air quality targets are defined for each receptor. Such variables represent violations of given environmental targets. Optionally, violations of targets can be balanced with surpluses (understood as a part of the difference between a target and its corresponding actual value).

3.3.2 External Decisions

Figure 1 on page 15 illustrates two types of inputs to the core model: decision variables $\boldsymbol{x}$ controlled by a user, and external decisions denoted by $\boldsymbol{z}$. In practice inputs $\boldsymbol{z}$ may include representations of various quantities that substantially influence the values of outcomes $\boldsymbol{y}$ but are not controlled by the user, for example:

- Regulations or commitments on environmental standards for air or water quality management models.
- Meteorological conditions assumed for modeling physical relations in environmental models, e.g. *an average*, or *a wet*, or *a dry*, or *a worst* year data for a water model.
- Forecasts of changes in demand for services, e.g. in telecommunication or transportation models.

In the RAINS model the external decisions $\boldsymbol{z}$ are represented by:

- Values representing the environmental standards that define constraints for various indices (such as maximum concentrations of various air quality indicators).
- Set of meteorological data used for calibration of the air pollution transport submodel.

While the external decisions are beyond control of the user of a DSS, he/she typically wants to examine a range of scenarios with various representations of external decisions in order to find out not only a solution which will best respond to a most likely representation of external

[7] Actually, components of $\boldsymbol{x}$ are often implemented as multidimensional (typically sparse) matrices. However, for a simplification of the presentation we use the term *vector* for containers of various types of variables used in this chapter. Also for the sake of brevity we call decision variables simply decisions, and we use the term *decision* for vectors of decisions $\boldsymbol{x}$.

inputs z but also a solution that will be *robust*, i.e. will be good also for various other compositions of z that should be considered. From the mathematical programming point of view, the external decisions are represented as given fixed values of variables, and therefore are treated as parameters of the model.

3.3.3 Outcome Variables

The consequences of the implementation of various decisions x are evaluated by values of outcome variables $y \in E_y$. In various fields of applications outcome variables are named differently, e.g. outcomes, metrics, goals, objectives, performance indices, attributes.

In the RAINS model one outcome variable represents the sum of costs of reductions of emissions, and four sets of other outcome variables correspond to various indices of air quality. While the definition of the cost is conceptually rather simple[8], an appropriate definition of air quality indices is rather complex. Environmental effects caused by acid deposition, excess nitrogen deposition (described by a two-element linear critical loads function), and by eutrophication are evaluated at each receptor by PWL (piece-wise linear) functions that represent accumulated excesses over the thresholds of the environmental long-term targets. If optional violations of environment standards are allowed, then a maximum (over a set of receptors) violation of each type of air quality indicator is also considered as an output variable. Moreover, other outcome variables are based on measures of a distance between the emission levels, and the so-called reference levels, which correspond to various commitments.

In other models a great variety of variables is used as outcome variables. For an illustration we mention: various types of costs (e.g. investment, operating and maintenance, and total annual cost), various probabilistic estimations (e.g. expected income, probability of insolence), attributes of elements of a solution (often represented by variables of the linguistic type, e.g. having values such as excellent, good, average, bad).

3.3.4 Types of Variables

The above discussed taxonomy of variables (decision, external decisions, outcome)[9] is of a primary interest for users of a model. There is however another taxonomy of variables, namely according to the type of variable in the sense of mathematical representation, e.g. continuous, integer, binary, fuzzy, stochastic, dynamic, linguistic.[10] In many situations a choice of the variable type is determined by its nature, e.g. a continuous variable for a concentration, binary variable for a logical (yes or no) choice. However, in many other situations a choice of variable's type implies a type of simplification made for a model specification, e.g. a continuous variable for a number of cars produced by a factory.[11] A choice of variables' types is often a result of a compromise between adequacy of a problem representation and resources needed for the development of a corresponding model. Such a choice is done together with a choice of type of relations, which is discussed below. Both choices require a deep understanding of their consequences, which can hardly be formally assessed. Therefore they are based on a mixture

[8] However its implementation is complex, and it requires advanced methodological and software approaches described by Amann and Makowski (2000) and Makowski (2004), respectively.

[9] In fact in a typical complex model a majority of variables are so-called *auxiliary variables*, which are introduced for various technical reasons, and are therefore not interested for users.

[10] A linguistic variable is an enumeration (using a set of descriptive labels, such as high, average, low) of a finite (typically small) number of values of integer variables (or of ranges of continuous variables).

[11] Such a choice is often done consciously – because rounding large real values to a nearest integer has a very small relative error, and solving an integer programming problem is much more difficult than a relaxed (when integer variables are replaced by continuous) one.

of modeler's knowledge, experience and intuition. This is why modeling complex problems remains to be a kind of art (or craft), see e.g. (Paczyński, Makowski and Wierzbicki, 2000) for a more detailed discussion of related topics.

3.3.5 Core (Substantive) Model

A mathematical model used for model-based decision support is often referred to as a *core model*, or a *substantive model*, to stress its role, which is to provide an evaluation of consequences that will result from an implementation of given decisions $\boldsymbol{x}$ and $\boldsymbol{z}$ without imposing any constraints on the user's preferences.[12] As already illustrated in Figure 1 on page 15, a mathematical model is used for predicting the consequences of decisions $\boldsymbol{x}$, which can be either proposed by a DM or computed by a DSS. The consequences are measured by values of outcome variables $\boldsymbol{y}$. Therefore, a model can be represented by mapping $\boldsymbol{y} = F(\boldsymbol{x}, \boldsymbol{z})$, where $\boldsymbol{x} \in E_x$, $\boldsymbol{z} \in E_z$, and $\boldsymbol{y} \in E_y$ are vectors of values of decisions, external decisions, and outcomes, respectively. For the sake of brevity we will assume further on that the external decisions z are given and represented as parameters of the mapping F.

In other words, a core model is composed of decision, output and auxiliary variables, and of the relations (inequalities, equations, etc.) between these variables that indirectly determine the set of admissible (feasible) decisions and corresponding solutions.

As already mentioned, a model specification (definitions of variables and relations between them) is based on both understanding of the modeled problem, and on modeling knowledge, experience, and intuition. It is therefore impossible to provide a complete prescription for model specification. Hence we provide only several general guidelines.

A model is a representation of a part of knowledge done for a specific purpose. As illustrated by the example of a cup of coffee in Section 3.1, such representations even of the same problem may be very different. The following practical examples illustrate some of the related issues:

- Not always much more detailed models provide better representations. A good illustration of this statement is given by Edwards (2001): the prediction of the temperature rise by doubling CO_2 in the atmosphere made in 1903 by Arrhenius is similar to predictions by modern climate models run on supercomputers.
- Application of integer variables and/or non-linear relations typically increases dramatically the demand of computing resources. In many situations a relaxed (instead of integer) problem and/or piece-wise linear (or quadratic) approximations of non-linear relations provide a satisfactory simplification of the more accurate model.
- Even for a given mathematical problem an alternative formulation (which is equivalent from the optimization point of view) can be dramatically easier to solve, see e.g. (Ogryczak, 1996) for a practical example.
- Models described in text books (or used as tutorial examples in modeling software) are typically rather simple, and the corresponding modeling technology (the way of model specification and analysis) is adequate to such models. However, for complex and/or large models (typically composed of sub-models) a very different modeling technology needs to be used, see e.g. (Makowski, 2004) for more detail.

Models are often classified into white-box or black-box models, according to how much knowledge about the types of relations between variables is used. Typically, a priori knowledge about the problem is used directly for a model specification. While this is a natural and recommended approach, sometimes the resulting model would be too complex to be actually used. In such situations it is often rational to apply a specific type of analysis to a more detailed model,

[12]The adjective *substantive* has been used for core models in order to stress their distinction from *preferential* models that express the preferences of decision makers.

which provides a basis for a specification of a simplified model, to use the simplified model for more computing resource demanding types of analysis (e.g. optimization-based), and then to check selected results with the detailed model. Such an approach has been successfully used e.g. for the RAINS model described above.

However, for many problems there is no a priori information about the relations between variables that can be effectively used for model specification. In such situations black-box models are often built based on analysis of available data. There are many successful implementations of this type of models. However, much less is published about pitfalls and risks related to using models built without a priori analytical knowledge. Such models typically rely on statistical analysis of available data, which sometimes provide also experienced modelers with very wrong results, see e.g. (Knight, 2002) for a recent example. However, even a correct statistical analysis may result in wrong conclusions. Consider for example analysis of a large sample of data from observations of two variables: pressure p and volume v of a gas contained in a cylinder with a variable volume. A statistical analysis of such laboratory data will likely result in a conclusion that $p \times v = const$ unless a person uses a priori knowledge, and includes the temperature into consideration. Such a trivial modeling mistake is unlikely but it illustrates the importance of using the relevant knowledge in model specification. A more elaborated discussion of simple examples can be found e.g. in (Cartwright, 1999).

Together with defining the mapping F, the core model defines a set of feasible decisions $X_0 \subseteq E_x$. In other words, $\boldsymbol{x}$ is feasible, if and only if $\boldsymbol{x} \in X_0$. The set X_0 is usually defined only implicitly by a specification of a set of constraints that correspond to logical and physical relations among all the variables used in the model.

The substantive model should include only logical and physical relations that are necessary to adequately represent relations between inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$. In addition to inputs and outputs, a model contains various intermediate and parametric variables (balance and/or state variables, resources, external decisions), conventionally called auxiliary variables. In a typical complex model decision and output variables are a small fraction of all variables. However, auxiliary variables are introduced for easing the model specification and handling, and are typically not interesting for an end-user of the model.

Therefore, the specification of a substantive model that defines F and X_0 *should include neither any relations that reflect conditions for acceptability of a solution by a user nor a preferential structure of a DM*. In other words, a *core model* accounts only for logical and physical relations between all the variables that define the mapping F and the set X_0 of feasible solutions. All other constraints and conditions that implicitly define acceptability of a solution by a user and those that represent a preferential structure of a DM should be included into an interactive procedure of the model analysis.

Finally, we should point out that the value of a mathematical model as a decision aid comes from its ability to adequately represent reality. Therefore, there is always a trade-off between the requested accuracy (realism) of the model and the costs (also time) of its development and providing the model with data. Hence the requested accuracy should be consistent with the accuracy really needed for the model and with the quality of the available data.

3.3.6 Substantive Model versus Mathematical Programming Models

A reader familiar with mathematical programming may be surprised, that a core model does not contain any goal function. This is a very different approach from the traditional formulation of mathematical programming problems in the form:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in X_0} \mathcal{F}(\boldsymbol{x}) \tag{1}$$

In order to provide a solution that corresponds well to the preferences of the user, the problem (1) requires:

- A unique specification of one criterion $\mathcal{F}(\boldsymbol{x})$ that adequately represents a preferential structure of a DM, which is very difficult, if at all possible, for most real-life situations.
- A definition of the set of feasible solutions X_0, which is typically not fixed, because it depends on definition on constraints corresponding to those objectives of the user, which can not be included in $\mathcal{F}(\boldsymbol{x})$.

This distinction between mathematical programming models and core (substantive) models is a very important one. Using substantive models is the recommended way of implementation of any model-based DSS.

We shall explain now *why* the core model should not contain any representation of a preferential structure of a DM.

It is usually not possible to uniquely specify a model that can yield a unique solution reflecting the preferences of a DM. For example, very often it is practically impossible (even for a good analyst or an experienced DM) to specify e.g. values for a group of constraints that would cause a feasible solution that corresponds well to preferences of a DM. In order to illustrate this point let us consider the regional water quality model described by Makowski, Somlyódy and Watkins (1996). A DM typically considers different waste water treatment technologies and the related costs, as well as standards for water quality. However he/she knows that specification of constraints for a group of (either ambient or effluent) water standards may lead to solutions that are too expensive. On the other hand, assuming constraints for costs (with water quality standards being goals) could result in an unacceptable water quality. Values of constraints are in such cases formally parameters in a corresponding optimization problem. But those values are in fact decisions that reflect the preference structure of a user. Setting constraints' value too tight would result in restricting the analysis of the problem to a (possibly small) part of feasible solutions (often resulting in analysis of only very expensive solutions, or even in making the set X_0 empty). A typical advice in such situations is to specify two types of constraints, so called hard and soft constraints which correspond to *must* and *should* types of conditions, respectively. But, in fact, dealing with soft constraints can easily be done within multiobjective model analysis, which is discussed in Section 3.6.

3.4 Modeling Process

Modeling is a network of activities, often referred to as a *modeling cycle*, or a *modeling process*, composed of:

- problem formulation,
- model (symbolic) specification,
- collection, cleansing, and verification of data,
- model implementation, verification and validation,
- model analysis,
- model management, including documentation of the whole modeling process.

Typically, a modeling process starts with an analysis of the problem, including a description of objectives and questions to be answered. Subsequently, a conceptual (qualitative) version of a model is set up to support further discussions between modeler and user. In this phase selections have to be made of types of variables and the mathematical relations between them, to be used for calculating answers with the model. In the next step the modeler translates this conceptual model into a *model specification*. The latter is of a generic nature. It is composed of mathematical (symbolic) relations, and implemented using either a general purpose modeling tool or by developing a problem specific model generator. Different types of variables and relations are used depending not only on the kind of the modeled problem, but also on the choice of model type that is relevant to its future use, available data, and resources for model development, analysis and maintenance. For any non-trivial problem, model specification is an iterative process which involves a series of discussions between developers (typically OR specialists) and users until a common understanding of the problem and its model representation is agreed. Substantial changes of model specification are usually made during this process.

A *model instance* is defined by the model specification and a selection of data that define parameters of its relations. During the model implementation several model instances are created and tested in order to verify that the symbolic model specification is properly implemented. Model instances differ by various selections of data used for *instantiations* of the model specification, which typically corresponds to various assumptions about the modeled problem. Typically many instances of a model are used for different sets of data corresponding to various assumptions that the user wants to examine in order to check to what extent the model adequately represents the problem. The data typically come from different sources (often also as results of analysis of other models); therefore, assembling and making data complete and consistent (e.g. defined in units consistent with specification of model relations) is a resource consuming process. An instance of the model is also called a *substantive model*, and is composed of relations between variables defined by a selected set of data, which is used for the definition of the model parameters.

The next phase of the modeling process is *model analysis*. A typical decision problem has an infinite number of solutions, and users are interested in those that correspond to their preferences (assumptions, trade-offs), which is often called *preferential structure* of the user. A preferential structure takes different forms for different ways of model analysis, e.g. for the classical simulation it is composed of values of input variables, for the single criterion optimization it is defined by a selected goal function, for multicriteria model analysis it is defined by an achievement scalarizing function. A preferential structure typically induces partial ordering of solutions (characterized by output variables) obtained for different combinations of values of inputs. Preferential structure in a well-organized modeling process is not included in the core model, but it is defined during the model analysis phase, when users typically modify their preferences.

A properly organized analysis of a model is the essence of any model-based problem support. Properly organized means that the user is supported in using all relevant methods of analysis, comparing the results, documenting the modeling process, and also in moving back to the first stage, whenever he/she wants to change the type of the model (e.g. for handling uncertainty, or imprecision of model parameters using a different type of variables or relations). During the model analysis different *computational tasks* are generated and solved by *solvers*, which are software specialized for specific types of mathematical programming problems.

While the scheme of the modeling process has not changed substantially over time, the complexity of each of its components has substantially increased. Therefore for complex models one needs to use a new modeling technology that adequately supports development of such models, see e.g. (Makowski, 2004) for a review.

3.5 Model Analysis

While the users need to be involved in the whole modeling process, they do not need to understand all methodological and technical details of its elements. However, some more detailed understanding of the methods for model analysis is helpful for using them efficiently. Therefore further on in this chapter we concentrate on one of the most successful approaches to analysis of complex models, namely on the Reference Point Optimization method. Here we outline the links of this method with the whole modeling process.

The first step is a selection by a user, out of the set of outcome variables $\boldsymbol{y} \in E_y$ of a subset of objectives $\boldsymbol{q} \in E_q$, where E_q is a space of objectives. Quite often objectives are referred to as criteria, and in this chapter these two terms will be used interchangeably. Usually E_q is a subspace of E_y, that is, the DM select some criteria q_i from the set of outcomes E_y. Sometimes also some of the decision variables $\boldsymbol{x}$ are used as criteria, but for the sake of consistency we assume that such a variable is simply represented by one of the outcomes $\boldsymbol{y}$. Such a set of objectives is typically modified during model analysis.

A partial preordering in E_q is usually implied by the decision problem and has obvious interpretations, such as the minimization of costs competing with the minimization of pollution. However, a complete preordering in E_q cannot usually be given within the context of a mathematical programming model. In other words, it is easy to determine for each objective separately, which solution (represented by vectors $\boldsymbol{x}$ and $\boldsymbol{q}$) is the best one. However, for conflicting objectives there are two sets of solutions:

- Pareto-optimal (often called efficient), i.e. a solution, for which there is no other solution with at least one criterion that has a better value while values of the remaining criteria are not worse.
- Dominated, i.e. solutions which are not Pareto-optimal.

Obviously, a Pareto-optimal solution is preferred over any dominated solution (assuming that the selected criteria represent well the preferential structure of a DM). However, a set of Pareto-optimal solutions (often called Pareto-set, or Pareto frontier) is typically composed of a infinite number of solutions, many of which are very different. Pareto-optimal solutions are not comparable in a mathematical programming sense, i.e. one can not formally decide which is better than another one.

However, DMs are able to express own preferences for various efficient solutions. One of the basic functions of multiobjective decision support is to provide various ways in which a DM may specify her/his preferences, analyze the resulting solutions, and use the results of comparative analysis of various solutions for modifications of preferences. There is no reliable formal way for separating a specification of preferences from a process of learning from the model analysis. It is a commonly known fact that decision making is not a point event, even in situations where it is realistic to assume that the problem perception does not change during the DMP. Therefore, the possibility of using a DSS in a learning and adaptive mode is a critical feature, which has been a key motivation for the development of the reference point method discussed below.

3.6 Reference Point Optimization

The reference point optimization, often also called the reference point methodology, or the aspiration-based methodology of multi-objective optimization, has been presented in detail elsewhere, see *e.g.* (Lewandowski and Wierzbicki, 1989; Wierzbicki et al., 2000). Here we shall recall some basic concepts related to this methodology starting, however, with some concepts basic for multi-objective optimization.

3.6.1 Pareto-Optimality and Efficiency

We mentioned in Section 3.5 the concept of Pareto-optimality also called vector-optimality or multi-objective optimality. Its essence is the evaluation of a decision or its outcome by several criteria (selected between decision outcomes): the decision is Pareto-optimal if we cannot improve (find another decision that improves) one of the criteria outcomes without deteriorating other criteria outcomes. Strictly speaking, the term Pareto-optimal is usually reserved for when we either maximize or minimize all criteria. In more general cases, when we minimize some criteria, maximize others, or even do different things with other criteria (e.g., keep them close to prescribed values), we speak about efficient decisions and outcomes.

The concept of efficiency is essential for model-based decision support, because it helps to select such decisions and outcomes that might be interesting for the decision maker or decision support system (DSS) user. If we define only one criterion, represented by a so-called objective function in a mathematical programming model, then its optimization usually defines only one decision and its outcomes.

Because there are usually many efficient decisions, their evaluation and overview must be organized somehow. Suppose that the general form of a substantive model for decision support is:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}); \quad \boldsymbol{x} \in X_0. \tag{2}$$

In this model $\boldsymbol{x} \in \mathbb{R}^n$ denotes a vector of decision variables, X_0 is a set of admissible decisions that is usually defined by a set of additional inequalities or equations called constraints, and $\boldsymbol{y} \in \mathbb{R}^m$ is a vector of model outputs or decision outcomes.

$Y_0 = \boldsymbol{f}(X_0)$ is called the set of attainable outcomes. The modeler, when analyzing the substantive model, might specify several model outputs as especially interesting; we call these objectives or criteria, and denote them by $q_i = y_j$, forming an objective vector $\boldsymbol{q} \in \mathbb{R}^k$, a vector in the objective space. While this vector and space might change during the decision process (according to specific tasks and changes of preferences specified by the modeler), we denote the relation between decisions and their outcomes by $\boldsymbol{q} = \boldsymbol{F}(\boldsymbol{x})$. $Q_0 = \boldsymbol{F}(X_0)$ is called the set of attainable objectives.

Let us assume that all objectives are, e.g., maximized. A Pareto-optimal decision and its outcome are such that there are no other admissible decisions and thus attainable objectives that would improve any objective component without deteriorating other components. A decision and its outcome that are not Pareto-optimal are called Pareto-dominated. Equivalently, Pareto-optimal decisions and outcomes can also be called Pareto-nondominated.

The sets of Pareto-optimal decisions and outcomes typically contain many elements, not just a single decision and its outcome. Thus, Pareto-optimality is an essentially weaker concept than single-criterion optimality. Pareto-optimality does not tell us which decisions to choose, it tells us only which decisions to avoid. This nonuniqueness of Pareto-optimal decisions has been considered a drawback in classical decision analysis; thus, in addition to a substantive model, a complete preferential model was usually assumed that specified a definite utility or value function whose maximum defined – hopefully, uniquely – “the optimal” decision and outcome.

Such an approach, however, is not necessarily the best for interactive decision support, where the user of the DSS (or the modeler in our case) is supposed to exchange information with the DSS, experiment with the model and modify possible solutions. The nonuniqueness of Pareto-optimal decisions is then an advantage, not a drawback. To use this advantage, we need only an additional way of controlling the selection of Pareto-optimal decisions by parameters specified by the user.

There are essentially two main methods of parameterizing Pareto-optimal decisions:

- By using weighting coefficients, i.e., specifying how much relative importance we assign to various objectives. Mathematically, the method corresponds to, e.g., maximizing the weighted sum of all objective functions over the set of admissible decisions. When the weighting coefficients are all positive, the maximization of the weighted sum results in Pareto-optimal decisions. However, more important is the issue of whether we could produce all Pareto-optimal decisions (which is called a complete parametric characterization of the Pareto frontier). When using the maximization of a weighted sum, we can sometimes produce all Pareto-optimal decisions and outcomes by changing weighting coefficients, but only under restrictive assumptions – e.g., the set of attainable objectives must be convex (or even strictly convex in order to get unique answers).

- By using goals or reference objectives in decision space, i.e., specifying what objective outcomes we would like to achieve. This method might work in a much more general setting than the method of using weighting coefficients, but it is more complicated mathematically. At first glance, an appealing mathematical method would be to minimize a distance measure or simply a norm of the difference between the goal and the attainable objective vector. Such techniques of norm minimization were first used historically, either in the displaced ideal method of Zeleny (1974) or in the broad family of goal programming techniques starting with the work of Charnes and Cooper (1977). However, simple examples show that norm minimization might produce decisions that are not Pareto-optimal, thus additional assumptions are necessary. They amount, generally, to limiting the use of goals to objective values that are highly unrealistic. This motivated the development of a different approach – the reference point approach – that uses reference objectives that can be realistic, but avoids norm minimization and instead uses more complicated functions to be optimized (usually, maximized), called order-consistent achievement functions.

Thus, reference point methodology could be considered as a generalization of goal programming, aiming at using arbitrary (not only unrealistic) goals or reference objectives and obtaining only efficient outcomes, at the cost of avoiding norm minimization and replacing it by optimization of a more complicated function. We shall discuss now the relations between these methods in more detail.

The main advantages of goal programming are related to the psychologically appealing idea that we can set a goal in objective space and try to come close to it. Coming close to a goal suggests minimizing a distance measure (usually a norm of the difference) between an attainable objective vector (decision outcome) and the goal vector.

The basic disadvantage relates to the fact that this idea is mathematically inconsistent with the concept of Pareto-optimality or efficiency. One of the basic requirements – a general sufficient condition for efficiency – for a function to produce a Pareto-optimal or vector-optimal outcome (when minimized or maximized) is an appropriate monotonicity of this function. However, any distance measure is obviously not monotone when its argument crosses zero. Therefore, distance minimization cannot, without additional assumptions, result in Pareto-optimal solutions.

Consider, for example, the simplest case when the goal vector is in itself an attainable decision outcome but not an efficient objective vector; norm minimization then leads to the obvious solution with objectives equal to the goals. Even for convex outcome sets, either special tricks or restrictive assumptions are needed in goal programming to provide for efficiency of obtained decision outcomes. If, however, the set of attainable objectives is not convex (e.g., discrete, as in Figure 2), then norm minimization generally cannot result in efficient outcomes. Both objectives q_1 and q_2 in this figure are to be maximized and the Pareto-optimal outcomes, denoted by circles, are to the “North–East” of the set of attainable objectives. There are many intuitively reasonable vectors of goals, such as $\bar{\boldsymbol{q}}^1$, which would produce inefficient outcomes, such as $\boldsymbol{q}^1$, if a norm as a measure of the distance is minimized.

The problem with setting a goal and trying to come close to it is how to provide for efficiency of resulting outcomes. There are two ways to do this: either to limit the goals or to change the sense of coming close to the goal.

Trying to limit the set of goals is the essence of the displaced ideal method of Zeleny (1974):

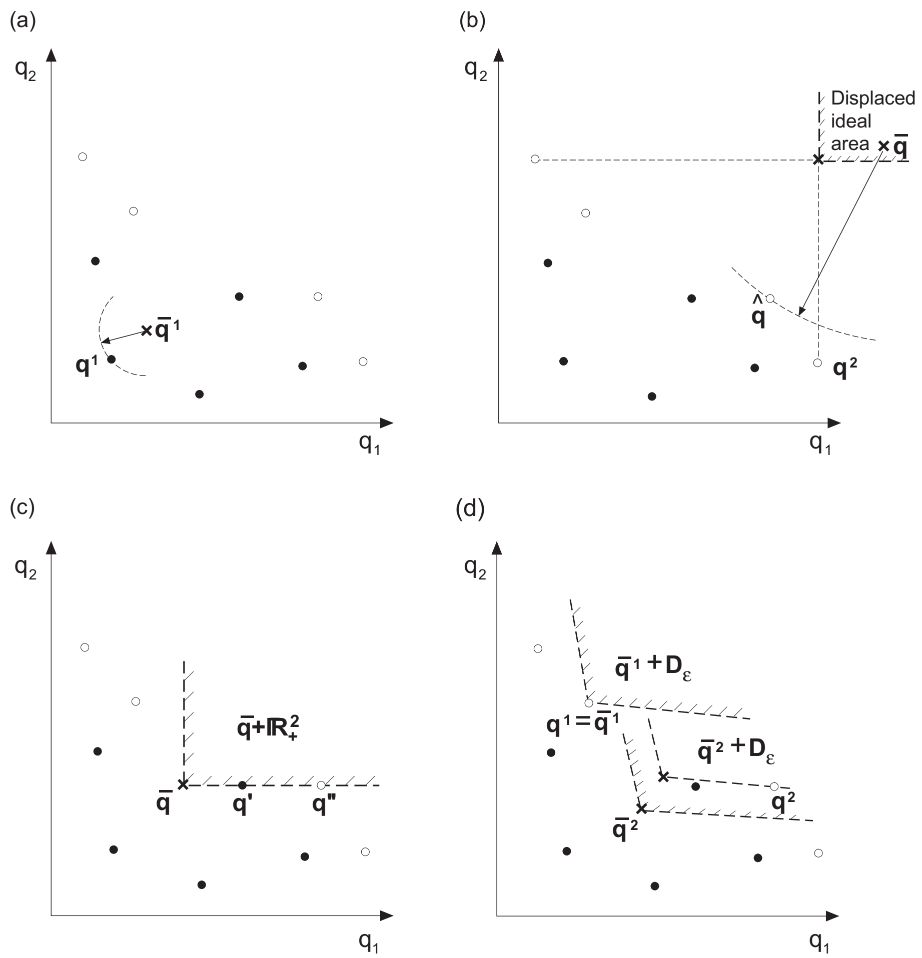

Figure 2: Examples of discrete outcomes using various methods: (a) goal programming or norm minimization; (b) displaced ideal; (c) maxmin method; and (d) reference point method. Circles indicate nondominated points, crosses indicate goals or reference points.

if we select goals that are sufficiently distant from the set of attainable objectives, then we can prove that norm minimization will result only in efficient outcomes, regardless of what norm we use or what properties the set of attainable objectives have. This is illustrated in Figure 2(b), where the goal $\bar{\boldsymbol{q}}$ is in the displaced ideal area and the outcomes resulting from norm minimization are efficient. However, such limitation of admissible goals means losing the intuitive appeal of the goal programming approach: if unrealistic goals have to be set, the approach loses its basic advantages.

Trying to change the sense of coming close to the goal changes the nature of the goal. Reference points are goals interpreted consistently with basic concepts of vector optimality; the sense of "coming close" to it is special and certainly does not mean norm minimization. If we accept the logic of Pareto-optimality, then coming close to a given reference point should mean:

- Objective outcomes are in some sense uniformly close to the given reference point, if the latter is not attainable (while the precise sense of uniform closeness might be modified by demanding that the resulting decisions and their outcomes remain Pareto-optimal).
- Objective outcomes are precisely equal to the given reference point, if the latter is Pareto-optimal, which, somewhat simplified, means attainable without any surplus.
- Objective outcomes are in some sense uniformly better than the given reference point, if the latter is attainable with some surplus. Such reference point is inefficient, not Pareto-optimal (where the sense of its uniform improvement can be variously interpreted).

The first two cases (almost) coincide with goal programming; the third case is, however, essentially different: it means not coming close in any traditional sense, but "coming close or better".

This change of the sense of coming close is deeply related to how people make decisions in reality and how computers should support decision-making. In turn, this is related to the concept of satisficing decisions of Simon (1957), which was used as a description of how people make actual decisions (particularly in large organizations) and the concept of quasi-satisficing decisions of Wierzbicki (1982), which describes how a computerized DSS should help a human decision maker.

According to Simon (1957), real decision makers do not optimize their utility when making decisions, for many reasons. Simon postulated that actual decision makers, through learning, adaptively develop aspiration levels for various important outcomes of their decisions. They then seek decisions that would result in one of two outcomes, that is:

- Objective outcomes as close as possible to the aspiration levels, if the latter are not attainable (which corresponds to an optimization of decisions, but in the sense of the distance from aspiration levels).
- Objective outcomes equal to aspiration levels, if the latter are attainable (which corresponds to stopping improvements).

Satisficing decision making can be mathematically represented by goal programming. In the case of attainable aspiration levels, the decision maker might learn to increase them, but usually not for current decisions, only for future ones. Many studies have shown that such satisficing behavior of a decision maker, though it might seem peculiar, is often observed in practice. In particular, the use of various reference levels by decision makers (such as aspiration levels, but also including reservation levels, of which the latter are very important in the theory of negotiations) has been repeatedly confirmed in practice.

Independent of whether a real, human decision maker would (or could, or should) optimize in all cases, we can require that a good computer program supporting decisions through model analysis should behave like a hypothetical, perfectly rational decision maker. An important exception is that the program should not "outguess" its user, the real decision maker, by trying to construct a model of her/his preferences or utility function, but should instead accept simple instructions that characterize such preferences.

Thus, the methodology of reference point approaches assumes two things: first, that the instructions from a user to the computerized DSS have the convenient (for users) form of reference points, including aspiration levels and possibly reservation levels, and second, that the user is not asked how she/he determines the reference points. An essential departure from Simon's assumptions and from goal programming techniques, however, is the following feature: the methodology of reference point approaches assumes that the computerized DSS tries to improve a given reference point, if this point is attainable.

Therefore, the behavior of the DSS – not that of its user – is similar to perfect rationality. It does not minimize a norm, but optimizes a special achievement function which is a kind of a proxy utility or value function (of the DSS) such that the decisions proposed by the DSS satisfy the three cases of "coming close or better" described previously. Because of the difference – in the last case of "coming better" – with the satisficing behavior, we call such behavior quasi-satisficing. The latter can be compared to the behavior of a perfect staff (either a single staff member or a team of them) that supports a manager or boss, who gives instructions to the staff in the form of reference levels. The staff works out detailed decisions that are guided by the given reference point.

However, being perfect, the staff does not correct attainability estimates (real, human staff might behave otherwise) and does not report to the boss that the reference point is attainable when it really is not. Instead, the staff proposes decisions that result in outcomes as close as possible to the desired reference point and reports these decisions together with their not quite satisfactory outcomes to the boss. If the reference point is attainable without any surplus, the perfect staff works out the decisions needed to reach this point and does not argue that a different point and different decisions might be better. If the reference point is attainable with surplus, the perfect staff does not stop working and start gossiping over drinks – as Simon's model of satisficing behavior would suggest. Instead, the staff works out decisions that would result in a uniform improvement of outcomes as compared to reference levels, and proposes decisions together with improved outcomes to the boss. Obviously, only a computer program could behave in this perfect, quasi-satisficing manner at all times.

However, goal programming corresponds precisely to satisficing behavior: if the aspiration levels are attainable, then there exist outcomes precisely equal to them, thus the corresponding distance is zero. Since we cannot get a distance less than zero, the optimization is stopped (the staff prepares drinks for relaxation).

Thus, reference point optimization is a generalization of the goal programming approach to such cases when we can and want to improve (minimize or maximize) certain outcomes beyond their reference points. For this purpose, a special class of order-consistent achievement functions, sometimes similar but always different from distance functions, was developed, investigated in detail, and applied in many examples and DSSs. Here we shall present only some basic concepts related to them.

3.6.2 Objective Ranges and Objective Aggregation

While the complexity of various types of multi-objective models has been already discussed, here we stress the characteristics of some models, already described in the simple form of equation (2) on page 25. Such a model describes a mapping from the set of admissible decisions X_0 into the set of attainable objectives Q_0. The latter set is not given explicitly – we usually only know how to compute its elements – and the issue is how to analyze the elements and properties of the Pareto ("North–East") frontier of this set. This is usually called a multi-objective analysis of a model.

After specifying any set of variables of a model as objectives q_i, the modeler should first know – at least approximately – the ranges in which these variables might vary. This is also important because we often aggregate objectives, i.e., combine them into one function (not necessarily by summation). Many objectives might also have various units of measurement and must be rescaled to dimension-free units before aggregation. Thus, any system supporting multi-objective optimization and model analysis must include a function for estimating these ranges.

The usual way of such an estimation is to compute a so-called ideal or utopia point by optimizing separately each objective, and to estimate its counterpart – the nadir point, which is composed of worst values of the objectives in the efficient set. A simple way of approximating nadir components (although certainly not the best) is to take the worst values of objective components that occur when we are computing the best values of other components during the calculations of the utopia point. The exact computations of the nadir point might be much more difficult than the computations of the ideal point, see e.g. (Isermann and Steuer, 1987).

In any case, we can assume that the modeler has defined (either arbitrarily or by computing the utopia point and estimating the nadir point) some estimates of ranges of each objective value:

$$q_{i,lo} \le q_i \le q_{i,up} \quad i = 1, \dots, k \tag{3}$$

where $q_{i,up}$ (for maximized objectives, and $q_{i,lo}$ for minimized ones) is at least as high or low as the corresponding utopia point component. The range $q_{i,up} - q_{i,lo}$ is approximately as large as the range utopia–nadir. After specifying such ranges, we can reduce objectives to dimension-free scales (e.g., percentage) and then discuss the relative importance of criteria, their weights, and interpret further concepts such as the trade-off coefficients. Note that without such a reduction (therefore without estimating objective ranges) any specification of the relative importance of objectives is pointless.

For dimension-free objectives or criteria, a classical way to compute an efficient point (e.g., a Pareto point for all objectives maximized) is to maximize a weighted linear scalarizing function:

$$s_1(\boldsymbol{q}, \alpha) = \sum_{i=1}^{k} \alpha_i \frac{q_i - q_{i,lo}}{q_{i,up} - q_{i,lo}}, \;\; \alpha_i \ge 0, \;\; \sum_{i=1}^{k} \alpha_i = 1 \tag{4}$$

where α_i are weighting coefficients that might be interpreted as weights of importance of objectives. If these coefficients are positive, then each maximal point of the above function is Pareto-optimal.

Conversely, if the set Q_0 is convex, for each Pareto-optimal decision $\hat{\boldsymbol{x}}$ there exist weighting coefficients $\hat{\alpha}_i \ge 0, \forall i = 1, \dots k$, such that the maximum of the function $s_1(\boldsymbol{F}(\boldsymbol{x}), \hat{\alpha})$ with respect to $\boldsymbol{x} \in X_0$ is attained at $\hat{\boldsymbol{x}}$.

Thus, it might seem that the weighted linear scalarizing function covers all that is needed for multi-objective optimization. However, there are at least two important reasons to consider such functions as highly inadequate. First, the necessary condition of Pareto-optimality, stated above with the help of the linear scalarizing function, does not account for nonconvex (particularly discrete) sets Q_0. The second reason for the inadequacy of a weighted sum for multi-objective analysis is the fact that maximal points of linear scalarizing functions might depend (in important cases) discontinuously on the weighting coefficients.

For these reasons, it is better to use other functions rather than the weighted sum in order to aggregate various objectives. Historically, and particularly in goal programming approaches (see (Charnes and Cooper, 1977)), much attention has been given to the use of a norm that characterizes a distance of an attainable outcome $\boldsymbol{q} \in Q_0$ from a reference point $\bar{\boldsymbol{q}} \in \mathbb{R}^k$:

$$s_2(\boldsymbol{q}, \bar{\boldsymbol{q}}) = \parallel \boldsymbol{q} - \bar{\boldsymbol{q}} \parallel . \tag{5}$$

We assume that all objectives are maximized, but the distance function should be minimized. Unfortunately, such a distance function is inadequate if we would like to use the reference point as the main parameter to change or control the selection of Pareto-optimal solutions. We would like to admit the use of any reasonable reference point, e.g., at least in the range $\boldsymbol{q}_{lo}$, $\boldsymbol{q}_{up}$, including both $\bar{\boldsymbol{q}} \notin Q_0$ and $\bar{\boldsymbol{q}} \in Q_0$. However, if $\bar{\boldsymbol{q}} \in Q_0$, then the minimal points of such a distance function are very rarely[13] Pareto-optimal.

For a general, nonconvex case, Pareto-optimality of the minimal points of a distance function can be secured only if we assume that the goals are in the displaced ideal set, as indicated in Figure 2(b), see (Zeleny, 1974). However, such unrealistic reference points, very distant from the set of attainable objectives, cannot easily be used as the main parameters for changing or controlling the selection of Pareto-optimal outcomes. Thus, in the displaced ideal and other similar techniques, weighting coefficients were additionally used in a distance function in order to control the selection of its minima. However, the dependence of Pareto-optimal or efficient points on weighting coefficients is not transparent enough. Therefore, the use of reference points as the main parameters controlling the selection of Pareto-optimal points would be much more attractive; reference points usually provide better interpretations for the modeler than weighting coefficients. For this reason, instead of a distance function we must use a different, albeit related, function. This function is called an achievement scalarizing function or an order-consistent achievement scalarizing function which, as opposed to a distance function, preserves the monotonicity even if the point $\boldsymbol{q} = \bar{\boldsymbol{q}}$ is crossed.

3.6.3 Simple Forms of Achievement Functions

We shall discuss here only the simplest forms of achievement functions. Suppose all objectives are maximized and are already rescaled to be dimension-free and aspiration levels $\bar{q}_i$ are used for these objectives. The simplest form of an order-consistent achievement function is then as follows:

$$\sigma(\boldsymbol{q}, \bar{\boldsymbol{q}}) = \min_{1 \le i \le k} (q_i - \bar{q}_i) + \varepsilon \sum_{i=1}^{k} (q_i - \bar{q}_i). \tag{6}$$

where $\varepsilon > 0$ is a small positive parameter. This function, a prototype of all order-consistent achievement scalarizing functions, is monotone with respect to all q_i, thus all its maxima are Pareto-optimal. Conversely, it can be shown, see e.g. (Wierzbicki, 1982), that for each Pareto-optimal decision $\hat{\boldsymbol{x}}$ defined in a specific[14] way there exist reference points[15] $\bar{\boldsymbol{q}}$ such that the maximization of (6) results in finding the selected Pareto-optimal decision $\hat{\boldsymbol{x}}$.

Other order-consistent achievement functions similar to (6) were also used in reference point methodology or other similar approaches to multi-objective optimization, see e.g., (Wierzbicki, 1982; Sawaragi et al., 1985; Steuer, 1986; Wierzbicki, 1992a). We give here an example of a general form of an order-consistent achievement function. Note that the terms $(q_i - \bar{q}_i)$ in (6) can be considered as the simplest example of more general functions $\sigma_i(q_i, \bar{q}_i)$ of the objective variable q_i and the reference or aspiration level $\bar{q}_i$. Such a function can also depend on

[13]Only if $\bar{\boldsymbol{q}}$ happens to be Pareto-optimal.

[14]I.e. with some additional requirements which, however, are reasonable in applications.

[15]Typically, there is an infinite number of such points for each $\hat{\boldsymbol{x}}$.

other parameters, e.g. on a *reservation level* $\bar{\bar{q}}_i$. Examples of such functions are shown in Figure 3 on page 37. Function σ_i must have special monotonicity properties and is often defined as a piece-wise linear function; this function is called the partial (often also the component) achievement function. By substituting the terms $(q_i - \bar{q}_i)$ in (6) by component achievement functions $\sigma_i(q_i, \bar{q}_i)$, we obtain the following more general form of the order-consistent achievement function:[16]

$$\sigma(\boldsymbol{q}, \bar{\boldsymbol{q}}) = \min_{1 \le i \le k} \sigma_i(q_i, \bar{q}_i) + \varepsilon \sum_{i=1}^{k} \sigma_i(q_i, \bar{q}_i). \tag{7}$$

The achievement function $\sigma(\boldsymbol{q}, \bar{\boldsymbol{q}})$ – in either form presented above, as well as other similar forms – is nondifferentiable. In the case of linear models, the nondifferentiability of the achievement function is unimportant, because in practice the component achievement functions $\sigma_i(q_i, \bar{q}_i)$ are concave, strictly monotone, piece-wise linear functions (increasing for maximized, decreasing for minimized criteria, respectively), and the maximization of the order-consistent achievement function (7) can be equivalently expressed as a linear programming problem by introducing additional constraints and auxiliary variables.[17] However, in the case of nonlinear models, optimization algorithms for smooth functions are more robust (i.e., work more reliably without the necessity of adjusting their specific parameters to obtain results) than algorithms for nonsmooth functions. Therefore, there are two approaches to the maximization of such achievement functions. The first approach is to introduce additional constraints and auxiliary variables similarly as for linear models. The second one uses a smooth approximation of the achievement function. The choice depends on the characteristics of the underlying non-linear model. Typically, for models with a large linear part, the first approach is more efficient, while for models with a dominating non-linear part the second one is likely to be better.

The reference point approach is related not only to the method of displaced ideal or goal programming, but also to many other techniques of multi-objective optimization. A popular technique (Polak, 1976) is to maximize the minimal of maximized objectives or of their increases above postulated reservation levels. Its use was indicated in Figure 2(c). It corresponds to the maximization of the achievement function (6) with $\varepsilon = 0$ which, however, has some disadvantages. This minmax method guarantees only weak Pareto-optimality. In Figure 2(c), weakly Pareto-optimal is the point $\bar{\boldsymbol{q}}'$, which is clearly dominated by the Pareto-optimal point $\bar{\boldsymbol{q}}''$. Note that the use of the achievement function (6) with $\varepsilon > 0$, as indicated in Figure 2(d), does not produce weakly Pareto-optimal points that could be improved in one component of the objective vector.

Finally, we briefly consider the concept of a neutral or compromise solution. Zeleny (1974) introduced the concept of a compromise (Pareto-optimal) solution, where its objective outcome is situated "somewhere in the middle" of the Pareto set in objective space. The precise meaning of somewhere in the middle is specified either by defining the displaced ideal point or by using various weighting coefficients. When using the basic achievement function (6) and assuming that all objectives are already rescaled to be dimension-free, we might, however, define a neutral compromise solution more precisely: it corresponds to the result of maximizing function (6) while putting the reference point precisely at the ideal point and using weighting coefficients equal to each other (i.e., not using them at all, as in the function (6)) for the dimension-free objectives with equal ranges. Such a neutral compromise solution is a good starting point in an interactive process of analyzing a multi-objective model.

[16] Sometimes – if the value 1 of a component achievement function has any importance, e.g. indicating that the aspiration level is reached while the value 0 corresponds to reservation level – the entire value of such a function $\sigma(\cdot)$ might be divided by $(1 + k\varepsilon)$, in order to give it the value 1 if all component functions have value 1.

[17] See Section 3.8 for details.

When using the reference point methodology, the process of analyzing a multi-objective model is simply constructed: first we let the decision maker or the modeler play freely with reference points and ask the DSS to respond to each reference point by maximizing the achievement function, e.g., of the form (6). This produces the Pareto-optimal/efficient decisions and their objective outcomes associated with this reference point. By doing this, the decision maker or modeler gains insight into the behavior of the multi-objective, substantive model of the decision situation. This initial phase might be supplemented, if necessary, with various other techniques of using reference points when supporting choice. Experience in multi-objective model analysis shows that the initial phase of learning by freely changing reference points might be most important, particularly for supporting the development of intuition of the decision maker or the modeler.

3.7 Sensitivity Analysis

In mathematical programming sensitivity analysis is typically understood as analysis of changes of an optimal solution caused by change of the data in the model.[18]

For linear programming models, a traditional approach of such analysis is based on properties of an optimal solution. It typically consists of calculations of ranges of changes of parameters for which an optimal solution does not change, and on using a dual solution for calculations of changes of value of a goal function for changes of some parameters that are small enough for allowing such a simple evaluation procedure. These methodological topics, which all form the subject of post-optimal analysis, and the corresponding software tools have been extensively developed. However, their applicability is practically limited to rather small, linear models.

Moreover, there are several problems with applications of these approaches to real problems. We restrict the discussion to the three basic issues:

- The concept and tools for sensitivity analysis have been developed and implemented for analysis of rather small models. As the complexity of models applied in decision support grows, it is either cumbersome or practically impossible to use these tools.
- For mixed-integer and non-linear types of models, there are even less developed tools for post-optimal sensitivity analysis.
- In many models the quality of dual solution is rather questionable, often the dual solution is practically non-unique. This is due to the fact that most of large models are numerically badly conditioned. Moreover, one uses for them efficient presolve algorithms, which greatly decrease the resources (time and memory) needed for solving large problems. However, these algorithms guarantee the quality of the primal solution but often result in unreliable dual solution, which is the basis for classical sensitivity analysis. Therefore, such an analysis requires a good understanding of various techniques and corresponding tools, which limits its applications to highly skilled specialists in mathematical programming.

These reservations concern sensitivity analysis in classical, scalar optimization (single-objective mathematical programming). In vector optimization and multiobjective decision support, basic concepts of sensitivity analysis must be changed. Mathematically, it is possible to analyze the sensitivity of the entire Pareto set, but it would require solutions of many more parametric optimization problems than is needed by more restricted single-criterion optimization approaches. Another solution would be to apply in a multiobjective analysis a generalized

[18]There exist also more general approaches to sensitivity analysis, concerned with the consequences of using conclusions such as an optimal solution or even an optimal feedback law derived from one model and applied to another model (e.g. with changed parameters), see (Wierzbicki, 1984), but the necessary tools for such generalized sensitivity analysis are not further developed yet.

approach to sensitivity analysis as in (Wierzbicki, 1984). However, even in such cases experiments with changed parameters for more complicated problems can grow very fast in volume.

On the other hand, vector optimization and multiobjective decision support offer better ways for providing some type of generalized sensitivity analysis. In classical single-criterion optimization only one criterion can be selected as a goal function, therefore typically several objectives are treated as constraints, for which one had to specify an acceptable value. However, in practice such values cannot be specified precisely, therefore their modifications are inevitable. This in turn requires sensitivity analysis of the impact of changes caused by specified constraining values for criteria that are treated as constraints. One needs to note that sensitivity analysis is typically based on dual solutions, which are often (especially for large and/or badly conditioned problems) not robust. Moreover, results of such an analysis are valid only in a (typically small) neighborhood of an optimal solution. These two facts commonly known for practitioners who deal with complex models explain why sensitivity analysis of optimal solutions provide very limited support for analysis of multicriteria problems that are analyzed through classical single-criterion optimization methods. These limitations and the demand for a truly multicriteria model analysis have been in fact one of the driving forces for the development of methods and tools for multiobjective model analysis.

3.8 Softly Constrained Inverse Simulation

An important help for the modeler can be the *inverse simulation*, in which she/he assumes some desired model outcomes $\bar{\boldsymbol{y}}$ and checks – as in the classical goal programming – whether there exist admissible decisions which result in these outcomes. *Generalized inverse simulation* consists in specifying also some reference decision $\bar{\boldsymbol{x}}$, and in testing, whether this reference decision is feasible and results in the desired outcomes $\bar{\boldsymbol{y}}$. This can be written in the goal programming format of the minimization of a norm, for which it is useful to apply the augmented Chebyshev norm. However, because norms are minimized, and we want to keep to the convention that the achievements are usually maximized, we use the following form of the achievement function:

$$\begin{aligned}\sigma(\boldsymbol{y},\bar{\boldsymbol{y}},\boldsymbol{x},\bar{\boldsymbol{x}}) &= -(1-\rho)(\max_{1\le i\le n} |\, x_i - \bar{x}_i \,| + \varepsilon \sum_{i=1}^{n} |\, x_i - \bar{x}_i \,|) \\ &\quad -\rho(\max_{1\le j\le m} |\, y_j - \bar{y}_j \,| + \varepsilon \sum_{j=1}^{m} |\, y_j - \bar{y}_j \,|) \qquad (8)\end{aligned}$$

The coefficient $\rho \in [0;1]$ indicates the weight given to achieving the desired output versus keeping decisions close to their reference values. It is assumed for simplicity sake that all variables are already re-scaled to be dimension-free.

A multi-objective optimization system based on reference point methodology can clearly be used also for inverse simulations. In such cases, we stabilize all outcomes and decisions of interest, and use for them partial achievement functions of the form $\sigma_i(y_i, \bar{y}_i)$ (or even $\sigma_i(y_i, \bar{y}_i, \bar{\bar{y}}_i)$), similar to those discussed in the Section 3.6.3 in terms of objectives q_i. Then an overall achievement function takes the form:

$$\begin{aligned}\sigma(\boldsymbol{y},\bar{\boldsymbol{y}},\boldsymbol{x},\bar{\boldsymbol{x}}) &= (1-\rho)(\min_{1\le i\le n} \sigma_i(x_i,\bar{x}_i) + \varepsilon \sum_{i=1}^{n} \sigma_i(x_i,\bar{x}_i)) \\ &\quad +\rho(\min_{1\le j\le m} \sigma_j(y_j,\bar{y}_j) + \varepsilon \sum_{j=1}^{m} \sigma_j(y_j,\bar{y}_j)) \qquad (9)\end{aligned}$$

It is more convenient for the modeler, if such functions are defined inside the decision support system which also has a special function *inverse simulation,* prompting her/him to define which (if not all) decisions and model outputs should be stabilized and at which reference levels.

Even more important for the modeler might be another generalization of the above function, called *simulation with elastic constraints* or *softly constrained inverse simulation*. Common sense decisions might appear inadmissible for the model, because it interprets all constraints as *hard* mathematical inequalities or equations. On the other hand, we have already stressed that it is a good modeling practice to distinguish between *hard constraints* that can never be violated and *soft constraints* which in fact represent some desired relations and are better represented as additional objectives with given aspiration levels. Thus, in order to check actual admissibility of some common-sense decision $\bar{\boldsymbol{x}}$, the modeler should first answer the question which constraints in her/his model are actually hard and which might be softened and included in the objective vector $\boldsymbol{q}$. Thereafter, inverse simulation with soft constraints might be performed by maximizing an overall achievement function similar as above, but defined with respect to objectives q_i:

$$\begin{aligned} \sigma(\boldsymbol{q}, \bar{\boldsymbol{q}}, \boldsymbol{x}, \bar{\boldsymbol{x}}) &= (1-\rho)(\min_{1 \le i \le n} \sigma_i(x_i, \bar{x}_i) + \varepsilon \sum_{i=1}^{n} \sigma_i(x_i, \bar{x}_i)) \\ & \quad + \rho(\min_{1 \le j \le m} \sigma_j(q_j, \bar{q}_j) + \varepsilon \sum_{j=1}^{m} \sigma_j(q_j, \bar{q}_j)) \end{aligned} \tag{10}$$

where we do not assume that all objectives are stabilized, but *include also maximized or minimized objectives e.g.* corresponding to inequalities in soft constraints. Again, it is better if the multi-objective optimization system has a special function called *elastic simulation*, which invokes the specific form of the achievement function and prompts the modeler to specify necessary data – in particular, to indicate constraints that might be softened.

We should stress here that, if either (9) or (10) is maximized with concave piece-wise linear partial achievement functions σ_i and for a linear model, then the underlying optimization problem can be converted to linear programming. In fact, if a partial achievement function – say, $\sigma_i(x_i, \bar{x}_i)$ – is piece-wise linear and concave, then it can be expressed as the minimum of a number of linear functions:

$$\sigma_i(x_i, \bar{x}_i) = \min_{l \in L_i} \sigma_{il}(x_i, \bar{x}_i) \tag{11}$$

where $\sigma_{il}(x_i, \bar{x}_i)$ are linear functions. Assume that a similar expression is valid for $\sigma_j(q_j, \bar{q}_j)$. The maximization of the function (10) can be then equivalently expressed as the maximization of the following function of auxiliary variables z, z_i, w, w_j:

$$(1-\rho)(z + \varepsilon \sum_{i=1}^{n} z_i) + \rho(w + \varepsilon \sum_{j=1}^{m} w_j) \tag{12}$$

with additional constraints:

$$\begin{aligned} \sigma_{il}(x_i, \bar{x}_i) &\ge z_i, \ \forall l \in L_i \\ z_i &\ge z, \ \forall i = 1, \ldots n \\ \sigma_{jl}(q_j, \bar{q}_j) &\ge w_j, \ \forall l \in L_j \\ w_j &\ge w, \ \forall j = 1, \ldots m \end{aligned} \tag{13}$$

Typically, component achievement functions $\sigma_i(\cdot)$ are concave, and therefore the maximization of an achievement function can be performed by the maximization of an auxiliary linear function with several additional (to the underlying model) linear constraints. Therefore the resulting optimization problem has similar characteristics as an optimization problem composed of a linear goal function and the constraints defined by the model. For linear models defined by only continuous variables this is a recommended approach.

Models with integer variables are typically much more difficult to be optimized than corresponding relaxed (where integer variables are replaced by continuous variables) models. For models with integer variables one can be tempted to consider non-concave piece-wise linear component achievement functions $\sigma_i(\cdot)$ because such functions can be represented by conversions that use auxiliary binary and continuous variables, which results in adding a small mixed-integer linear part to the underlying model that for medium size models does not increase remarkably the size of the original problem. However, the resulting optimization problem may be much more difficult to be solved than a problem resulting from concave component achievement functions. This is more likely for problems for which heuristic algorithms have to be applied for optimization. Therefore it is recommended to use concave piece-wise linear component achievement functions $\sigma_i(\cdot)$, which in practice provides a sufficiently good representation of user preferences. This approach does not remarkably increase the complexity of the resulting optimization problem. This recommendation is even much more stronger for non-linear models, because an introduction of integer variables to such models dramatically increases the complexity of the resulting optimization problem. However, as already discussed in Section 3.6.3, for some non-linear models smooth approximations of concave piece-wise linear functions $\sigma_i(\cdot)$ can be a more efficient approach that the use of original functions.

3.9 Relations between Fuzzy Sets and Reference Point Methods

Theoretical background and a number of applications of fuzzy sets to decision making are discussed e.g. by Zimmermann (1987) and Sakawa (1993). Both the fuzzy membership functions (discussed in detail e.g. by Zimmermann (1978) and Zimmermann (1985)), and the component achievement functions discussed in this chapter reflect the degree of satisfaction with a given set of criteria values. Therefore, it is interesting to consider some similarities and differences in applications of the Reference Point Optimization and Fuzzy Sets methods to model-based decision-making support.

Both approaches (fuzzy sets and reference point) assume, that for each criterion the user selects two values (called here the *aspiration* and the *reservation* level, respectively). The user considers solutions for which the criterion value is better than the aspiration level as good, and solutions for which this value is worse than the reservation level as bad. A goodness of solutions for which a criterion value is between the aspiration and reservation values depends on a measure of distance between the criterion value, and the aspiration and reservation levels.

In the fuzzy sets approach a value of a membership function (denoted[19] here by $\mu(q)$) is interpreted as a degree to which the corresponding solution belongs to a set of *good* solutions. For maximized criteria $\mu(q)$ is a non-decreasing function conforming to the following requirements:

$$
\begin{array}{lclll}
\mu(q) & = & 0 & \text{for} & q \leq R \\
0 & < \mu(q) < & 1 & \text{for} & R < q < A \\
\mu(q) & = & 1 & \text{for} & q \geq A
\end{array}
\tag{14}
$$

where R and A denote the reservation and aspiration values, respectively. An example of such a function is shown in Figure 3 (a) by the dashed line (which for the segment corresponding

[19] For a simplification of the presentation, criteria indices in membership functions $\mu_i(\cdot)$, and in component achievement functions $\sigma_i(\cdot)$ are omitted in this sub-section.

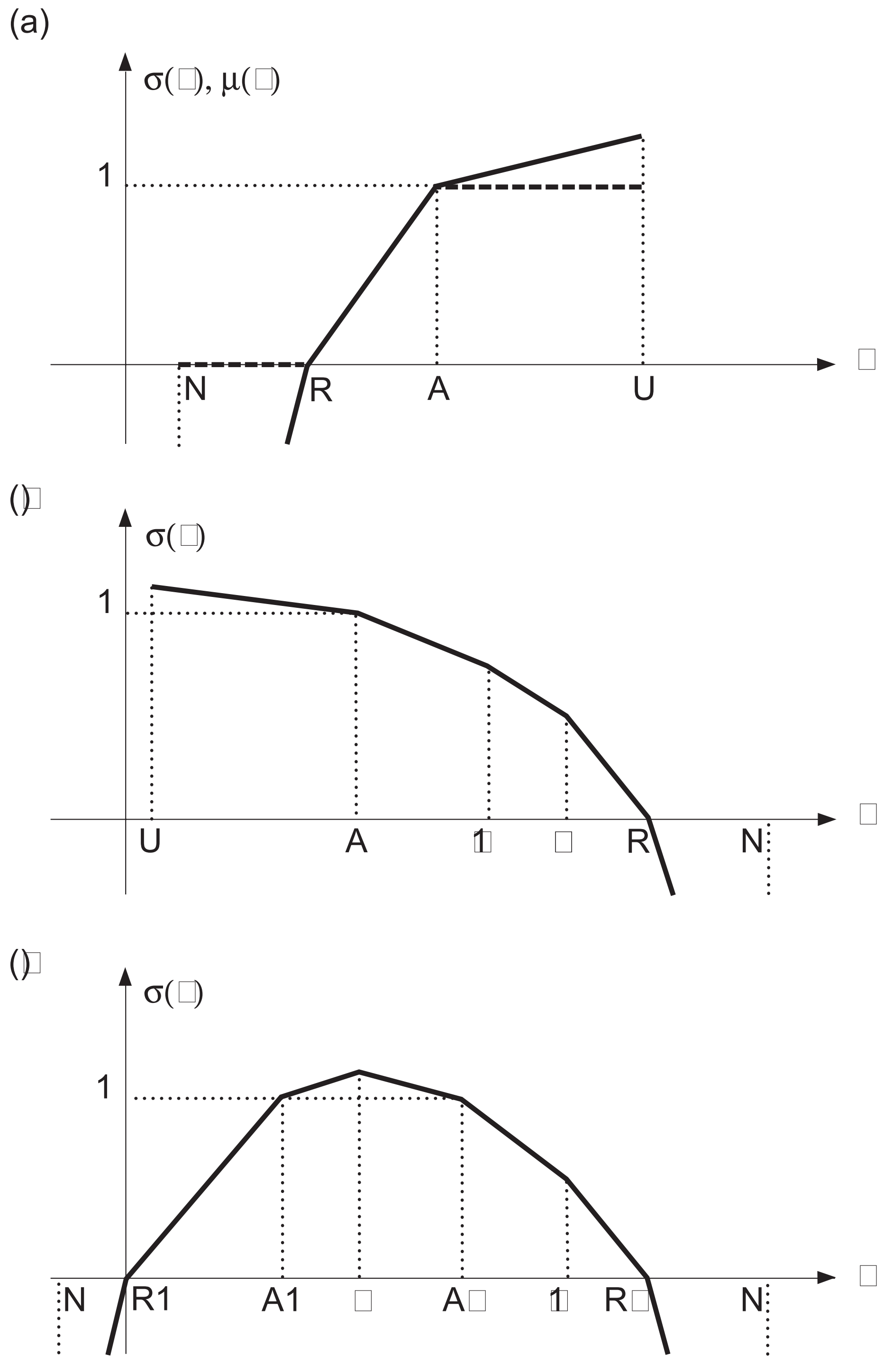

Figure 3: Examples of component achievement functions $\sigma(q)$ for three types of criteria: (a) maximized (together with the corresponding membership function $\mu(q)$ shown by the dashed line), (b) minimized, (c) goal (stabilized), respectively. The letters N, R, A, U, and T denote values of nadir, reservation, aspiration, utopia, and target, respectively.

to values $R \leq q \leq A$ is covered by the solid line). The best and worst criterion values are called *Utopia* and *Nadir*, respectively, and are typically computed during the initial phase of the problem analysis. These values are denoted in Figure 3 (a) by letters U and N, respectively.

The reference point approach described in this chapter uses a so called extended-valued membership function as component achievement functions $\sigma(\cdot)$. Such an extension of the membership function concept has been proposed by Granat and Wierzbicki (1994), who also suggested a method of constructing various forms of order-consistent component achievement scalarizing functions based on membership functions describing the satisfaction of the user with the attainment of separate objectives. For a maximized criterion $\sigma(q)$ is a strictly increasing function conforming to the following requirements:

$$\begin{array}{lcllll} \sigma(q) & < & 0 & \text{for} & q < R & \\ \sigma(q) & = & 0 & \text{for} & q = R & \\ 0 & < \sigma(q) < & 1 & \text{for} & R < q < A & \\ \sigma(q) & = & 1 & \text{for} & q = A & \\ \sigma(q) & > & 1 & \text{for} & q > A & \end{array} \tag{15}$$

An example of the most simple (defined by only four points: utopia, aspiration, reservation, and nadir) PWL (piece-wise linear) function $\sigma(\cdot)$ for a maximized criterion is shown in Figure 3 (a) by the solid line.[20]

An example of a PWL function $\sigma(\cdot)$ for a minimized criterion is shown in Figure 3 (b). For this example the user has specified (in addition to the aspiration and reservation levels denoted by A, and R, respectively) values of his/her degree of satisfaction for two more values of criteria, denoted by q1, and q2, respectively. Note that the concavity requirement corresponds well to the nature of the problem since one accepts small changes of $\sigma(\cdot)$ when a criterion value is better or close to an aspiration level. The speed of such a change should increase along with moving away from an aspiration level towards a reservation level, and should be greatest for the criteria values worse than the corresponding reservation level. Such features of $\sigma(\cdot)$ are justified not only by a theoretical reasoning but also by experience from real-world applications, which shows that users are primarily interested in improving those criteria whose values are close to the corresponding reservation levels (especially, if the values are much worse than these levels).

In some (e.g. engineering) applications a criterion should neither be maximized nor minimized, but it is desired that its value is possibly close to a *target*. For such cases a *goal type* (also called *stabilized*) criterion is suitable. An example of a component achievement function $\sigma(\cdot)$ for such criterion is shown in Figure 3 (c). The goal type criterion requires a specification of a target value (denoted by T), and of two pairs of aspiration and reservation levels (for the criterion values smaller and greater than the target value, respectively), which are denoted in Figure 3 (c) by pairs (A1, R1) and (A2, R2), respectively. For the goal-type criterion a component achievement function is composed of two parts, defined respectively for the criterion values smaller and larger than the given target. Obviously, the first (second) part has properties of $\sigma(\cdot)$ for maximized (minimized) criteria, respectively. Optionally, the user can specify a level of his/her satisfaction with the criterion values between aspiration and reservation levels, as illustrated by the value q1 in Figure 3 (c).

The minimized, maximized, and goal types of criteria are the most commonly known and used. However, the Reference Point Optimization method has been successfully applied also for other types of criteria. Several of them are discussed by Makowski (2001). Moreover, we would like to point out that modular software tools are available for easy implementations of

[20]For practical reasons (the value of $\sigma(N)$ is typically much smaller than -10, see e.g. (Makowski, 2001) for details) only a small part of the segment for criteria values smaller than the reservation level is shown.

user-friendly interfaces to the Reference Point Optimization method (including an interactive definition of criteria, components achievement functions $\sigma(\cdot)$, comparisons of solutions, etc.), see e.g. (Makowski and Granat, 2000) for details.

The above summary shows that there are many similarities, and many differences between the Reference Point and the *Fuzzy Multi-objective Programming* approaches. The main similarity is in the interpretation of values of the membership function $\mu(\cdot)$ and the component achievement function $\sigma(\cdot)$ for the criteria values between the aspiration and reservation levels: both functions reflect a goodness of the corresponding solution in respect to the corresponding criterion. There are however two types of differences between these two approaches.

First, functions $\sigma(\cdot)$ for minimized and maximized criteria are strictly monotone in their whole domains, while membership functions $\mu(\cdot)$ are constant for the criterion values better than the aspiration level, or worse than the reservation level. Thus the maximization of functions $\mu(\cdot)$ does not force improvements of criteria, if their reservation and/or aspiration levels were selected too optimistically and/or too pessimistically, respectively.

The second difference is due to the specification and use of the $\mu(\cdot)$ and $\sigma(\cdot)$ functions.

The fuzzy set approaches assume that the membership functions are elicited before the interactive analysis of the problem. The interactive fuzzy multi-objective programming as proposed in (Seo and Sakawa, 1988; Sakawa, 1993) uses given membership functions (one function for each criterion) for the interactive procedure in which the user specifies the aspiration levels of achievement of the membership values for all of the membership functions, called the reference membership level. Hence, this method requires prior specification of aspiration and reservation levels which are used for the definitions of the membership function. Moreover, it is implicitly assumed that the criteria values for all interesting solutions are between the corresponding aspiration and reservation levels (because the applied membership function does not differentiate between solutions with values better than the aspiration level and between those with values worse than the reservation level). The user interactively specifies the reference membership levels for each membership function, which can be interpreted as a degree of achievements of the aspiration for each criterion (scaled by the difference between aspiration and reservation). Therefore, in this approach the user cannot change aspiration levels in terms of criteria values, because they have to be specified a priori for the definition of the membership function.

The reference point method does not use the membership function directly. It assumes that the user may change aspiration and reservation levels (and optionally specify her/his preferences for selected values of criteria between the aspiration and reservation levels) during the interaction upon the analysis of previously obtained solutions. The user interactively specifies the preferences in the space of the criteria values which seems to be more natural than a specification of preferences in terms of degrees of achievements of membership function values. A selection in the criteria space can, however, be interpreted in terms of Fuzzy Sets by a definition of a membership function for a linguistic variable (e.g. *good solution*) for each criterion, and an ex post interpretation to which degree a solution belongs to a set of *good* solutions. Moreover, there is no need for restrictions for the specification of aspiration and reservation levels in the criteria space. This is important for the analysis of large-scale complex problems for which it might be difficult to specify attainable/non-attainable reservation/aspiration levels, respectively.

Namely, the membership functions $\mu(\cdot)$ have to be elicited before an initial iteration, while the component achievement functions $\sigma(\cdot)$ are interactively changed by the user upon analysis of obtained solutions, which allows for progressive changes of the achievement functions during a learning process.

4 Decision Support Systems

Given the complexity of the DMP, and of multi-paradigm modeling, discussed in Sections 2 and 3, respectively, an appropriate computerized support for decision-making related to any complex problem can only be provided by a customized computing environment, which is conventionally called Decision Support System (DSS).

The concept of DSS was first used in the late 1970s, see e.g. (Tukey, 1977), in opposition to data processing systems developed in 1950s. A DSS was supposed to have more functions than Data Base Management Systems (DBMS), or Management Information Systems (MIS), see e.g. (Codd, 1970) and (Emery, 1987), respectively. One of the early definitions of DSS given by Sprague (1983) stresses that a DSS is an interactive computer-based system designed to help DMs use data and models to find solutions to unstructured[21] problems. While the classical definition of DSS can still be considered valid, the meaning of "use data and models" has changed substantially along with the change of context of modern decision-making discussed in Section 2. Thus also the DSS functionality and architecture has changed considerably in comparison to DSSs developed earlier than even a decade ago. These developments have been possible due to fast changes in hardware and software technologies, which have made it possible to address the needs of DSS users in a far better way than previously.

Models developed for decision-making support represent this part of knowledge relevant to a decision making process, which can be used in a more efficient way in a computerized form than provided by other means. The meaning of this statement implies that the models, and the way they are used cannot be defined a priori. The knowledge useful for model-based representation is typically heterogeneous, therefore it is often initially represented by several models developed and tested separately before they are combined into one model, or in several larger models that are used as a system of models. In fact most of successful applications show that the requirements for model-based support change along a (typically long) process of development and application of a model. In some cases the modeling process (which starts with structuring the modeled problem) is even more important than the final version of a model, and the end-product of this process, which is a DSS into which the developed model is included. Moreover, support of different phases of DMP (discussed in Section 2) may require different models.

A modeling process is typically composed of the following interlinked elements:
- model (symbolic) specification,
- collection and verification of data to be used for model instantiation,
- model verification and testing,
- generation of various instances of the model,
- comparative analysis of each of these instances.

The modeling process is in fact also a learning process for both the model developers and future users. During this adaptive process the assumptions and requirements for the model are modified until a best possible fit between possibilities offered by the modeling technology, available knowledge, and data on one side, and the needs of the users on the other side is achieved. A more detailed discussion of these elements of the modeling process (related to the RAINS model outlined in Section 3.2 but applicable also to many other complex models) can be found in (Makowski, 2000). Different methods and tools are used for different elements of modeling process, which typically requires a substantial amount of resources. Therefore it is typically rational to reuse software that was developed (and tested) for other applications. However, for making reusing software components practicable, the modeling environment has

[21]Unstructured in this context means that a problem cannot be adequately represented by a mathematical programming problem, whose unique solution provides a satisfactory advice for making a decision.

to be composed of modular tools that can be configured according to various needs.

In order to illustrate the advantages of a modular structure of a DSS, we outline here the architecture of two DSSs applied in two different areas, namely regional water quality management, and land use planning. Each of these applications was developed for simpler (than that for which the the RAINS model was developed) problems. However, they illustrate well a modular structure of a DSS necessary when using various modeling paradigms. Moreover, they also show how easily one can use in practice various modular modeling tools to develop a DSS which supports the reference point methodology to a model analysis.

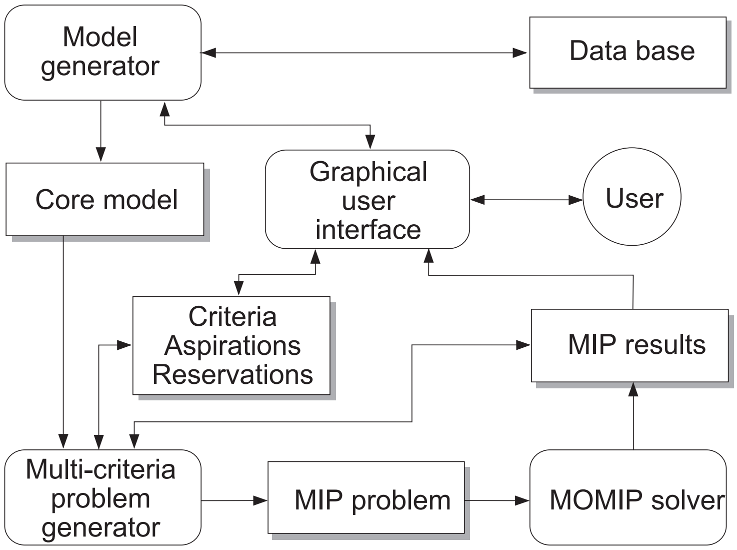

Figure 4: The structure of a Decision Support System for the water quality management in the Nitra River River basin.

The structure of the DSS used for the water quality management in the Nitra Basin, see Makowski et al. (1996) for details, is illustrated in Figure 4. The implementation of this DSS is based on modular and portable software tools that support the following functions of a DSS:

- A problem-specific model generator for generating the core model that relates wastewater emissions, treatment decisions, and the resulting ambient water quality. It is important to stress again that the core model includes only physical and logical relations between the decision and output variables, and not the preferential structure of the DM.
- A graphical user interface (GUI) for handling all the interaction with the user, who uses the GUI to first select an instance of the model he/she wants to analyze, and then specifies during an interactive process his/her preferences. The latter is done by selecting outcome variables that are used as criteria. Then for each selected criterion the user defines (through a specialized graphical interface) a component achievement function $\sigma_i(\cdot)$, which has properties discussed in Section 3.6.
- The GUI is linked with the MCMA package, which handles the definitions of criteria, aspiration, and reservation levels, as well as the generation of a multi-criteria problem. The MCMA includes ISAAP (Interactive Specification and Analysis of Aspiration-based Preferences by Granat and Makowski (2000), which is directly linked with LP-MULTI by (Makowski, 1994b), which in turn is a modular tool for handling multi-criteria problems using the reference point methodology presented in more detail in Section 3.6. The resulting parametric mixed-integer programming (MIP) problem is based on the core (substantive) model and on an interactively specified preferential structure of a DM.

- A modular solver for mixed integer programming problems, MOMIP, developed by Ogryczak and Zorychta (1996).
- A data interchange tool LP-DIT by Makowski (1994a). This tool provides an easy and efficient way for defining and modifying MIP problems, as well as interchanging of data between a problem generator, a solver, and software modules that serve for problem modification and solution analysis.

The structure of the DSS for land-use planning described by Antoine, Fischer and Makowski (1997) is the same as that presented above. Only one substantial module, namely, another problem-specific model generator for generating the land-use core model, had to be developed. Another difference is of a technical nature. The resulting optimization model is of the LP type, therefore another solver is automatically selected by the DSS instead of the MOMIP solver.

In addition to problem-specific model generators (and associated utilities for data management), each of these two DSSs has problem-specific utilities for analysis of the results, but the kernel of the software for both DSSs is the same. This illustrates well the power of modular modeling tools, and the associated efficiency resulting from the structured design of software.

A more detailed discussion of the DSS architecture is beyond the scope of this chapter. However, the relations between the decision-making process and the DSS architecture, its basic components, and implementation issues, all illustrated by a modular DSS actually implemented for environmental case studies, are presented in more detail by four case studies described in (Makowski and Wierzbicki, 2000). The fast developments in various areas that contribute to the decision support methods and technology call for new approaches to knowledge management and representation by analytical models, and its use for decision-making support by utilization of also fast developments of global information networks. The implications of these developments on future DSS are discussed in (Makowski and Wierzbicki, 2003).

5 Conclusions

This chapter focused on the use of models of knowledge in decision support. All these concepts – of knowledge, models, decisions and support – can be used in very general but also in a very specific sense. We have discussed some more specific interpretations of these concepts that resulted from our experience in diverse applications of decision support.

Because of the variety of decision problems and of habitual domains of DMs, there will probably never be just one method of model-based decision support. In fact, no single modeling paradigm alone is sufficiently good enough to identify and analyze various enough policy options for any complex decision problems that are necessary for making rational decisions. Rather, an integration of various modeling methods and tools is needed to provide the best available support possible to analyze complex problems.

It is worth to note that a number of diverse modeling paradigms has been developed to a large extent in parallel by researchers who tried their best to find better (than those offered by traditional mathematical programming) methods for solving various real-world problems. For example, the fuzzy set approaches were motivated by problems that could be adequately represented neither by traditional (crisp) models nor by stochastic models. Similarly, multicriteria model analysis methods were developed in response to limitations of traditional single-criterion optimization approaches. For a quite long time these two streams have been nurtured by research communities that were to a large extent separated. In this chapter we have shown that in fact some concepts in both approaches are rather similar (although named differently), and that both approaches belong to the class of methods that are now described as *soft computing.* Both of them have contributed greatly to model-based decision support, because *support* implies that only *soft computing* (as opposed to approaches based on providing an optimal solution of a precisely defined mathematical programming problem) can be helpful in a better understanding of a complex problem, which is a prerequisite for finding a satisfactory solution to it.

Lessons learned from the applications of various modeling paradigms to diverse types of real-world problems, and the recent abundance of computing hardware and software tools makes it possible to integrate several methods of a model specification and analysis, and to apply them to large and complex problems. Such an integration calls for a collaboration of specialists, who have concentrated their efforts – and therefore have substantial experience – in a particular method. Therefore, one should expect that diverse integrations of various modeling paradigms will be used more broadly to improve decision-making support in a wide range of practical problems.

However, the key role in actual decision making will stay with human decision makers. Thus, we have also stressed the role of human intuition – together with a rational theory of intuition that makes it possible to understand how to enhance intuition by computerized decision support. We have also stressed the interactive, learning aspects of decision support processes. All these aspects can be summarized by one concluding postulate of *user sovereignty*. This postulate includes the postulate of *user friendliness* of information systems, but is further reaching, and describes the dominant role of a human decision maker in human-computer interactions.

References

Ackoff, R.: 1967, Management misinformation systems, *Management Science* **14**(4), 43–89.

Ackoff, R.: 1979, The future of operational research is past, *Journal of OR Society* **30**(2), 93–104.

Amann, A. and Makowski, M.: 2000, Effect-focused air quality management, *in* Wierzbicki et al. (2000), pp. 367–398. ISBN 0-7923-6327-2.

Antoine, J., Fischer, G. and Makowski, M.: 1997, Multiple criteria land use analysis, *Applied Mathematics and Computation* **83**(2–3), 195–215. available also as IIASA's RR-98-05.

Axelrod, R.: 1984, *The Evolution of Cooperation*, Basic Books, New York.

Bergson, H.: 1903, Introduction à la métaphysique, *Revue de la métaphysique et de morale* **11**, 1–36. Translated by T.E. Hulme as *Introduction to Metaphysics*, New York, 1913 and 1949.

Bertalanffy, L.: 1968, *General Systems Theory: Foundations, Development, Applications*, Braziller, New York.

Bosc, P. and Kacprzyk, J.: 1995, *Fuziness in Database Management Systems*, Springer Verlag, Berlin, New York.

Carlsson, C. and Fullér, R.: 2002, *Fuzzy Reasoning in Decision Making and Optimization*, Physica Verlag, New York.

Cartwright, N.: 1999, *The Dappled World. A Study of the Boundaries of Science*, Cambridge University Press, Cambridge.

Chapman, C.: 1988, Science, engineering and economics: OR at the interface, *Journal of Operational Research Society*.

Charnes, A. and Cooper, W.: 1967, *Management Models and Industrial Applications of Linear Programming*, J. Wiley & Sons, New York, London.

Charnes, A. and Cooper, W.: 1977, Goal programming and multiple objective optimization, *J. Oper. Res. Soc.* **1**, 39–54.

Codd, E.: 1970, A relational model for large shared data banks, *Comm. ACM* **13**(6), 377–387.

Dreyfus, H. and Dreyfus, S.: 1986, *Mind over Machine: The Role of Human Intuition and Expertise in the Era of Computers*, Free Press, New York.

Edwards, P.: 2001, Representing the global atmosphere: Computer models, data, and knowledge about climate change, *in* C. Miller and P. Edwarrds (eds), *Changing the Atmosphere. Expert Knowledge and Environmental Governance*, The MIT Press, Cambridge, London, pp. 31–65.

Emery, J.: 1987, *Management Information Systems, The Critical Strategic Resource*, Oxford University Press, New York.

Fink, E.: 2002, *Changes of Problem Representation*, Springer Verlag, Berlin, New York.

Glushkov, V.: 1972, Basic principles of automation in organizational management systems, *Upravlayushcheye Sistemy i Mashiny*.

Granat, J. and Makowski, M.: 2000, Interactive Specification and Analysis of Aspiration-Based Preferences, *EJOR* **122**(2), 469–485. available also as IIASA's RR-00-09.

Granat, J. and Wierzbicki, A. P.: 1994, Interactive specification of DSS user preferences in terms of fuzzy sets, *Working Paper WP-94-29*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Grauer, M., Thompson, M. and Wierzbicki, A. (eds): 1985, *Plural Rationality and Interactive Decision Processes*, Vol. 248 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

Hacking, I.: 1964, *Scientific Revolutions*, Oxford University Press, Oxford.

Hloyningen-Huene, P.: 1993, *Restructuring Scientific Revolutions*, The University of Chicago Press, London.

Isermann, H. and Steuer, R. E.: 1987, Computational experience concerning payoff tables and minimum criterion values over the efficient set, *European J. Oper. Res.* **33**, 91–97.

Kahneman, D. and Tversky, A.: 1982, The psychology of preferences, *Scientific American* **246**, 160–173.

Keeney, R. and Raiffa, H.: 1976, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, J. Wiley & Sons, New York.

Knight, J.: 2002, Statistical error leaves pollution data up in the air, *Nature* **417**(13), 677.

Kuhn, T.: 1964, A function of thought experiments, *in* I. Hacking (ed.), *Scientific Revolutions*, Oxford University Press, Oxford. Originally published in *L'aventure de la science, Melanges Alexandre Koyre*, Vol. **2**, pp. 307-334, Hermann, Paris 1964.

Kuhn, T.: 1970, *The Structure of Scientific Revolutions*, The University of Chicago Press, Chicago.

Lewandowski, A. and Wierzbicki, A. (eds): 1989, *Aspiration Based Decision Support Systems: Theory, Software and Applications*, Vol. 331 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

Liu, B.: 2002, *Theory and Practice of Uncertain Programming*, Springer Verlag, Berlin, New York.

Lorentz, K.: 1965, *Evolution and Modification of Behavior: A Critical Examination of the Concepts of the "Learned" and the "Innate" Elements of Behavior*, The University of Chicago Press, Chicago.

Maclean, D.: 1985, Rationality and equivalent redescriptions, *in* M. Grauer, M. Thompson and A. Wierzbicki (eds), *Plural Rationality and Interactive Decision Processes*, Vol. 248 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York, pp. 83–94.

Makowski, M.: 1994a, LP-DIT, Data Interchange Tool for Linear Programming Problems, (version 1.20), *Working Paper WP-94-36*, International Institute for Applied Systems Analysis, Laxenburg, Austria. Available on-line from `http://www.iiasa.ac.at/~marek/pubs`.

Makowski, M.: 1994b, Methodology and a modular tool for multiple criteria analysis of LP models, *Working Paper WP-94-102*, International Institute for Applied Systems Analysis, Laxenburg, Austria. Available on-line from `http://www.iiasa.ac.at/~marek/pubs/`.

Makowski, M.: 2000, Modeling paradigms applied to the analysis of European air quality, *EJOR* **122**(2), 219–241. available also as IIASA's RR-00-06.

Makowski, M.: 2001, Modeling techniques for complex environmental problems, *in* M. Makowski and H. Nakayama (eds), *Natural Environment Management and Applied Systems Analysis*, International Institute for Applied Systems Analysis, Laxenburg, Austria, pp. 41–77. ISBN 3-7045-0140-9.

Makowski, M.: 2004, Structured modeling technology, *EJOR*. (to appear).

Makowski, M. and Granat, J.: 2000, Interfaces, *in* Wierzbicki et al. (2000), pp. 283–307. ISBN 0-7923-6327-2.

Makowski, M. and Wierzbicki, A.: 2000, Architecture of decision support systems, *in* Wierzbicki et al. (2000), pp. 48–70. ISBN 0-7923-6327-2.

Makowski, M. and Wierzbicki, A.: 2003, Modeling knowledge in global information networks, *4th Global Research Village Conference. Importance of ICT for Research an Science: Science Policies for Economies in Transition*, KBN (the Polish State Committee for Scientific Research), and OECD (the Organization for Economic Co-operation and Development), Information Processing Centre, Warsaw, pp. 173–186. draft version available from `http://www.iiasa.ac.at/~marek/pubs/prepub.html`.

Makowski, M., Somlyódy, L. and Watkins, D.: 1996, Multiple criteria analysis for water quality management in the Nitra basin, *Water Resources Bulletin* **32**(5), 937–951.

Ogryczak, W.: 1996, A note on modeling multiple choice requirements for simple mixed integer programming solvers, *Computers & Operations Research* **23**, 199–205.

Ogryczak, W. and Zorychta, K.: 1996, Modular optimizer for mixed integer programming, MOMIP version 2.3, *Working Paper WP-96-106*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Paczyński, J., Makowski, M. and Wierzbicki, A.: 2000, Modeling tools, *in* Wierzbicki et al. (2000), pp. 125–165. ISBN 0-7923-6327-2.

Pawlak, Z.: 1991, *Rough Sets. Some Aspects of Reasoning about Knowledge*, Kluwer Academic Publishers, Dordrecht.

Polak, E.: 1976, On the approximation of solutions to multiple criteria decision making problems, *in* M. Zeleny (ed.), *Multiple Criteria Decision Making*, Springer-Verlag, New York. Originally published in *L'aventure de la science*, *Melanges Alexandre Koyre*, Vol. **2**, pp. 307-334, Hermann, Paris 1964.

Popper, K.: 1959, *The Logic of Scientific Discovery*, Hutchinson, London.

Popper, K.: 1975, The rationality of scientific revolutions, *in* R. Harre (ed.), *Problems of Scientific Revolution*, University Press, Oxford, pp. 72–101.

Popper, K.: 1983, *Realism and the Aim of Science*, Hutchinson, London.

Pospelov, G. and Irikov, V.: 1976, *Program- and Goal-Oriented Planning and Management*, Sovietskoye Radio, Moscow.

Radermacher, F.: 1994, Decision support systems: Scope and potential, *Decision Support Systems* **12**(4/5), 257–265.

Raiffa, H.: 1997, *Decision Analysis: Introductory Lectures of Choices Under Uncertainty*, MacGraw-Hill Companies, New York, Tokyo, Toronto.

Rapoport, A.: 1989, *Decision Theory and Decision Behaviour, Normative and Descriptive Approaches*, Vol. 15 of *Theory and Decision Library, Mathematical and Statical Methods*, Kluwer Academic Publishers, Dordrecht, Boston, London.

Ruan, D., Kacprzyk, J. and Fedrizzi, M. (eds): 2001, *Soft Computing for Risk Evaluation and Management*, Physica Verlag, New York.

Sakawa, M.: 1993, *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, New York, London.

Sawaragi, Y., Nakayama, H. and Tanino, T.: 1985, *Theory of Multiobjective Optimization*, Academic Press, New York.

Seo, F. and Sakawa, M.: 1988, *Multiple Criteria Decision Analysis in Regional Planning: Concepts, Methods and Applications*, D. Reidel Publishing Company, Dordrecht.

Simon, H.: 1955, A behavioral model of rational choice, *Quarterly Journal of Economics* **69**, 99–118.

Simon, H.: 1957, *Models of Man*, J. Wiley & Sons, Chichester, New York.

Simon, H.: 1958, *Administrative Behavior, a Study of Decision Making Process in Administrative Organization*, Macmillan, New York.

Sprague, R.: 1983, A framework for the development of decision support systems, *Decision Support Systems: A Data Based, Model-Oriented, User-Developed Discipline*, Petrocelli, Princeton, N.J.

Springer, S. and Deutsch, G.: 1981, *Left Brain - Right Brain*, Freeman, San Francisco.

Steuer, R.: 1986, *Multiple Criteria Optimization: Theory, Computation, and Application*, J. Wiley & Sons, New York.

Stewart, T.: 1992, A critical survey on the status of multiple criteria decision making theory and practice, *OMEGA, International Journal of Management Science* **20**(5/6), 569–586.

Tukey, J.: 1977, *Exploratory Data Analysis*, John Wiley & Sons, New York.

Tversky, A. and Kahneman, D.: 1985, The framing of decisions and philosophy of choice, *in* G. Wright (ed.), *Behavioral Decision Making*, Plenum, New York, pp. 25–42.

Vollmer, G.: 1984, Mesocosm and objective knowledge, *in* Wuketits (1984b).

Wierzbicki, A.: 1977, Basic properties of scalarizing functionals for multiobjective optimization, *Mathematische Operationsforschung und Statistik, s. Optimization* **8**, 55–60.

Wierzbicki, A.: 1980, The use of reference objectives in multiobjective optimization, *in* G. Fandel and T. Gal (eds), *Multiple Criteria Decision Making, Theory and Applications*, Vol. 177 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York, pp. 468–486.

Wierzbicki, A.: 1982, A mathematical basis for satisficing decision making, *Mathematical Modelling* **3**(5), 391–405.

Wierzbicki, A.: 1984, *Models and Sensitivity of Control Systems*, Elsevier-WNT, Amsterdam, Warsaw.

Wierzbicki, A.: 1986, On the completeness and constructiveness of parametric characterizations to vector optimization problems, *OR Spektrum* **8**, 73–87.

Wierzbicki, A.: 1992a, Multi-objective modeling and simulation for decision support, *Working Paper WP-92-80*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Wierzbicki, A.: 1992b, Multiple criteria games: Theory and applications, *Working Paper WP-92-79*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Wierzbicki, A.: 1992c, The role of intuition and creativity in decision making, *Working Paper WP-92-78*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Wierzbicki, A.: 1993, On the role of intuition in decision making and some ways of multicriteria aid of intuition, *Multiple Criteria Decision Making* **6**, 65–78.

Wierzbicki, A. and Makowski, M.: 2000, Modeling for knowledge exchange: Global aspects of software for science and mathematics, *in* P. Wouters and P. Schröder (eds), *Access to Publicly Financed Research*, NIWI, Amsterdam, the Netherlands, pp. 123–140.

Wierzbicki, A. and Wessels, J.: 2000, The modern decision maker, *in* Wierzbicki et al. (2000), pp. 29–46. ISBN 0-7923-6327-2.

Wierzbicki, A., Makowski, M. and Wessels, J. (eds): 2000, *Model-Based Decision Support Methodology with Environmental Applications*, Series: Mathematical Modeling and Applications, Kluwer Academic Publishers, Dordrecht. ISBN 0-7923-6327-2.

Wuketits, F.: 1984a, Evolutionary epistemology - a challenge to science and philosophy, in *Concepts and Approaches in Evolutionary Epistemology* (Wuketits, 1984b).

Wuketits, F. (ed.): 1984b, *Concepts and Approaches in Evolutionary Epistemology*, D. Reidel Publishing Co., Dordrecht.

Yu, P.: 1985, *Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions*, Plenum Press, New York, London.

Yu, P.: 1990, *Forming Winning Strategies, An Integrated Theory of Habitual Domains*, Springer Verlag, Berlin, New York.

Yu, P.: 1995, *Habitual Domains: Freeing Yourself from the Limits on Your Life*, Highwater Editions, Shawnee Mission, Kansas.

Zadeh, L.: 1965, Fuzzy sets, *Information and Control* **8**, 338–353.

Zadeh, L. and Kacprzyk, J.: 1999, *Computing with Words in Information/Intelligent Systems: Foundations*, Springer Verlag, Berlin, New York.

Zeleny, M.: 1974, A concept of compromise solutions and the method of the displaced ideal, *Comput. Oper. Res.* **1**, 479–496.

Zimmermann, H.: 1978, Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems* **1**, 45–55.

Zimmermann, H.: 1985, *Fuzzy Set Theory – and Its Applications*, Kluwer Academic Publishers, Boston, Dordrecht, Lancaster.

Zimmermann, H.: 1987, *Fuzzy Sets, Decision Making, and Expert Systems*, Kluwer Academic Publishers, Boston, Dordrecht, Lancaster.