THESIS ON INFORMATION AND SYSTEMS ENGINEERING C34

Structural Viability Criteria in Analysis of Complex Datasets

EINAR POLIS

TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology
Department of X
Chair of X

The thesis was accepted for defense of the degree of Doctor of Philosophy in Computer Engineering on MMM DD, 201X

Supervisors: Prof. Enn Õunapuu

Opponents:

Defense of the thesis: MMM DD, 201X

Declaration: I hereby declare that this doctoral thesis, submitted for the doctoral degree at Tallinn University of Technology, is my original investigation and achievement and has not been submitted for any other degree or examination.

/Einar Polis/

Abstract

Kokkuvõte

Acknowledgements

CONTENTS

INTRODUCTION.......... 5
The Chapters to Come.......... 5
Conventions Used.......... 5
1. UNDERLYING THEORIES.......... 6
1.1 Philosophical Background.......... 6
1.2 Metaphoric Interpretation.......... 10
1.3 Wholistic Analysis and Design.......... 16
1.4 Infotechnological Applications.......... 35
2. CASE STUDY.......... 43
2.1 Sample Domain.......... 43
2.2 Data Generation.......... 47
2.3 Data Collection.......... 47
3. PROBLEM STATEMENT.......... 49
3.1 Summary of Issues.......... 49
3.2 Current Limitations.......... 50
3.3 Research Question.......... 50
3.4 Research Methodology.......... 51
4. ANALYSIS PROCESS.......... 53
4.1 Framework.......... 53
4.2 Property Generation.......... 56
4.3 Sequence Generation.......... 57
4.4 Case Study Data Analysis.......... 57
5. CONFIRMATORY SURVEY.......... 61
5.1 Survey Design.......... 61
5.2 Survey Conduction.......... 62
5.3 Survey Analysis.......... 62
CONCLUSIONS.......... 63
Main Contributions.......... 63
Future Research.......... 63
REFERENCES.......... 64
APPENDICES.......... 68
A1. Common Functions.......... 68
CURRICULUM VITAE.......... 69
ELULOOKIRJELDUS.......... 71
LIST OF PUBLICATIONS.......... 73

List of Figures

Figure 1. Locations with less and more life 8
Figure 2. Conceptual source to target mapping... 11
Figure 3. LIVING BUILDING blend.. 16
Figure 4. The whole, centers, and fields.. 17
Figure 5. STRONG-CENTER property example 18
Figure 6. THICK-BOUNDARY property example.. 19
Figure 7. LEVELS-OF-SCALE property example 20
Figure 8. ALTERNATING-REPETITION property example 21
Figure 9. LOCAL-SYMMETRIES property example 22
Figure 10. POSITIVE-SPACE property example.. 23
Figure 11. ROUGHNESS property example 24
Figure 12. CONTRAST property example... 25
Figure 13. DEEP-INTERLOCK AND AMBIGUITY property example.................... 26
Figure 14. ECHOES property example 27
Figure 15. GOOD-SHAPE property example 28
Figure 16. INNER-CALM property example... 29
Figure 17. THE VOID property example... 30
Figure 18. NOT-SEPARATENESS property example... 31
Figure 19. Wholeness-extending transformation sequence example 33
Figure 20. Overview of Thriving Systems Theory.. 36
Figure 21. INTIMACY GRADIENT pattern.. 38
Figure 22. Data analysis pattern language... 42
Figure 23. Case study customer site examples.. 44
Figure 24. LD module analysis visualization... 59
Figure 25. AA module analysis visualization... 60
Figure 26. Survey form 62

List of Tables

Table 1. Objectivist and constructivist interpretations in ISD............................. 9
Table 2. DATA IS BUILDING metaphor mapping 13
Table 3. Fundamental property interaction matrix... 32
Table 4. Role separation in core banking system projects 46
Table 5. DATA ENTITY (ORGANIC) metaphor 55
Table 6. DATA ENTITY (MECHANISTIC) metaphor.. 55

INTRODUCTION

This introductory chapter discusses the aim of thesis and explains conduction of underlying research, presents the overall organization of thesis and conventions used

The Chapters to Come

Conventions Used

Term first used in *italic*

Names of structural properties, patterns, blends in SMALL CAPS

Blocks of commented R code with outputs:

```
#these are sample commands with output
rver <- R.Version()
rver$nickname

## [1] "Wooden Christmas-Tree"
```

'##' denotes output of the last command

1. UNDERLYING THEORIES

This chapter presents an overview of the underlying theories

1.1 Philosophical Background

Assumptions made and approach taken will be explained below

1.1.1 Commonsense Assumptions

This thesis has been heavily influenced by theories of architectural theorist and practitioner Christopher Alexander (1977; 1979; 2002; 2005; 2012). Inspired by these, we consider some elementary common sense claims to be true:

- Things benefit from people really caring of them;
- People who care benefit from caring as well;
- People tend to care of particular kind of living things.

Related to these claims, in our particular subject area of interest, we attempt to add value by proving that:

- Similar quality can be attained by design;
- Existence of such quality can be measured.

We can illustrate these assumptions with well-known 10,000-year clock design challenge. Brand (2000) brings out, how in building something which is expected to last that long, there are not only technical problems to overcome, but it is also very important to make it appealing to the people. Although extreme data longevity is not exactly the topic of this thesis, designing data sources which can serve well for a long time is a closely related desirable goal

1.1.2 Reductionist vs. Holistic Approaches

Commonsense assumptions from the previous point do not seem to fit well into contemporary worldview. Spirituality generally has no place neither in sciences of natural nor in sciences of artificial (to use Simon's (1996) way of categorization). Balance in both sciences is very much in favor of reductionism over holism. Nutritional scientist Colin T. Campbell explains these concepts in the following way:

> If you are a reductionist, you believe that everything in the world can be understood if you understand all its component parts. A wholist, on the other hand, believes that the whole can be greater than the sum of its parts (Campbell, 2013:47).

He claims that "modern science (and the biological and health sciences in particular) has embraced the dogma of reductionism to the exclusion of common sense and fairness (Campbell, 2013:56)." This may have very serious practical consequences. His example concerns widespread rejection of the whole-food plant-based diet as an option from scientific and public debates, thus potentially missing the opportunity to solve efficiently numerous health and ecology problems.

Stuart Kauffman (2008) is another distinguished scientist who has extensively written on such issues. He presents his main concerns regarding problems with contemporary scientific worldview as “4 injuries”, which should be overcome by “reinventing the sacred”:

1) *Division between science and humanities*. There is a “separate worlds” problem, but it must be realized that science is not the only path to knowledge and understanding;
2) *Reductionist worldview*. The world we live in is a world of fact without values;
3) *Spirituality is at best questionable*. We are thus cut off from deep aspects of our humanity;
4) *Lack of global ethics*. Lack a shared framework of values that spans our traditions and our responsibilities to all of life.

These injuries marginalize the role of science and scientist in society. Moreover, our comprehension of the emerging complexity may suffer as well.

Situation with sciences of artificial is not better. Alexander has written extensively over the years on ugliness of most modern buildings and townscapes, which people really do not like, compared to those produced by something he calls *natural process* in traditional cultures. He started with quite formal approach (Alexander, 1964), moved on by introducing more subjective (e.g. Quality Without a Name) elements (Alexander et al, 1977; Alexander, 1979), and culminated with recognizing the importance of spiritual aspects (Alexander, 2005) in creation of truly beautiful objects.

Perhaps the grounding idea of Alexander’s approach is attribution to every object in the world a certain degree of life. This degree is a characteristic of the object which at the same time it is connected and affecting people having relation to it. These people will also possess various degree of life, depending on liveliness level of the related objects. Alexander seems to come to this conclusion mostly by asking people to describe their feelings in comparing different objects and by finding many similarities in answers. Alexander (2002) contains several examples of comparisons, herein we demonstrate his approach by comparing liveliness levels of city center neighborhoods in “Figure 1. Locations with less and more life”.

As provided sample comparison hopefully illustrates, such experiments can be easily influenced by artistic rendition of objects, individual tastes, fashion and preference to conditions closer to their natural habitat, etc. For example, conditions displayed on the second photo took centuries to evolve and are requiring relatively little effort to maintain, while in the surrounding on second photo are result of rapid property and have not had the time to evolve. Alexander’s books contain photos by many renowned photographers (e.g. Henri Cartier-Bresson), who certainly have been capable of projecting desired qualities. To reduce the artistic bias, this thesis intentionally relies only on author’s amateur photos, taken by ordinary phone camera and having no aesthetic aspirations whatsoever.

Figure 1. Locations with less and more life

Described difficulties in comparing the photos have already pointed to many problems with subjectivity. We can add avoidance of religious dogma and other historical factors to this, to understand better the reasons for prevalence of reductionism. Campbell is very well aware of this:

> Holism reminds scientists of the word *holy*, which smacks of religion. And many scientists are as hostile to religion as religious fundamentalists are to science. When they encounter the word *holistic*, they think of sloppy, “fairy-tale” belief systems that have no place in a serious exploration of the “real world” (Campbell, 2013:48).

We agree with this reasoning and follow his example by using *wholism* spelling instead of *holism* in our discussion of analysis methods, to indicate awareness of related issues. Possibly one of the reasons for not learning from mistakes is the lack of emotional connection with outcomes. We will rely more than customary on wholistic approach, which does not have to mean the lack of measurability and reproducibility.

Computer science and information technology fields are suffering from similar problems. Low success rate of information technology projects is widely known. On a very generic level, we can say that the data is organised into *information systems*. This term has been widely used for many decades. Being a combination of two difficult to define terms, it is utilized for referring to wide range of socio-technical creations. Several attempts have been made to define this term and main domain concepts rigorously. These are dependent on interpretation and distinction of fundamental categories like data, information and knowledge; also from selection of system elements. A typical example of formal informal system definition is provided by van Hee (1994), according to which "Discrete dynamic system consists of active components or actors that consume and produce passive components or tokens (van Hee, 1994: 4). …another subtype of discrete dynamic systems is called an information system and is characterized by two properties (i) all actors in an information system are consuming and producing information objects exclusively; (ii) an information system is part of another discrete dynamic system (called the target system in this context); the role of the information system is to support the target system (van Hee, 1994:10).

Table 1. Objectivist and constructivist interpretations in ISD

Ontological Unit	**Objectivist Interpretation**	**Constructivist Interpretation**
Data/Information/ Knowledge	Descriptive facts and relevant descriptive facts	Socially constructed meanings signifying intentions
Information Systems	Technical system	A form of social systems realizing human intentions
Human Beings	Deterministic systems, rational agents	Voluntaristic systems with consciousness and free will
Technology	Technology as a causal agent (technological determinism)	Malleable structures subject to social and human choice
Organizations and Society	Stable structures	Interaction systems or socially constructed systems (nominalism)

Differences in images/mental models and value systems can be illustrated by the well-known cognitive gap between two prominent stakeholder groups: so called business people and information systems developers. Social systems can only be sensed indirectly and are more dependent on our way of thinking than engineered systems. Dietz (2006) explains how communications between individual people give arise to different kind of enterprises and how according to interpretative paradigm such enterprises have a tendency to become similar to what we are thinking of them. However, software engineers tend to be having an objectivist viewpoint. To illustrate this difference, we adopted “Table 1. Objectivist and constructivist interpretations in ISD” from Iivari et al. (1998).

Objectivist and constructivist interpretations are certainly interrelated. For example, we can make many assumptions on the organizational culture of a particular enterprise by looking at the related artifacts, even at the software, as to more tangible items for basing the analysis. On the other hand, in many cases the design of a technological system is heavily influenced by constructivist or subjectivist reasoning

1.2 Metaphoric Interpretation

Overview of conceptual metaphor theory with relevant examples

1.2.1 Conceptual Metaphors

Lakoff & Johnson (2003) have developed a comprehensive theory for integrating subjective and objective worldviews by what they call *conceptual metaphor*. According to this theory, metaphors are not just a matter of language (literal metaphors), but something on what our entire conceptual system is based.

People in general are supposedly equipped with roughly the same set of primary metaphors. These are physical and arise from the fact that we have bodies of the sort we have and that they function as they do in our physical environment. Primary metaphors form a basis for emergent metaphors, which arise through our interactions with the world. There is no such thing as a meaning of a sentence in itself, independent of any people. Meaning mostly arises in terms of other concepts and understanding through experiential gestalt.

Metaphors have a fundamental role in innovative thinking and creating a new meaning. They unite reason and imagination. Imaginative language helps to identify a possibility space for alternative solutions. Metaphors allow the designer to think unconventionally and encourage the application of novel ideas to design problems. As design is in many ways a collaborative work, they are also helpful in consolidating different perspectives. In general, the employment of metaphors for example by building architects is well documented in literature through a vast number of examples. Nevertheless, not many empirical investigations have been conducted to verify the contribution of metaphors to design (Casakin, 2007).

Metaphors are also frequently used in explaining the role software intense systems. Perhaps the most common is nervous system metaphor, used for example by Gates (1999). Larger variety of metaphors is coming for software and software development domains. McConnell (2004) has proposed software farming metaphor and collected references to other metaphors used in software development: science, art, process, hunting, driving a car, game, bazaar, and more. Spinellis et al. (2009) has proposed an agile concept of metaphor and found music metaphor particularly descriptive for explaining software, more appropriate than frequently used building analogy. Music and improvisational music in particular, or just improvisation (Kautz, 2009) has been sometimes used for referring not only to software, but also to its development process

1.2.2 Transfer of Meaning

Understanding and experiencing one kind of thing in terms of another is the essence of a metaphor. Lakoff & Johnson (2003) propose, that this happens in a systematical manner. Two most common mechanisms are described below.

Mapping. All metaphors are structural (in that they map structures to structure); all are ontological (in that they create target domain entities); and many of them are orientational (in that they map orientational image-schemas). This facilitates highlighting or hiding certain aspects of the phenomena. We illustrate concept mapping in "Figure 2. Conceptual source to target mapping" by using projections P_1 to P_3 from source to target domain:

- P_1: does not reach the target, thus hiding related parts of source domain. As an example, in case of metaphor THEORIES ARE BUILDINGS, we usually would not expect theories to have corridors and these parts will be omitted from projection;
- P_2: reaches the target and enriched it with new meaning. For example, we can understand how theories are built if we understand building domain;
- P_3: is not originated from source domain. There are parts of target domain unique to it or originated from other sources. For example, proving the theory will not be explained by building domain concepts.

We should note that the distinctions described above are somewhat arbitrary. It is very well possible to imagine a theory to have dark corridors with no end. However, this is not a common tradition and requires creative efforts for changing the common understanding.

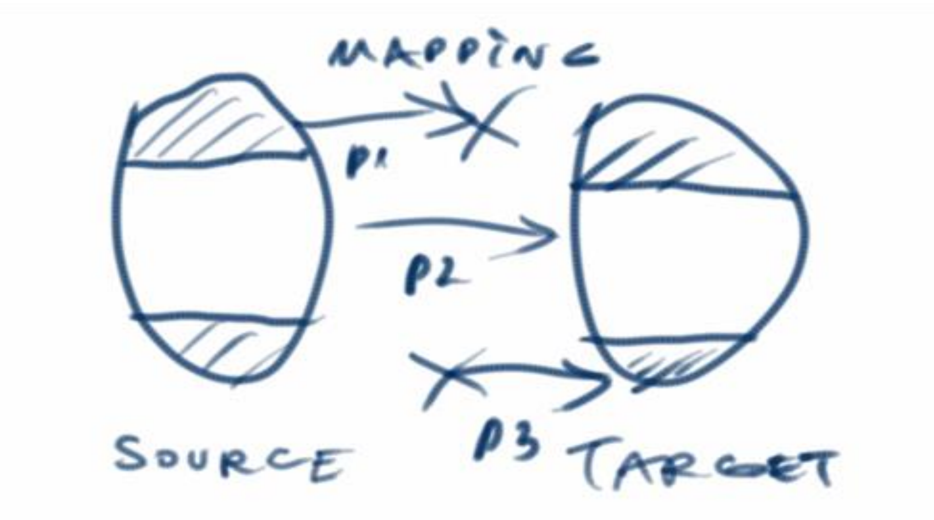


Figure 2. Conceptual source to target mapping

Blending. Basic domain to domain mapping might not be sufficient and we need to have more domains connected as a network. Fauconnier and Turner (2002) proposed a structure for such network and called this kind of conceptual metaphor formation *blending* or conceptual integration. The structure has the following specifically interconnected parts (spaces):

- Generic space. Determines common elements;
- Input spaces. May related to each other as source and target domains;
- Blend. Emerges from input spaces to form a conceptual metaphor.

Connection of spaces and placement of particular elements is illustrated based on a practical example in "Figure 3. LIVING BUILDING blend". The figure also highlights how conceptual metaphors typically employ a more abstract concept as target and more concrete physical concepts as their source (Kövecses, 2010)

1.2.3 Building Source Domain

This thesis follows a long tradition of explaining software-intensive system concepts with metaphors sourced from building domain. For example, Booch et al. (2005) have famously illustrated expected modelling effort levels by comparing them with differences in architectural work required for building a dog kennel, a house or a skyscraper.

Software architecture concept itself seems to have emerged from building architecture. Of course, computerized information systems have not been around for thousands of years, but still many theoreticians and practitioners are sure that many of the concepts are transferrable. By *architecture* is meant here organizational principles of the structure of an artifact, especially of a complex artifact (in different domains, languages, cultures, and by different authors, terms 'design' and 'architecture' may have slightly different meanings). Selection of an appropriate architecture is one of the most important tasks in design. For example, modular architecture might be appropriate if flexibility is required, while integrated architecture is suitable in case of extreme optimization. Architecture is usually related to specifying location of things and their relationships in two-dimensional or three-dimensional space.

Alexander is an architect who has had most influence in software-intense system designs, especially by introduction of a design pattern concept (Alexander, 1977). Later he has written extensively on natural ways for creating great buildings (Alexander, 1979; 2003) and about problems caused by mechanistic worldview in building architecture. For example, previous century all over the world huge amounts of urban housing was developed by following Le Corbusier's famous metaphor about houses as "machines built for living in". Many problems with them appeared later, people do not want to inhabit them anymore and large blocks containing such buildings have been demolished.

This thesis is focused on a subset of information technology, thus making formal mapping of concepts would be easier. On the other hand, completeness cannot be guaranteed and not even aimed. "Table 2. DATA IS BUILDING metaphor" is the result of mapping the most relevant concepts.

Table 2. DATA IS BUILDING metaphor mapping

Building	Data	Explanation
some construction elements	-	No obvious analogies in data domain (e.g. balcony, stairs, etc. not commonly used in discussing data)
ornament	-	No obvious analogies in data domain
some construction elements	similarly named elements	Foundation, doors, windows can have useful analogies for structuring (e.g. core, interfaces, views)
material	format	Using lasting materials has much in common to using viable storage formats
integrity	integrity	Structural integrity of the building has parallels with data integrity
beauty	presentation	Beauty of data can be abstract, but not of its presentation
-	compression	No obvious analogies in building domain
-	encryption	No obvious analogies in building domain

Blending of building, life (discussed in next point), and mind domain concepts helps to clarify liveliness in buildings and subsequently in people, as claimed by Alexander. We will be able only to present a very simplified presentation of a complex blend, just for highlighting the main points

1.2.4 Life Source Domain

Among the most often used metaphor sources with a lot explanatory power are the machine and the organism. Metaphors from life domain have found particularly wide coverage in organizational analysis and in management literature in general. Some theorists and practitioners have studied long-term success of enterprises and found many similarities between them and living systems (Collins, 2004; de Geus, 1997; Korten, 1999; Ashar & Lane-Maher, 2004). Others (Morgan, 1997) have found the life metaphor just useful for explaining the functioning of enterprises, especially when they behave like intelligent or learning entities. On the other hand, thinking patterns in modern organizations seems still to be heavily influenced by once very successful reductionist scientific management thinking.

Living systems domain covers an enormous range of phenomena in the natural world. Many generalizations and a rigorous selection of essential characteristics have to be made in providing an overview of it. In doing this is possible to rely on several theories developed in recent decades which are attempting the same. However, those theories tend to be voluminous too, frequently presenting a life-long effort by their authors and requiring thousands of pages for presenting them entirely. An attempt has been made below to gather together aspects of living systems which are most characteristic and relevant to this thesis.

Despite many efforts in biological and ecological sciences, no commonly accepted definition of life exists. In a very literal sense, we can mean by life a complex phenomenon caused by a certain type of carbon-based chemistry.

Specific characteristics are attributable to its main units (organisms), to their subsystems (cells) and suprasystems (groups). Some of the distinctive characteristics all or most of them possess are the following:

- *Metabolism.* Transformation of energy by converting chemicals and energy into cellular components (anabolism) and decomposing organic matter (catabolism). Living things require energy to maintain internal organization and to produce the other phenomena associated with life;
- *Growth.* Maintenance of a higher rate of anabolism than catabolism. A growing organism increases in size in all of its parts, rather than simply accumulating matter;
- *Homeostasis*. Regulation of the internal environment to maintain a constant state;
- *Adaptation.* The ability to change over time in response to the environment;
- *Response to stimuli.* Often expressed by motion, but can take many forms, from the contraction of a unicellular organism to external chemicals, to complex reactions involving all the senses of multicellular organisms;
- *Reproduction.* The ability to produce new individual organisms, either asexually from a single parent organism, or sexually from two parent organisms.

Several attempts have been made in the last few decades for gathering all the essential aspects of life into a comprehensive general living systems theory, despite of overwhelming size and complexity of the domain. Such theories should instead of breaking the phenomena down into components investigate dynamic patterns of the relationships of living entities with their environment. Two relatively well-known classic theories derived from generic systems theory are Living Systems Theory (LST) by Miller (1978) and Viable Systems Model (VSM) by Beer (1985). According to analysis by Scwaninger (2006), Miller's concept of complexity is objectivist and Beer's approach constructivist; he also found both theories "most probably complementary". Although the backgrounds of authors are different, they have been able to create notably interdisciplinary theories.

LST is an example of a theory which tackles the complexity of the domain by packaging and functional modeling. It states that living systems exist at eight levels of increasing complexity (cells, organs, organisms, groups, organizations, communities, societies, and supranational systems) and identifies 20 subsystems for processing matter/energy or information which are present at all levels. Entities are regarded as organized systems maintaining thermodynamically highly improbable energy states by continuously interacting with their environments. They thereby absorb substances with less entropy and greater information-content than they return to this environment. The critical subsystems for matter-energy processing are: ingestor, distributor, converter, producer, storage, extruder, motor, and supporter; for processing information: input transducer, internal transducer, channel and net, timer, decoder, associator, memory, decider, encoder, and output transducer. Two subsystems are common for both matter-energy and information: reproducer and boundary. Proposals have been made for utilizing LST in engineering. Cowan (1999) discusses usefulness for modeling systems accordingly on conceptual level. Samuelson (2006) has proposed a set of graphical symbols for presenting every subsystems in models. However, these have not found widespread use yet and also the other theories for functional modeling (e.g. ontology based) of living systems domain seem to become more common.

VSM has taken somewhat more abstract approach compared to LST, by focusing more on the informational perspective. This emphasizes importance of feedback mechanisms in decision making and consequent learning. Specific agency patterns arise in systems capable of doing that. We can have two different viewpoints on agents and results of their actions. More commonly life is considered something that life-forms, a special kind of things, are living from birth until the end of their existence. On the other hand, not only the existence in itself is important, but also the way of existence — alternatively, we can treat life as a special quality things possess, by being more or less alive. According to VSM, five subsystems are necessary to meet the demands of surviving in the changing environment for any autonomous system. These subsystems are present recursively in all levels of organization. Viability is system's ability to maintain its independent existence.

There are many more, sometimes mathematically advanced formal viability theories, as summarized by Aubin et al. (2011). However, prevalent theories to a large extent are ignoring certain particularly interesting aspects of viability. We will present alternative approach while discussing wholistic analysis and design in next point. In "Figure 3. LIVING BUILDING blend" we have presented a conceptual basis for this.

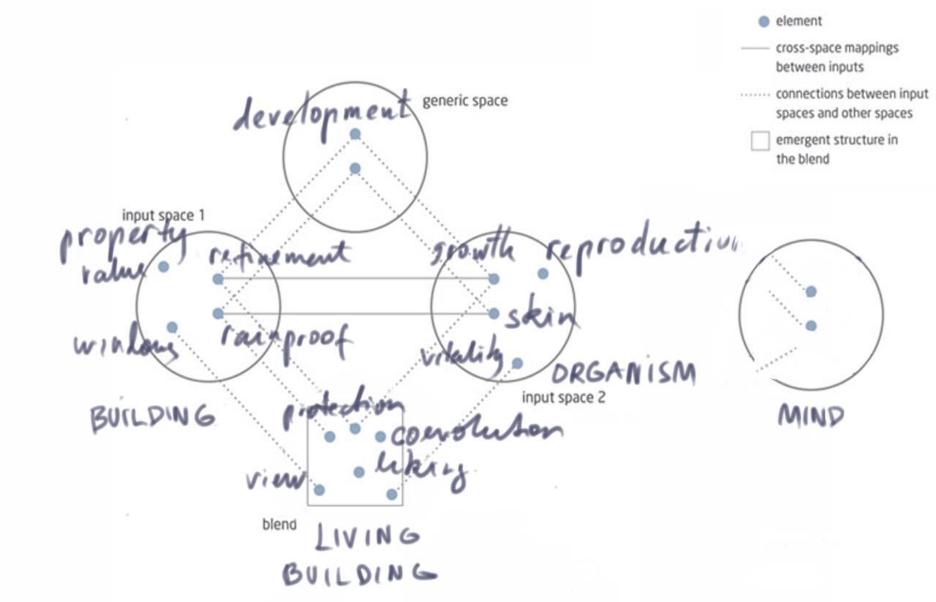


Figure 3. LIVING BUILDING blend

1.3 Wholistic Analysis and Design

Non-traditional approach based on earlier discussions. We are looking at this by relying mostly on Alexander's (2002; 2002a; 2005) work

1.3.1 <u>Base Elements</u>

Alexander's theory of life is an example of *structuralism*. There are structural similarities between the structures which have similar degree of life. To understand this, at a very abstract level we have to consider a particular system (entity) as wholeness formed by a field of interrelationships among centers in a space.

Wholeness determines how well different parts of a region in space interact. The wholeness is made of parts; the parts are created by the wholeness. This can be achieved only by co-evolution of the parts and the whole.

Centers are a special class of entities, which appear within the larger whole as distinct and noticeable parts which can be defined only recursively in terms of other centers. It is important to note that centers are not isolated entities, but work together for supporting life. In the living structure, each center is a being (Alexander, 2004).

Center always has a spatial locus and it forms fields. Natural order emerges as an arrangement of things based on their arrival at positions as a reaction to forces that guide their movement. These forces result from the relative location or context in which an element resides. Certain structures emerged from the continuous influence by forces will be preserved over time and develop into a whole; others will not have sufficient stability to persist and disappear in time.

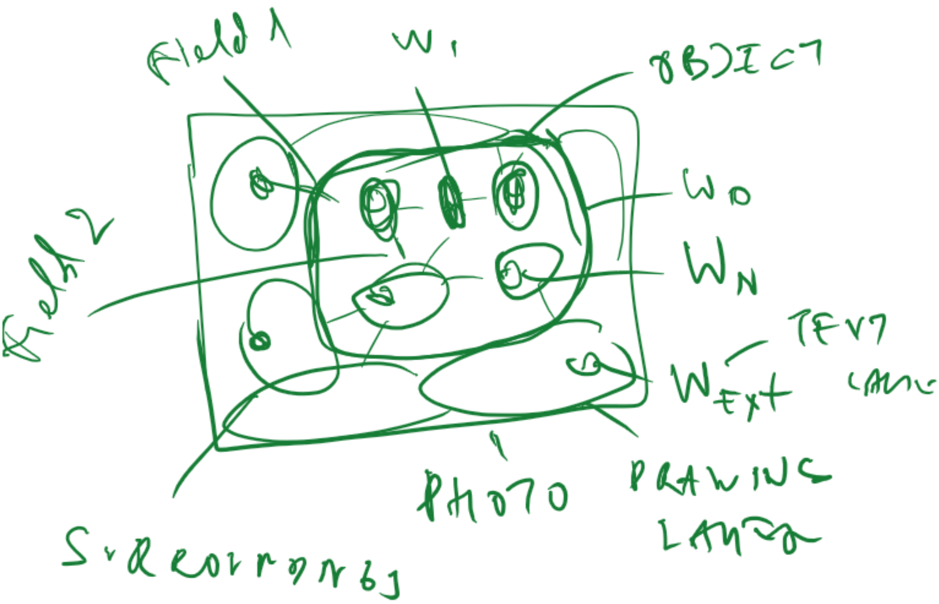


Figure 4. The whole, centers, and fields

We need executable definitions. Alexander presented his (especially later) theories in a form of lengthy textual explanations with photographic illustrations. But he realized and started working on more compact formal presentations as well:

> … define the wholeness W as the system which is created by the region R, together with the measure c and all those subregions which have measure more than some threshold and thus qualify as centers (Alexander, 2002:447).

1.3.2 Structural Properties

Certain center configurations or structural properties exist, which are constantly present in all living structures. Alexander identified around 15 of such properties. The exact number of properties is disputable, because of their interconnectedness (see “Table 3. Fundamental property interaction matrix”), but the order of magnitude should be certain. These properties provide valuable means for assessing liveliness of artificial systems.

STRONG-CENTER. This property defines the way that a strong center requires a spatial field-like effect, created by other centers, as the primary source of its strength (Alexander, 2002).

We illustrate this property by comparing two views at the city in “Figure 5. STRONG-CENTER property example”. On the upper photo we see no centers forming. On the lower photo strong centers are evident and clearly separating the office building (left), the church, and the bank (right).

Figure 5. STRONG-CENTER property example

Pronk et al. (2008) have written an article on mathematical beauty in architecture which attempts to collect existing quantitative statements and find new mathematical representations of 15 properties by Alexander. For strong-center property they proposed a centre of gravity of the specific field as suitable interpretation.

THICK-BOUNDARY. This property defines the way in which the field-like effect of a center is strengthened by the creation of a ring-like center, made of smaller centers which surround and intensify the first. The boundary also unites the center with the centers beyond it, thus strengthening it further (Alexander, 2002).

We illustrate this property by comparing two buildings in “Figure 6. THICK-BOUNDARY property example”. Building on the left has virtually no borders with surroundings and hardly stands out (does not form a strong center). Building on the right has thicker boundary to the world and also between its own structural elements.

Figure 6. T*HICK*-B*OUNDARY property example*

Pronk et al. (2008) note that according to Alexander a boundary needs to be of the same order of magnitude as the center which is being bounded. Boundaries are strongly related to geometrical forms. These start with a single point. The relation between two points can be defined as a line. A line can be further developed by implementing another point so that a triangular shape is created. This triangle transforms to a rectangle which transforms into a polygon by implementing new points within the total composition. The final form can be related to the base shape of the circle. So every bounding area can be related to this circular base form.

LEVELS-OF-SCALE. This property defines the way that a strong center is made stronger partly by smaller strong centers contained in it, and partly by its larger strong centers which contain it (Alexander, 2002).

We illustrate this property by comparing two bank buildings in “Figure 7. LEVELS-OF-SCALE property example”. Window-to-wall ratio and partitioning of windows is more proportional in case of the building on the lower photo.

Pronk et al. (2008) hypothesize in line with Alexander that perceptually good jump in scale is around 1:3 and note that this ratio seems to be closely related to the natural number e = 2.718 as well as π = 3.14.

Figure 7. LEVELS-OF-SCALE property example

ALTERNATING-REPETITION. This property defines the way in which centers are strengthened when they repeat, by the insertion of other centers between the repeating ones (Alexander, 2002).

We illustrate this property by comparing two sides of the street in "Figure 8. ALTERNATING-REPETITION property example". While on the left side the color of otherwise relatively similar-looking building alternates, it remains the same on the right side, making orientation on the latter side more difficult.

Figure 8. ALTERNATING-REPETITION property example

Pronk et al. (2008) show how alternation can be added by varying multiplier coefficients in simple series.

LOCAL-SYMMETRIES. This property defines the way that the intensity of a given center is increased by the extent to which other smaller centers which it contains are themselves arranged in locally symmetrical groups (Alexander, 2002).

We illustrate this property by comparing two buildings of large multinational banks in “Figure 9. LOCAL-SYMMETRIES property example”. Building on the upper photo possesses somewhat brutal global symmetry only. The other building is an example of more elaborate local symmetries which helps to adapt it with surrounding lower and smaller buildings.

Figure 9. LOCAL-SYMMETRIES property example

Pronk et al. (2008) add that the mirror effect is a result of mathematical beauty which is very easily recognized by the human brain. The symmetry of elements attracts the human eye to the central point where the mirroring takes place, so that the new center is being highlighted. Mirroring can be done in relation to a point, line or surface.

POSITIVE-SPACE. This property defines the way that a given center must draw its strength, in part, from the strength of other centers immediately adjacent to it in space (Alexander, 2002).

We illustrate this property by discussing the space between the cylindrical column and the wall in “Figure 10. POSITIVE-SPACE property example”. This space does not form a strong center; neither does it intensify the column. This is an example of suboptimal use of space. Perhaps the column is there to strengthen the construction, but it does not contribute to wholeness of the room. Space between the wall and the column is wasted and not positive.

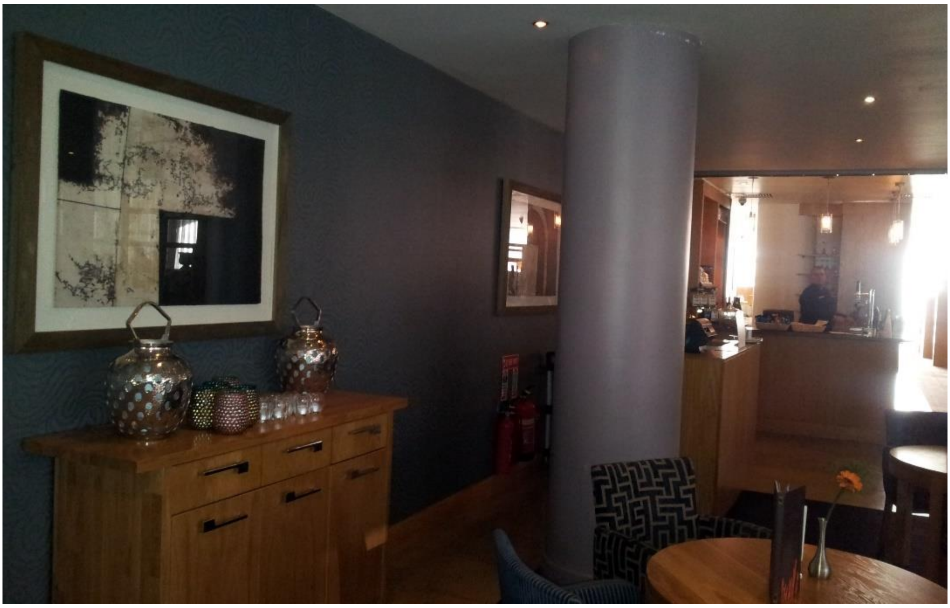
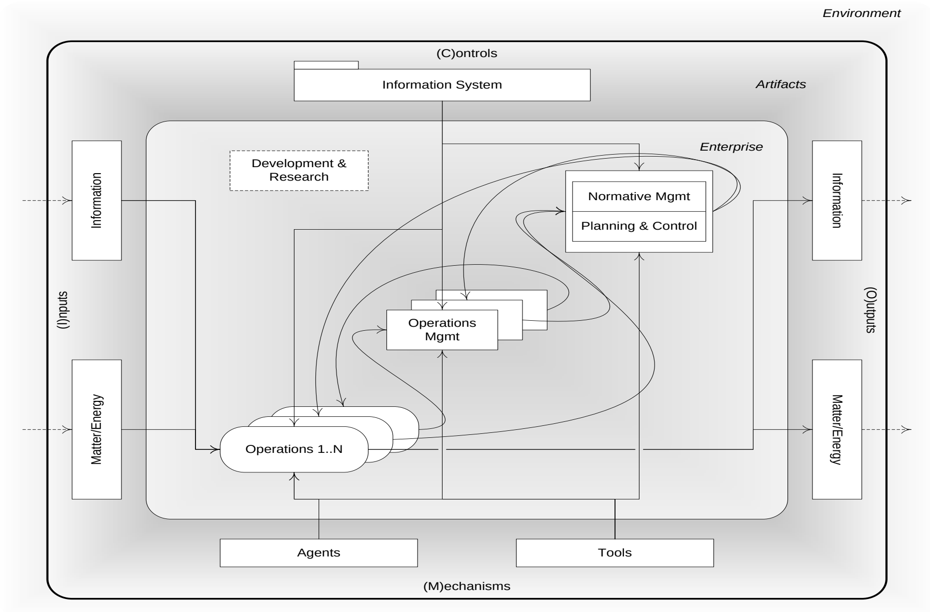


Figure 10. POSITIVE-SPACE property example

Pronk et al. (2008) explain this with the void which results from the positioning of mass has its own identity due to its usefulness and the relationship with the surrounding mass. When reviewing positive space in a two dimensional way, positive space can be the result of translation, rotation and minimal transformation of the negative space: the mass. Interaction between the mass and the void results in the fact that every bit of space is very intensely useful and that there is no leftover space which is not useful.

ROUGHNESS. This property defines the way that the field effect of a given center draws its strength, necessarily, from irregularities in the sizes, shapes, and arrangements of other nearby centers (Alexander, 2002).

We illustrate this property by comparing two graphical models in "Figure 11. ROUGHNESS property example". Upper model is presented using vector graphics, lower is a handwritten scribble. Polished image creates an impression of completeness, while the lower image might encourage conversations and further improvements.



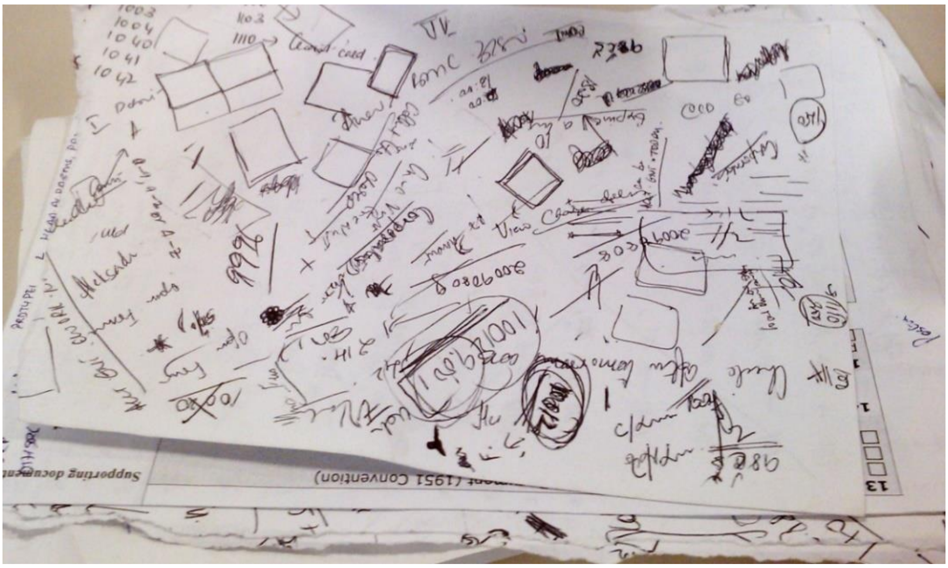


Figure 11. ROUGHNESS property example

Pronk et al. (2008) claim that this effect can be related to a normal deviation in which most points are very near to the average, with some exceptions.

GRADIENT. This property defines the way in which a center is strengthened by a graded series of different-sized centers which then “point” to the new center and intensify its field effect (Alexander, 2002).

We illustrate this property by house plan in “Figure 21. INTIMACY GRADIENT pattern”. The rooms in that house become gradually more personal, depending on their distance from the entrance. However, in this case the rooms are differentiated not so much based on their structural properties but on their designation.

Pronk et al. (2008) note that gradient shows remarkable similarity with mathematical functions. The mathematical concept of a function expresses dependence between two quantities; one which is given by the input which produces the output. Of major importance considering this function is that it should not be constant, but gradually changes. The function itself determines the rate, intensity and direction of the gradient. The tangent of the function is very important in defining these characteristics of the gradient at a certain position. When the tangent of a function is not zero, the related direction of the function describes a gradient.

CONTRAST. This property defines the way that a center is strengthened by the sharpness of the distinction between its character and the character of surrounding centers (Alexander, 2002).

We illustrate this property by comparing two neighboring office buildings in “Figure 12. CONTRAST property example”. Structural elements are quite clearly more distinguishable on the left building.

Figure 12. CONTRAST property example

Pronk et al. (2008) propose to use contrasting functions for understanding this structural property.

DEEP-INTERLOCK AND AMBIGUITY. This property defines the way in which the intensity of a given center can be increased when it is attached to nearby strong centers, through a third set of strong centers that ambiguously belong to both (Alexander, 2002).

We illustrate this property by comparing balconies of two hotels in "Figure 13. DEEP-INTERLOCK AND AMBIGUITY property example". Balcony on the upper photo forms a space between the hotel room and the street, which seem belong to both. Balcony on the lower photo seems unnatural extension of the hotel to the street, leaving the guest exposed and unwilling to stay there. This is not the case with the first balcony.

Figure 13. DEEP-INTERLOCK AND AMBIGUITY property example

Pronk et al. (2008) are not optimistic regarding possibility to formalize this property. In a psychological way of thinking it is possible to state that the interrelation between different elements can refer to mathematical symbols which are used to calculate with numbers. These symbols are also the mediators between the elements, the numbers, of math to create a new outcome. So when the property of deep interlock and ambiguity is considered in a more objective way, mathematics is not able to give a proper definition.

ECHOES. This property defines the way that the strength of a given center depends on similarities of angle and orientation and systems of centers forming characteristic angles thus forming larger centers, among the centers it contains (Alexander, 2002).

We illustrate this property by row of buildings in “Figure 14. ECHOES property example”. Majority of buildings loosely follow Georgian architecture style and are good candidates for forming a coherent whole. There is, however, modern building in the middle, similarly sized but still seemingly not in harmony with others.

Figure 14. ECHOES property example

Pronk et al. (2008) propose that the relation between echoing elements in a larger whole can be the result of a transformation by mathematical aspects like translation, scaling and rotation or a combination of them.

GOOD-SHAPE. This property defines the way that the strength of a given center depends on its actual shape, and the way this effect requires that even the shape, its boundary, and the space around it are made up of strong centers (Alexander, 2002).

We illustrate this property by comparing two bridges crossing the Liffey River in Dublin (see “Figure 15. GOOD-SHAPE property example”). The upper bridge is rather ordinary looking simple structure, while the lower bridge is more unusual contraction formed to resemble Irish national symbol of the harp.

Both seem to fit their surroundings, but more naturally shaped bridge seems to attract more people to stop and admire the scenery.

Figure 15. G*OOD*-S*HAPE property example*

Pronk et al. (2008) found this property to create a problem mathematically and that it is impossible to define a good shape in an objective manner.

INNER-CALM (also known as SIMPLICITY). This property defines the way the strength of a center depends on its simplicity — on the process of reducing the number of different centers which exist in it, while increasing the strength of these centers to make them weigh more (Alexander, 2002).

We illustrate this property by comparing two at first glance very similar-looking buildings in "Figure 16. INNER-CALM property example". Remarkable differences still exist. The building on the upper photo is designed to look almost angry, to stand out, perhaps intentionally. On the other hand, building on the lower photo possess much more simplicity and thus having very different, almost calming, effect.

Figure 16. INNER-CALM property example

Pronk et al. (2008) stress that in order to create understandable architecture the use of very simple, geometrical shapes is necessary. The proportions of these geometrical shapes need to be unusual. Therefore, the shapes are a result of an unusual parameter like length, size, width, etc. It is also important that there are little imperfections within the total composition, although they are necessary to create a living whole as explained in the paragraph about roughness. Mathematically this means the use of understandable shapes, like squares and triangles, yet they should have one unusual parameter which will determine the composition.

THE VOID. This property defines the way that the intensity of every center depends on the existence of a still place — an empty center — somewhere in its field (Alexander, 2002).

We illustrate this property by comparing two public squares in "Figure 17. THE VOID property example". The upper photo depicts a place which almost seems like a waste of space, a location where nobody wants to stay. On the lower photo the space between modern buildings is more pleasantly arranged, forming a strong center in itself.

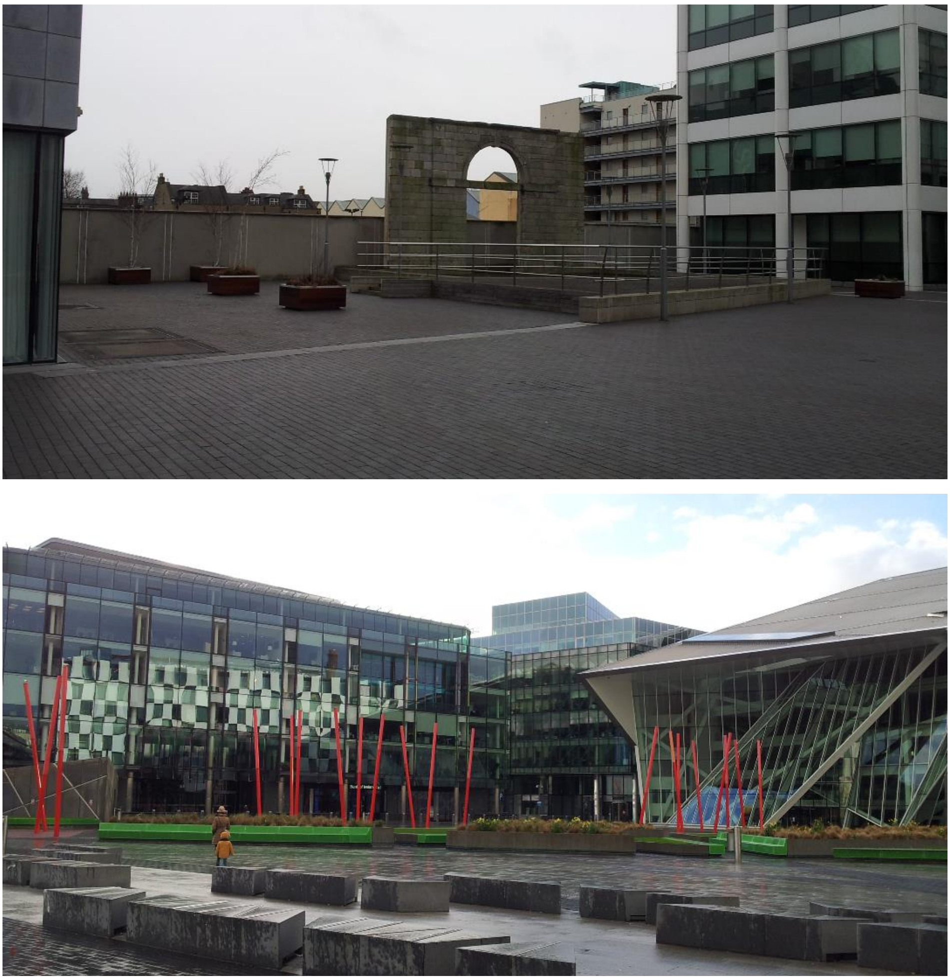


Figure 17. THE VOID property example

Pronk et al. (2008) note that void is used to develop activity by the usage of empty space. Surrounding structures form inner spaces and the absence of objects result in a space with the opportunity to develop activities. Void is necessary in every building, it results in a place of rest and calm. The surrounding fabric determines the feeling of the void, yet it is important that the void has enough space to exist and to be used. It is impossible to relate this property to a mathematical aspect.

NOT-SEPARATENESS. This property defines the way the life and strength of a center is merged smoothly — sometimes even indistinguishably — with the centers that form its surroundings (Alexander, 2002).

We illustrate this property in “Figure 18. NOT-SEPARATENESS property example”. Alexander considers this the most important property of all properties; the object cannot have life when it is not unified with its surroundings, even though it should contain the other fourteen properties in it. Lenovo/IBM TrackPoint pointing stick and “mouse buttons” below the space bar form a non-separable whole with Thinkpad keyboard, an integral part of the computer experience for many people (also used in creating this thesis). The manufacturer, perhaps influenced by image-driven minimalist designs of competitors, has tried to remove parts of the whole, but feedback from irritated users has forced them return to original design.

Figure 18. NOT-SEPARATENESS property example

Pronk et al. (2008) claim that this property cannot be defined in an objective way.

Additional properties. Not many suggestions for additional properties have been made. Quillien et al. (2009) propose to include WEAK LINKS (which complex stable systems can reorganize when strong links fail), SMALL WORLDNESS (combines the benefits of close neighbors with the convenience of long-range contacts, optimizing group access to resources), and NESTEDNESS (allows a complete network to look like a single leaf from the top network’s perspective). We limit our coverage to original fifteen properties, as we are more interested in applicability of structural properties than about the question of completeness of a structural property set

1.3.3 Property Interconnectedness

Structural property examples in the previous point were selected to highlight a property under consideration. This does not mean that the multiple properties cannot be present at the same time; quite the opposite is likely to be the actual case. Even in our examples, we could have designated some of the pictures differently. For example, the traditional bridge in "Figure 15. GOOD-SHAPE property example" does not only possess GOOD-SHAPE property, but is also a STRONG-CENTER.

It might also be difficult to differentiate the properties. For example, in “Figure 13. DEEP-INTERLOCK AND AMBIGUITY property example” the better balcony was chosen to exemplify DEEP-INTERLOCK AND AMBIGUITY; for the same reasons we might have used it to exemplify THICK-BORDER between the hotel room and the street. Therefore, it might be useful to cluster the properties, as attempted by Waguespack (2010) and shown in figure “Figure 20. Overview of Thriving Systems Theory”.

Table 3. Fundamental property interaction matrix

● indicates that property A depends on property B or we need property B for a complete understanding of property A		Property B														
		STRONG-CENTER	THICK-BOUNDARY	LEVELS-OF-SCALE	ALTERNATING-REPETITION	LOCAL-SYMMETRIES	POSITIVE-SPACE	ROUGHNESS	GRADIENTS	CONTRAST	DEEP-INTERLOCK AND AMBIGUITY	ECHOES	GOOD-SHAPE	INNER-CALM	THE VOID	NOT-SEPARATENESS
Property A	STRONG-CENTER	○			●	●			●	●					●	●
	THICK-BOUNDARY	●	○		●	●			●	●	●					
	LEVELS-OF-SCALE	●	●	○						●			●			
	ALTERNATING-REP…	●			○		●			●	●		●			●
	LOCAL-SYMMETRIES			●		○	●			●					●	
	POSITIVE-SPACE	●	●	●		●	○	●		●			●		●	
	ROUGHNESS	●					●	○	●				●	●		●
	GRADIENTS	●		●		●		●	○	●		●				●
	CONTRAST		●				●		●	○	●				●	●
	DEEP-INTERLOCK …				●		●	●		●	○	●				●
	ECHOES			●		●		●	●			○	●			●
	GOOD-SHAPE	●		●			●		●		●	●	●	●		
	INNER-CALM					●						●	●	○	●	●
	THE VOID		●	●		●	●			●				●	○	
	NOT-SEPARATENESS		●				●	●	●		●			●	●	○

Alexander (2002:238) has built a property interaction matrix which is adapted here as “Table 3. Fundamental property interaction matrix”. As we can see, the interactions/dependencies are not symmetrical (e.g. THICK-BOUNDARY is dependent on STRONG-CENTER, but there is no dependency the other way around). There is also an instance of self-dependency in case of GOOD-SHAPE

1.3.4 Structure Transformation

The structural properties of the living structure are not only useful analysis tools, but form a basis for increasing system liveliness by using a method which Alexander (2002a) calls structure-preserving transformations (SPT) or wholeness-extending transformations (WET). These are sequences of the properties added to the structure for extending wholeness with every step.

The process is a key to increasing liveliness of systems as living systems are not fabricated but generated. Complex adaptations in generated structure take place as part of differentiation. This is a process of dividing and differentiating a whole to get the parts, rather than adding parts together to get a whole; the generative process is top-down rather than bottom-up approach. Existing centers strengthened by adding new centers, by developing centers into more complex centers or by removing dysfunctional centers. We illustrate this in “Figure 19. Wholeness-extending transformation sequence example”.

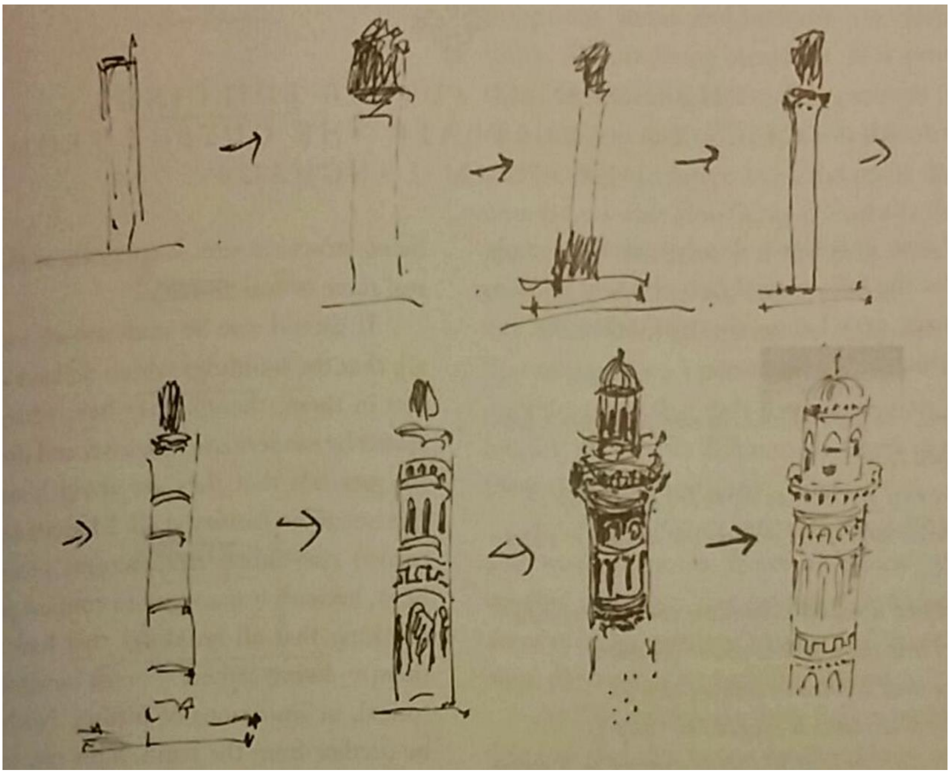


Figure 19. Wholeness-extending transformation sequence example

❖ Unfolding steps (replace figure and sequence below with own example)

Select next step which extents wholeness by following the least action principle:

1) First, we give the “stick” a top (this is the BOUNDARY transformation)
2) Next we give it a base (another BOUNDARY transformation)
3) The shaft is now given a good shape (by the STRONG-CENTER and GOOD-SPACE transformations)
4) …

We can present a generalized sequence for unfolding the wholeness:

1) Identify centers
2) Apply sequences of patterns (properties instead?)
3) Systematically remove antipatterns (negative properties)
4) Assess liveliness
5) Stop if alive enough (never?) else back to step 1

Even the most alive system systems have a tendency to break appart in time without extra care. Alexander has not dealt much with the liveliness maintenance problem. He mentiones though that death of entity might extend wholeness of the overall system (Alexander, 2002a:108). Opinions differ on necessity of dying, its importance in evolutionary adaptations and in adding value to life, both for humans and their creations like enterprises (de Geus, 1997). Disappearance may cause significant economical damage, loss of knowledge, and suffering

1.3.5 Formation of the Self

Perhaps the most controversial part, but also the culmination of Alexander’s argument, is reliance on the role of *self* in liveliness assessment. Firstly, sensing of life happens wherever the interaction of the centers resonates with the self of the observer. On the other hand, truly living structure must have a self on its own. Therefore, in generating the living structure:

> For every artist, for every builder, this must be true: as I work I must try to create a structure which appears like I to me (Alexander, 2004:70).

An individual life-form or organism is essentially a pattern, which should not change too drastically in order to preserve the identity and to be considered the same thing (Goertzel, 2006). Somehow the patterns and related sense of uniquness must match. Explaining this is rather difficult, but experiencing flow, growth, or beauty is not that much. Alexander summarizes this by:

> In order to create living structure, we must please ourselves (Alexander, 2004:272).

The creation of artifacts is usually a collective effort. Naturally, the concept of self is attributable to groups too. DeMarco & Lister (1999) have studied well performing software development teams. According to their observations, these are groups of people so strongly knit that the whole is greater than the sum of the parts. The teams are usually marked by a strong sense of identity. Finally, the goals of individuals have all bought on to the common goal.

Self and self-awareness in artifacts is nowadays mostly discussed in the artificial intelligence context. This should not prevent us from looking for the self in more ordinary objects. Particularly interesting aspect would be the coevolution of a person and an artifact

1.4 Infotechnological Applications

In some sense, wholistic methods described earlier are just beginning to be explored in computer science and information technology fields. There are some attempts to generate Alexandrian structural properties algorithmically, e.g. by Hoverd & Stepney (2010). Perhaps the most complete coverage can be found in Waguespack (2010). On the other hand, a subset of methods, design patterns, is heavily utilized and even extended to other activities, for example to analysis. We will present a brief overview of the latter as well

1.4.1 Thriving Systems Theory

Design of information systems is a social activity with all related consequences. As DeMarco & Lister (1999:36) stated in the context of information systems development: “twentieth century psychological theory holds that man's character is dominated by a small number of basic instincts: survival, self-esteem, reproduction, territory, and so forth. These are built directly into the brain's firmware”. This includes more or less gradual growth towards greater fitness in surrounding environment. There is also a question of direction of such development and about the meaning of existence. For an individual, this is closely related to the question of freedom of choice, free will and the role of consciousness; subjectively the more freedom means more life.

Life domain based metaphors are often used by all kind of system builders. Phrases like ‘the system has gone live’ or terms like ‘deadline’ are quite common in their language. We may intuitively assume that it has especially high potential in enterprise information systems domain. Firstly, living beings possess many characteristics attributable to enterprises or desirable for information systems, like adaptivity or robustness. In addition, the ability to distinguish living from non-living seems natural to human mind, thus the life metaphor can facilitate common understanding between different stakeholders and concentration on important things. As metaphor has to be known well for being useful, therefore, relative popularity of life sciences at the moment is additional reason for exploring possibilities for using the life metaphor.

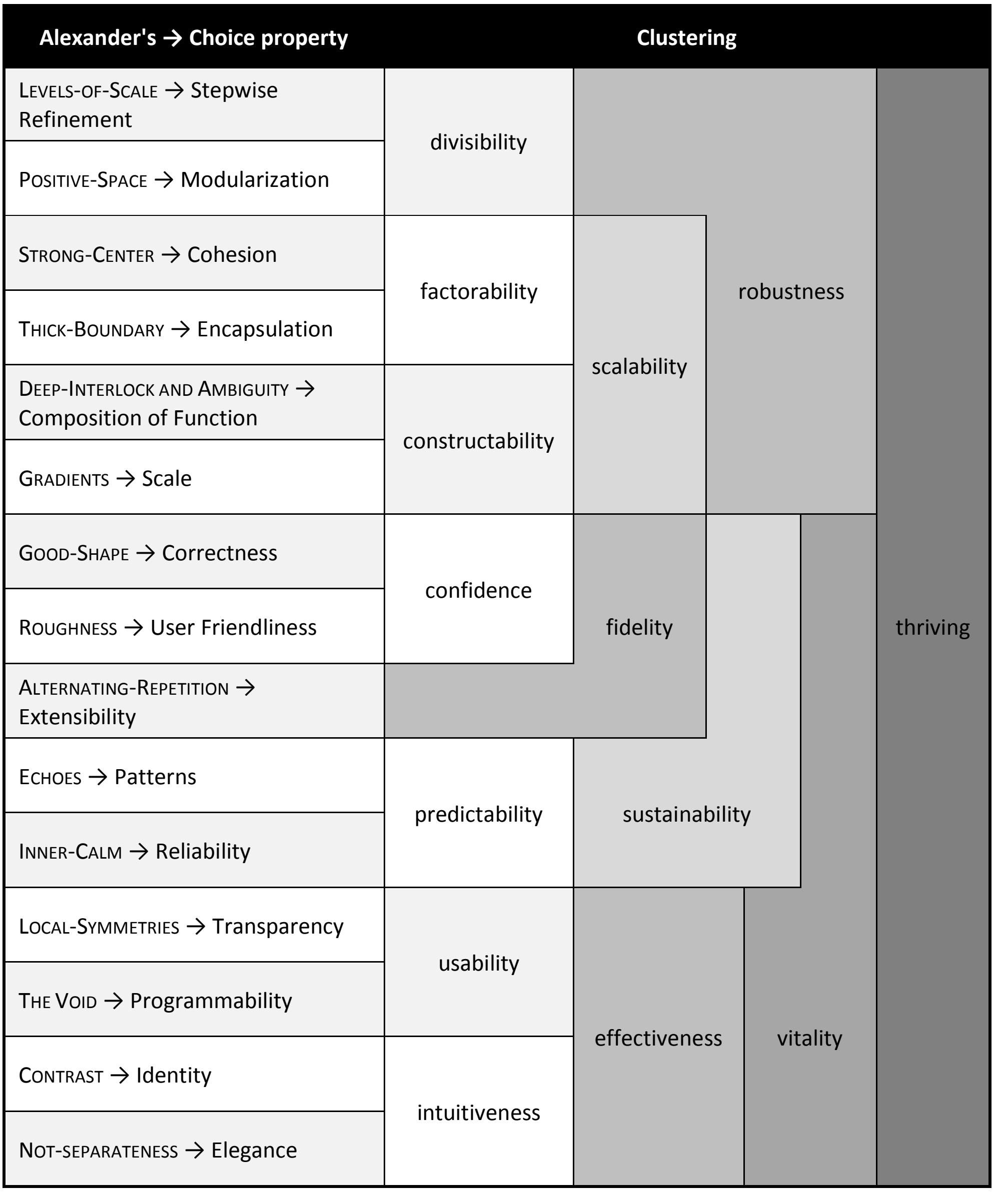


Figure 20. Overview of Thriving Systems Theory

Despite of that, there are only few academic attempts to tackle its utilization in engineering (e.g. article by Cowan et al. (2006)) or in information systems development in particular (e.g. article by McDavid (2005)). To date, perhaps a monograph by Waguespack (2010) is the most comprehensive study of this possibility. Thriving Systems Theory (TST) presented there is essentially an attempt to apply Alexander's latest thinking in information systems domain. As information systems are somewhat less tangible than the objects Alexander has been concerned with, in TST centers are replaced with more abstract choices made in development and properties of the living structure with so called choice properties respectively. Every center property is mapped to exactly one choice property, although sometimes quite serious approximations were needed. Because of interdependency between the properties, they can be grouped together by using cluster analysis technique. For example, Stepwise Refinement and Modularization as the closest properties can be grouped together to Divisibility property, which combined with Scalability are forming Robustness cluster, which finally together with Vitality forms a Thriving whole (see “Figure 20. Overview of Thriving Systems Theory”)

1.4.2 Design (and other) Patterns

Prominent role in information systems development belongs to design patterns. Original design pattern (Alexander et al., 1977) concept was introduced for urban planning and buildings, but became more popular in software engineering, especially in object-oriented programming. Some well-known attempts for utilizing design patterns in various software engineering areas are listed below:

- object-oriented design (Gamma et al., 1995);
- object-oriented analysis (Fowler, 1996);
- data modeling (Hay, 1995; Hay, 2006);
- software configuration management (Brown et al., 1999);
- project management (Brown et al., 2000);
- business modeling (Penker & Eriksson, 2000);
- development organizations (Coplien & Harrison, 2004);
- computer game development (Nystrom, 2014).

According to Alexander, the patterns were expected to express collective knowledge and are not so much based on formal proof on why they are working, but on common sense and shared vocabulary. Alexander (1979) thought difficulty for finding a real comparable to anything in theoretical physics, but once determined it could be reused efficiently by anybody from novices to masters. Respectively their application is more like an activity based on common sense and a shared vocabulary. Also an opposite concept of antipatterns is used, e.g. by Brown et al. (1998; 1999; 2000) and Laplante & Neill (2006). Many software houses and academic institutions are still publishing papers on and with patterns.

We present a sample design pattern in “Figure 21. INTIMACY GRADIENT pattern”. This pattern from (Alexander et al., 1977) deals with the problem/conflict:

> Unless the spaces in a building are arranged in a sequence which corresponds to their degrees of privateness, the visits made by strangers, friends, guests, clients, family, will always be a little awkward.

The solution/resolution to the problem:

Lay out the spaces of a building so that they create a sequence which begins with the entrance and the most public parts of the building, then leads into the slightly more private areas, and finally to the most private domains.

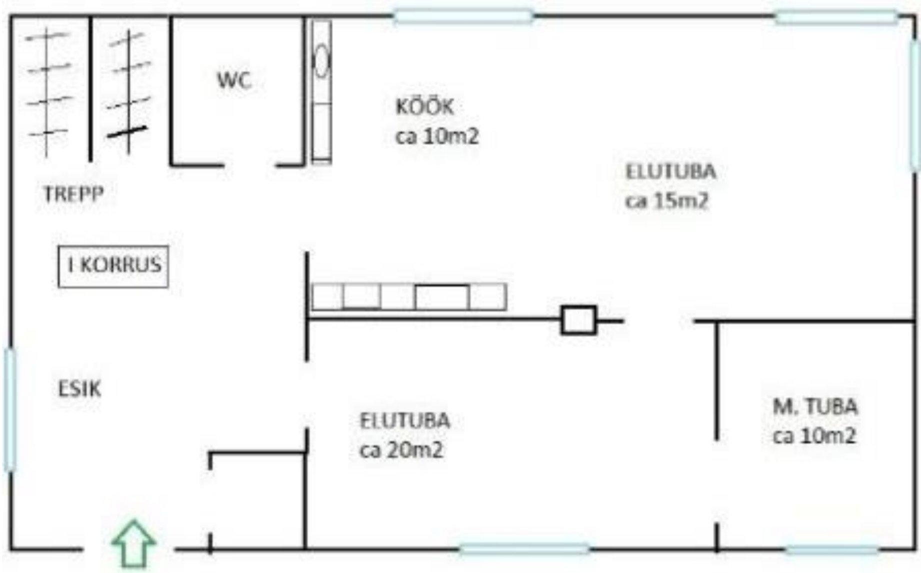


Figure 21. I*NTIMACY* G*RADIENT pattern*

Patterns dealing with problems in particular area can be arranged to a *pattern language*. This metaphoric connection to natural language domain indicates that patterns are expected to work together and their application to follow a certain order. We illustrate the structure of a possible (very high level) pattern language in "Figure 22. Data analysis pattern language".

Later theories of Alexander (2002a) see design patterns as stereotypical center configurations, special cases of structural properties. Generative sequences can be considered a second generation pattern language

1.4.3 Contemporary Art and Science of Data Analysis

The system development methods commonly promote importance of having some kind of generic structure or principles in place. This so called architecture should form a basis for detailed design and finally for implementation. By McDavid (2005) "an architectural approach emphasizes the need for multiple levels of abstraction, standardized interfaces offered by well-defined modules, encapsulation, information hiding, and the like. This allows specialists to focus on their parts of the problem, and to meet at well-defined interfaces, but to understand the other levels of the problem that provide a context for their work. This doesn't mean the problem goes away, but that there is a principled approach to address it." Spinellis et al. (2009) have collected several architectural principles from various sources:

- *Versatility*. Offer “good enough” mechanisms to address a variety of problems with an economy of expression;
- *Conceptual integrity*. Offer a single, optimal, nonredundant way for expressing the solution of a set of similar problems;
- *Independently changeable*. Keep its elements isolated so as to minimize the number of changes required to accommodate changes;
- *Automatic propagation*. Maintain consistency and correctness, by propagating changes in data or behavior across modules;
- *Buildability*. Guide the software’s consistent and correct construction;
- *Growth accommodation*. Cater for likely growth;
- *Entropy resistance*. Maintain order by accommodating, constraining, and isolating the effects of changes.

According to Parnas (1998), there is a subtle distinction between computer science and more practice-oriented software engineering. Many decision-makers in EIS field have computer science degree and their thinking is influenced by current scientific paradigms and objectivist worldview. In this context this might be reflected in issues like overspecialization, too simplistic explanations to problems and insufficient creativity. We may say, that many of the problems can be attributed to ignoring the question of value — an attitude in science which Maxwell (2004) has metaphorically called “neurotic”. Boehm (cf. Biffl et al., 2006:24), one of the main originators of Value-Based Software Engineering (VBSE), has noted that “much of current software engineering practice and research is done in a value-neutral setting, in which:

- Every requirement, use case, object, test case, and defect is treated as equally important;
- Methods are presented and practiced as largely logical activities involving
- mappings and transformations (e.g., object-oriented development);
- “Earned value” systems track project cost and schedule, not stakeholder or business value;
- A “separation of concerns” is practiced, in which the responsibility of software engineers is confined to turning software requirements into verified code.”

Even if slightly exaggerated, the points listed above are very common in software engineering practice. VBSE aims to improve the situation by connecting traditionally value-neutral computer science with value-based theories, in particular with utility theory, decision theory, dependency theory, and control theory. In such 4+1 architecture a central positions belongs to theory W, also known as the Enterprise Success Theorem, which is stating that “Your enterprise will succeed if and only if it makes winners of your success-critical stakeholders”.

Modern data analysis (recent years broader term *data science* is frequently used) relies heavily on computation and inherits many approaches from software engineering. Of course, it has its own specifics and connections to other domains (e.g. statistics) as well. We have summarized common themes identified during literature review in form of a pattern language below. Basic form of pattern consisting of the name, the context and the problem, the solution (introduced by the word *therefore*), and the source is used below for capturing a pattern. First three are the essential parts of the pattern. Frequently additional elements have been found necessary. We have included the sources of a pattern from those.

EPICYCLES. Data analysis is difficult to formalize artful process. Even good analysts are not always able to explain how they achive the results. Straight path from question to answer is possible only in the simplest cases.

Therefore, organize the analysis process cyclically or accept its cyclical nature in self-organization.

Iterative approach is widely known, though perhaps (formally) followed less than would be beneficial. Peng & Matsui (2015) have popularized the term *epicycles* in data analysis context.

GOOD QUESTION. Every analysis should answer a question. Unfortunately, the latter is not always clearly stated or well targeted.

Therefore, constantly refine the question during the analysis. Good questions are: 1) of interest to the audience; 2) not answered before; 3) grounded in a plausible framework; 4) anwerable in principle; and 5) specific.

Peng & Matsui (2015) have discussed the importance of having a good question and provided the list of characteristics of good questions above.

TIDY DATA. Data can exist in various formats, locations, may change unexpectedly, etc. The process of cleaning and preparing the data frequently requires more time and effort than the actual data analysis itself. On the other hand, running analysis on a messy data makes interpretation of results difficult.

Therefore, preprocess a snapshot of data before starting with inference or predictions. After preprocessing: 1) each variable forms a column; 2) each observation forms a row; and 3) each type of observational unit forms a table. The data will be in (interchangeable) long or wide table form.

Wickham (2014) terminology was used here.

EXPLORATORY ANALYSIS. The analyst might not be familiar with the data and very likely the same applies to the audience as well.

Therefore, perform and report back a descriptive analysis of existing data in the form of summary statistics and plots.

Peng & Matsui (2015) includes comprehensive description of the explanatory analysis.

INFERENCE. Usually we are not only interested in present (observed sample) data but also from the data not present (population data). We suppose association between the limited number of predictors and the outcome.

Therefore, choose statistical inference methods from wide body of available methods, according to amount and nature of data. In some cases might be also possible to explain the mechanism causing the association.

Statistical methods have been around for quite some time.

PREDICTION. We might be not that much interested in association between a particular predictor and outcome but are trying to identify a model which best predicts the outcome. There might be many predictors and understanding how the model works is considered unimportant.

Therefore, choose appropriate machine learning or data mining algorithms for analysis.

Wide body of data mining and machine learning methods exists.

MECHANISM. Context. Problem.

Therefore, solution.

Peng & Matsui (2015) discuss this.

ROUTINE COMMUNICATION. Analysis is frequently a social construct. This is particularly true in commercial or government settings. The question and possible data sources may become clear only in the middle of analysis process.

Therefore, a lot of effort must be put into frequent communication. Both informal (e.g. clarifications and early feedback) and more formal (e.g. concluding presentation) communication have their place in the process.

Peng & Matsui (2015) discuss this extensively.

REPRODUCIBLE RESEARCH. Human readable texts and visualizations are difficult for machine processing. On the other hand, source code might be not easy to understand without explanations. Without looking at the data, the whole analysis might be questionable.

Therefore, publish the results of analysis by including not only human readable explanations but also the data and the program code, so that other people or subsequent analysis iterations can verify the findings and build upon them.

Sources can be traced back to Knuth (1984).

In “Figure 22. Data analysis pattern language” relationships between patterns are visualized. Language syntax is limited to sequencing.

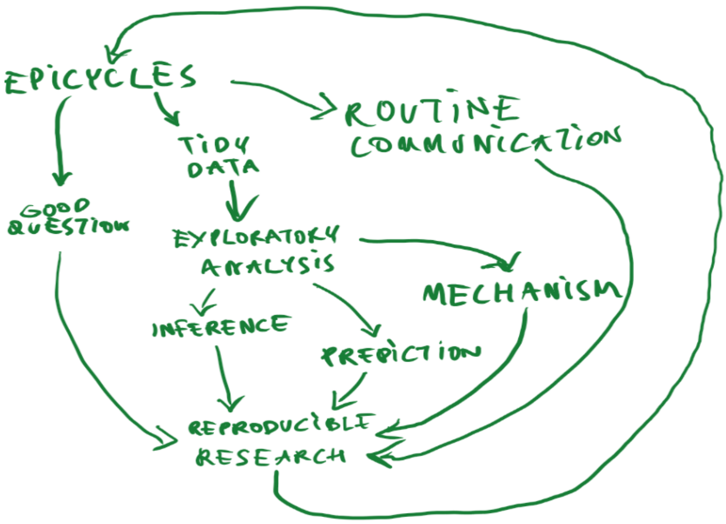


Figure 22. Data analysis pattern language

2. CASE STUDY

This chapter presents a selection of real-world problems from the financial services domain. These will be used later in explaining the proposed solutions

2.1 Sample Domain

The reason for financial services domain selection is author's involvement in related software projects. The aim of this thesis is neither to provide a comprehensive overview of the financial systems domain nor to contribute to improving of financial services in any other way than via betterment of supporting information technology services. Nevertheless, to understand the influence of the particular surrounding environment to information system projects, some specifics of the domain have to be explained

2.1.1 Financial Sector Specifics

Financial sector has a prominent role in our society. As an example, the financial value only of the derivatives market is bigger than the rest of the world economy. On the other hand, compared to domains like aviation or nuclear energetics, consequences of failure, though still causing significant suffering, are less dramatic. Also, though high-frequency trading, fraud detection, etc. algorithms can become quite esoteric, most of everyday banking operations are more mundane. Finally, there are many procedural safety systems in place, making systems development more relaxed.

Culture is considered an essential aspect of modern organizations. It can be studied by looking at the related artifacts, of what impressive buildings in central locations are most visible. Development centers are usually located in more cost-effective areas and are not particularly suitable for creative work. As illustrated by photos taken in South-East Europe and South-East Asia in "Figure 23. Case study customer site examples", grey tonality is prevalent; offices are equipped with anonymous furniture, and emotion-less technology. The employees are exposed to all kind of disruptions, which are usually fought by listening to music or obsessive web surfing. Seldom items indicating passion towards one's profession can be found; instead printouts of cartoons and other satirical representations of office work reality are common.

There are certainly many variations inside the financial sector too; even between branches of the same institution. These variations are dependent on regional (e.g. 'Swiss Bank') or ideological (e.g. 'Islamic banking') differences, the markets where the institution is operating (e.g. retail or investment banking, life insurance, etc.), business volumes and other aspects, with not least importance the direction related people have decided to follow. This makes reusing of solutions from one institution in another complicated.

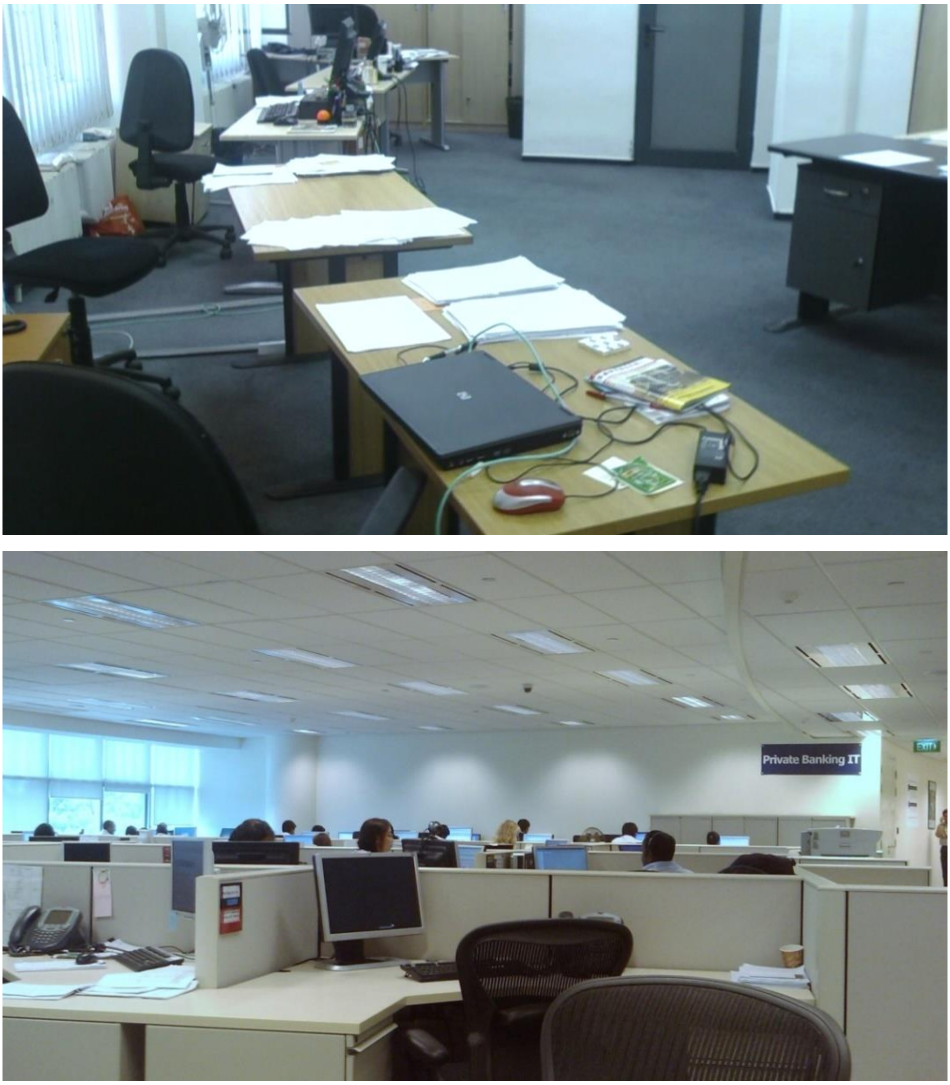


Figure 23. Case study customer site examples

❖ Room for analysis like presented in this thesis

- Possibilities for recovering the lost spirituality
- Needs of travelling contractors in modern job market

2.1.2 Core Banking Systems

There is a need to build integrated software systems for supporting basic banking operations. It is customary to call these *core banking systems*. Such systems can be built in-house from scratch or nowadays these are more commonly based on packaged software solutions.

Packaged software implementation is a highly practical matter, without deep theoretical background, although several comprehensive guidebooks are written, like Tayntor (2005). The term *packaged software* is typically, and also in this thesis, used for referring to upscale ready-made enterprise application suites, rather than to so called shrink wrap packages sold through retail outlets or via network delivery. Packaged software usage is very common nowadays; in comparison with in-house built systems it is expected to reduce costs, shorten implementation time, lower the risks and import new knowledge to the organization. Unfortunately, there are many steps which can go wrong: vendor selection, analysis of gaps between existing practice and functionality of the new software, legal issues, understanding of requirements, post-implementation maintenance, etc., resulting in the opposite outcome.

From the vendor or solution supplier viewpoint, the situation may look different. As packaged software implementations typically require from months to years of configuration, customization and integration work, to set the system up for the specific needs of each individual business, they may expect steady income for a longer period, but instead they will find themselves facing unrealistic expectations, resistance inside the customer organization and disputes over the costs.

Software vendors frequently have collected their experiences with implementing the particular product and are offering this as a recommended methodology for implementing the system in the customer organization. Alternatively, methods standardized within the organization or understanding of industry best practices might be attempted for improving the outcome of implementation projects; complexity needs to be managed even in low-tech implementations and standards used for facilitating mutual understanding. There is a classic waterfall approach for dividing the work into phases like requirements analysis, system design, testing, etc. Some other methodologies organize these steps into smaller iterations (e.g. Unified Process) or aim to constantly improve the development organization’s ability to deliver the results (e.g. CMMi). Such “big methodologies” are frequently criticized (e.g. by DeMarco & Lister, 1999) and lightweight, more social aspects oriented, “agile methods” proposed instead (like XP, Scrum, etc.). Methodological ideas may come also from management theories, like business engineering or reengineering.

❖ Brief overview of related projects

In practice the transformational activities tend to be focused on bigger changes, organized around projects and development programmes. After achieving related higher-level goals, we should find the enterprise in the next stable state. Goals are converted into requirements for information system (or software) development process, usually with more concrete kind of organization. The latter roughly assumes either predictable or chaotic environment, proposing correspondingly either machinelike waterfall or more flexible agile approaches, without much analysis like proposed here for identifying the situation.

The issues tackled in this thesis have been experienced by the author in several core banking systems projects from 15 customer sites in the last decade. Unfortunately, precise quantitative data has not been always available. But in every sample project at least some of the generic problems listed below were experienced:

- inability to go live: the project was cancelled and new core system not deployed (projects E, G, H, I);
- classical success criteria of "within time, within scope, and within budget" was not met (D, F, J, L, M) or appeared insufficient later (B, C, K);
- difficulties with upgrades: subsequent releases of the software appeared very hard to deploy (A, B, C);
- infrequent reuse of developments: most projects were preceded or expected to succeed by similar projects in other branches of the bank (except G, J), but only in few instances significant reuse of customizations (and only at most by 50%) took place;
- stressed stakeholders (E, I), in extreme cases medical help was required (C);
- no noticeable learning from previous experiences.

Taylorist separation of planning/doing and specialization is presented in "Table 4. Role separation in core banking system projects". From requirements (supposedly originating from goals of the institution) to implemented software

Table 4. Role separation in core banking system projects

Role	**Description**
Program manager	
Project manager	Frequently more than one per project
Business analyst	Communicates requirements and performs functional parameterization
Technical consultant	
Test analyst	
System administrator	Perform technical parameterization

Architects, techno-functional consultants, management consultants, project office workers, etc. can be encountered as well.

❖ Business aspects

Projects in financial sector usually have sufficient funding. Naturally, a lot of effort is put on tracking spending, with related positive and negative consequences. Exposure to vast amounts of money has additional side-effects: bankers are known for their conservative dress code and high staff turnover rates. Their public image is not always positive.

- Different stakeholder interests, long-term viability seldom a concern
- Banking system dev seems not to value structure preservation much

2.1.3 <u>Temenos T24 Overview</u>

Brief overview of Temenos T24 Core banking system

- ❖ Functional aspects
  - Good customization options
  - Data spread over around 3000-4000 system tables and unlimited number of locally created tables
- ❖ Technical aspects
  - Flexibility in sake of performance
  - Key-record structure
  - Multi-value data model (decreases number of physical tables)
- ❖ LD & AA overview

2.2 Data Generation

Generate sample data for analysis

- ❖ Scripts for test data generation

Three generic cases nick-named as good, bad, and ugly customer loans

- ❖ Running generation scripts for T24 record generation

Repeat generation n times and note down elapsed time to understand structure and performance relation?

- ❖ Service the loans till maturation
  - Capture the loan
    - 2 month annuity loan
    - Separate disbursement and repayment accounts
  - Upfront fee deduction
  - Disbursement
  - Repayment 1
    - Half of customers with no money to repay
  - Full repayment
  - Past due repayment
  - Closure
    - Fully repaid
    - Written off

2.3 Data Collection

Collect and transform the generated data

- ❖ Collection points

Data must be collected in relevant times during the loan lifecycle

2.3.1 Structure Extraction

- Recursive extraction starting from origin
- Extraction from references to origin
- Find relations by scanning whole database for contract references

2.3.2 Data Extraction

- Extract summary data based on extracted structure

```
#Load observations file
obs <- read.csv('data/obs.csv', stringsAsFactors=FALSE)

#Observations data frame structure
str(obs, width=60, strict.width='cut')

## 'data.frame':    48891 obs. of  4 variables:
##  $ dtstamp : int  20160308 20160308 20160308 20160308 2016..
##  $ object  : chr  "SC.CASH.FLOW11" "SC.CASH.FLOW11" "SC.C"..
##  $ property: chr  "field" "field" "field" "field" ...
##  $ value   : chr  "@ID" "SECURITY.ACC" "SUB.ASSET.TYPE" ""..

#Summary of object properties observed
table(obs$property)

##
##   field linkdef
##   43805    5086

#Sample observations of each property
t(sapply(unique(obs$property), function(x) {
  sobs <- obs[obs$property==x,]; sobs[1, ] }))

##         dtstamp  object                            property  value
## field   20160308 "SC.CASH.FLOW11"                  "field"   "@ID"
## linkdef 20160308 "LD.RECEIPT.ENTRY>CURRENCY.CODE" "linkdef"  "CURRENCY"
```

3. PROBLEM STATEMENT

This dividing chapter summarizes the limitations of existing methods covered in previous chapters. Grounding principles for alternative approach will be presented. These will be further elaborated into corresponding solutions in subsequent chapters

3.1 Summary of Issues

Several issues were reported in regards to case study projects. There are multiple possible reasons for such problems. Unrealistic goals might have been set, which were impossible to achieve with available resources and within expected timeframe. Execution of the project could have been problematic or risks (bad luck) managed poorly. Of course, similar issues with information systems development projects in general are frequently reported in literature. Encountered problems are not specific only to those institutions and particular software implementation projects, but are variations of classic complex questions from enterprise information systems development domain:

- Why so many information system projects fail?
- How to measure the success of an enterprise/information systems architect or designer?
- Why experienced developers and enterprises are not learning from failures?

It is reasonable to assume that satisfactory answers can be found to such questions. Even among sample projects, which in general cannot be considered very successful, are some examples of better performing projects. For example, in some projects wide range and complex functionality was introduced while leaving the practical implementation work to modestly paid local developers, who learned specifics of a particular software system by doing and also whose banking knowledge level was usually below the financially significantly better rewarded external consultants of other projects. In some other projects technically capable and well maintainable core system was built, which really supported the expansion of customer business opportunities and was able to grow together with increased volumes. Sometimes experienced consultants were present, who could after a short period of time estimate quite precisely the outcome of the projects, based on mostly on intuition. This should make us optimistic about possibilities for adding predictability to such efforts. So far seems that development methods are chosen based on personal preferences or company traditions, not based on suitability.

This thesis does not attempt to deal with all the possible problems listed above. To maintain concreteness, it concentrates only a particular subset of issues. However, many explanations and solutions are applicable to other development and analysis activities. Using for the case study packaged banking software implementation specifics should not limit generalization possibilies drastically either

3.2 Current Limitations

In case study projects very little relience on academic (scientific, professional) knowledge was encountered. These activities were not considered art either. Not much seems to have changed in decades past since this remark:

> The statistics about reading are particularly discouraging: The average software developer, for example, doesn't own a single book on the subject of his or her work, and hasn't ever read one. That fact is horrifying for anyone concerned about the quality of work in the field (DeMarco & Lister, 1999:29).

Such a wide range of issues is difficult to cover otherwise than metaphorically. Currently prevalent mechanistic approaches seem to be inadequate. This thesis is proposing that the life metaphor is particularly useful for explaining success and failure of information systems development projects. It might be especially useful for analyzing the surrounding environment of the enterprise and adjusting the development requirements and methods based on that. It can be also helpful in managing the complexity and for reducing specialization gaps between experts from various areas by providing a consolidated focal point. Ultimately we may expect people involved in such problems feel more content and their personal goals aligned with those of the enterprise.

To utilize the promising benefits of the life metaphor by connecting it with existing information systems development theories, extra efforts are needed. So far only few attempts have been made for utilizing living systems perspective in engineering, rare examples are articles by McDavid (2005) and Cowan (1999), or monograph by Waguespack (2010). Surroundings and organizational culture are important building blocks of the living enterprise. This helps to explain the raise of design patterns movement and why they may be going out of fashion. It is still uncertain if we are culturally ready to grasp most recent ideas of Alexander, which are even more unorthodox than the thinking which led to widespread recognition of design patterns in software development.

Unfortunately, most pattern implementations have concentrated on utilitarian value of the individual pattern as “a solution to a problem in a context”, by omitting most of Alexander’s philosophy (sometimes explicitely, like by Fowler (1996:5)), generative power of pattern languages and thus possibility for explaining how true masters are working. This might be one of the reasons, why design patterns are criticized in recent times by acclaimed programmers (Seibel, 2009) and there seems to be little progress in this field in recent years

3.3 Research Question

This thesis is dealing with the following questions:

- Objective detectability of structural viability;
- Applicability of structural viability in data analysis.

Earlier, we attempted to show the potential of structural viability theory in domains like presented by the case study. However, as explained in the previous point, current methods are still inadequate for practical application and require further elaboration with additional proofs.

The aim of this thesis is to demonstrate benefits of value and viability analysis in information systems development. Of course, there is a danger of sticking to crude analogies or mechanically copying observations from nature, but on the other hand, unnecessary formality may negatively affect the usability of the theory. Above all, it must support generation of better information systems. We may expect the main concepts of such theory to be easy to comprehend, presented compactly and referring as much as possible to already existing theories. Although generality was intended, also necessity for scaling down to somewhat low-tech situations as covered by case studies was kept in mind

3.4 Research Methodology

A quest to answer the research question is inspired by the following recommendation:

> At its best and most useful, science combines the arts of wholistic observation, reductionist observation, and experimentation in pursuit of human well-being (Campbell, 2013:215).

Comprehensive theories of living systems have been around for some time and experiences of their application are most valuable. Holmberg (2008) discusses several important points on an example of LST, what must be taken into account in proposing a new theory:

- *Isolationism*. Lack of theoretical and interventionist variety (LST is not in any significant degree related or positioned to other system theories);
- *Epistemology*. LST may be of good help for understanding the deterministic modalities but it has to be complemented with other approaches for grasping the normatively dominated modalities (e.g. creedal and ethical);
- *Emergence*. While e.g. Boulding (1961) tried to find new emerging properties at each new level, LST concentrates on things, which are preserved from one level to the next one;
- *Anticipation*. System’s ability to act beforehand in order to arrive in a better state at a later time, while the more common reactive behavior stands for acting afterwards in order to restore an earlier state;
- *Evolution*. Teachability, visibility, and methodology (LST provides a vast amount of knowledge concerning life and living systems but it says literally nothing about what to do with that knowledge).

He suggests that by adding observers, intentions, multiple views, relations of power and other inventions from later research and development the modeling tools LT would become even more powerful and comprehensive. Special attention should be put into relationship between living systems and artifacts.

To satisfy the above goals in this thesis, several assumptions are made, or in other words, axioms are postulated, as a basis for a new theory. These are listed below:

- the world where we live exists independently in itself;
- information systems development is a complex creative activity;
- our creations are dependent on understanding of the world;
- nobody understands all the aspects of a complex system;
- people understand things differently, but there are many commonalities;
- group creativity needs shared understandable metaphors;
- we cannot change people and social systems, only influence them;
- good metaphors have to be inspiring and constructive.

Based on these assumptions it is possible to specify characteristics, what the generic theory of life metaphor usage in information system development must possess, which can be adapted to specific needs and shared among stakeholders. We believe that not only the senior managers or high-end consultants have the power to influence the enterprise with their thinking, but all stakeholders:

> There may be some individuals whose images are of peculiar importance in an organization. … Nevertheless, in the dynamics of an organization all images are important and none can be neglected (Boulding, 1961:63).

We start looking for answers to the research question by formulating a corresponding analysis process. After that we present the results of a survey which was conducted to confirm both the groundings and outcomes of this work. Unfortunately, although case study examples were used both in the analysis process definition and in the confirmatory survey, we did not had an opportunity to conduct experiments for comparing approaches presented here with ordinary practice

4. ANALYSIS PROCESS

This chapter describes analysis of data structures according to theories discussed earlier

4.1 Framework

❖ Combine static and dynamic aspects, plus the self

$W = f(W_i, W_e) + \varepsilon$ (static)

$W_t \rightarrow W_{t+1}$ (dynamic)

4.1.1 Generic sequence

1) Preliminary steps
   - Question formulation
   - Data collection
2) Unfolding steps
3) Assess result and reinstiate if needed

❖ Human (wholistic) evaluation between steps

Fully executable end result, but analysis cannot be fully automated in principle

❖ Analysis object for sequence maintenance

id	parent	cmd	struct	timelineevent

Methods: new, print, add, remove

4.1.2 Structure as network (static)

❖ Horizontal/vertical organization

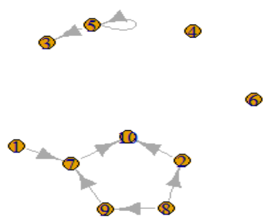
MODULE > TABLE > FIELD

❖ Adjacency matrix

❖ Observations narrow to wide format transformation for node attributes

```
library(igraph, quietly=TRUE)

adjm <- matrix(sample(0:1, 100, replace=TRUE, prob=c(0.9,0.1)), nc=10)
adjm

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    0    0    0    0    0    1    0    0     0
##  [2,]    0    0    0    0    0    0    0    0    0     1
##  [3,]    0    0    0    0    0    0    0    0    0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0
```

```
## [5,]    0    0    1    0    1    0    0    0    0     0
## [6,]    0    0    0    0    0    0    0    0    0     0
## [7,]    0    0    0    0    0    0    0    0    0     1
## [8,]    0    1    0    0    0    0    0    0    1     0
## [9,]    0    0    0    0    0    0    1    0    0     0
## [10,]   0    0    0    0    0    0    0    0    0     0

g1 <- graph.adjacency(adjm)
g1

## IGRAPH D--- 10 8 --
## + edges:
## [1] 1-> 7 2->10 5-> 3 5-> 5 7->10 8-> 2 8-> 9 9-> 7

plot(g1)
```



❖ Views of 1) context 2) entities and attributes 3) hierarchy?

Compact display possibilities

❖ Graphical representation of node and edge attributes

igraph capabilities: color, shape, etc.

4.1.3 Structure as lifeline (dynamic)

❖ Process has structure

Perhaps it is most intuitive to understand life as a process. This process is divided into stages which are separated by distinctive events:

- t_0: inception;
- t_1: birth;
- t_2: adulthood;
- t_3: decline;
- t_4: death. Decomposition starts;
- t_5: assimilation.
- t_6: disappearance.

To be alive is a process of going through different stages of life, in each having different levels of liveliness and relations with the environment. Other rhythms like circadian, etc.

❖ Co-evolution

We are particularly interested in directed coevolution. Information systems are expected to cease to exist as the enterprise discontinues, though common thinking is mostly concerned about the life of separate subsystems, which are likely to be decommissioned and replaced with new systems from time to time. Sometimes the subsystems begin their independent life as spin-off packaged software, as in case of software implemented in sample projects.

4.1.4 Self development

Life metaphor makes possible to conveniently introduce unusual topics like the joy of work into practices of development organizations

❖ Function vs intrinsic value

The concept of ‘intrinsic value’ is widely used in philosophy and also for describing certain financial instruments. However, it is generally not a part of common vocabulary in information systems development projects. Therefore, we will further explore possibilities of using alternative metaphorical ways for expressing the same, by also stressing in some ways subjective nature of value and importance of agency.

One of the universal characteristics of living entities is their operational closedness. In other words, we cannot change these entities, or at least make that a primary activity, but only influence them.

❖ Not to be confused with entity from ERD

Table 5. DATA ENTITY (ORGANIC) metaphor

Organism	**Entity**	**Explanation**

Sometimes we want in designing the systems to stress opposite characteristics, depending on the environmental conditions and our moral criteria, which makes characteristics attributable to lifeless, machine-like structures more favorable. Realistically our creations contain elements both for increasing and decreasing liveliness, thus making the whole more or less alive. Moreover, sometimes we may want to have a machine-like tool, shelter or enterprise, only to produce concrete results in a particular timeframe. Universality, self-organization, etc. characteristics in that case are only hurting the efficiency.

Table 6. DATA ENTITY (MECHANISTIC) metaphor

Machine	**Entity**	**Explanation**

This does not mean that organism and machine metaphors must exclude each other; on the contrary, they can complement each other in forming more complete image of the system. We may want the organization to act like machine, predictably and optimally

4.2 Property Generation

Abbreviations introduced for convenience

4.2.1 STRONG-CENTER [SCtr]

- Conceptual

Number of connections. Centraility measures

- Functions (fully in appendix)
- Generated examples

4.2.2 THICK-BOUNDARY [Bdry]

Interfacing

4.2.3 LEVELS-OF-SCALE [LoSc]

Hierarchical organization and column counts

4.2.4 ALTERNATING-REPETITION [AltR]

Entities with substantially different amounts of data alternating

4.2.5 LOCAL-SYMMETRIES [LSmt]

TBD

4.2.6 POSITIVE-SPACE [PSpc]

Table filling rate

4.2.7 ROUGHNESS [Rghn]

TBD

4.2.8 GRADIENT [Grdt]

TBD

4.2.9 CONTRAST [Cotr]

TBD

4.2.10 DEEP-INTERLOCK AND AMBIGUITY [DI&A]

Excistence of connecting entitities

4.2.11 ECHOES [Echo]

TBD

4.2.12 GOOD-SHAPE [GShp]

No anomalities in entity structure (root to shoot axis)

4.2.13 INNER-CALM [Simp]

Simplification possibilities

4.2.14 THE VOID [Void]

Removal transformation

4.2.15 Cluster Generation

- ❖ Alternative clustering to what proposed in TST
- ❖ From interaction matrix as directed network

4.3 Sequence Generation

- ❖ How to preserve a property from previous steps (not to remove)

Gene metaphor. As Boulding (1961:62) put it, “in organizations the image resides in the genotype, not in the phenotype”. What will be actually transferred: Artifacts, memes, identity, …

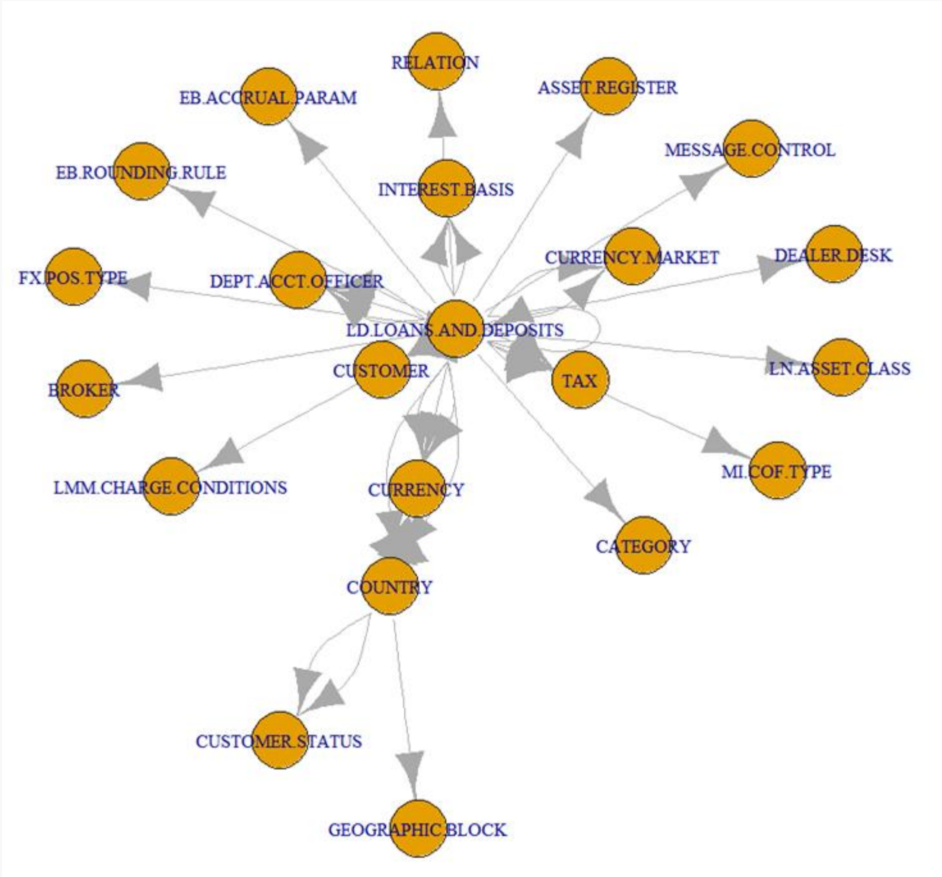
- ❖ Presentation from snapshots of static structures:
  - Animated gif using corresponding R library
  - Lattice for document
  - Markdown
- ❖ Simulation

4.4 Case Study Data Analysis

- ❖ Data transformation
- ❖ Running the analysis

```
#Common functions
source('func.R')

#Load observations file
obs <- read.csv('../data/obs.csv', stringsAsFactors=FALSE)

#Create analysis sequence list (perhaps a tree later)
ang <- make_empty_graph()
angList <- list(ang)

ang <- addCenter(ang, 'LD.LOANS.AND.DEPOSITS')
angList[[2]] <- ang

ang <- addCenter(ang, 'COUNTRY')
angList[[3]] <- ang

ang <- voidCenter(ang, 'BILL.REGISTER')
angList[[4]] <- ang
```

```r
plot(ang)
```



RELATION
EB.ACCRUAL.PARAM
ASSET.REGISTER
MESSAGE.CONTROL
EB.ROUNDING.RULE
INTEREST.BASIS
FX.POS.TYPE
DEPT.ACCT.OFFICER
CURRENCY.MARKET
DEALER.DESK
LD.LOANS.AND.DEPOSITS
BROKER
CUSTOMER
TAX
LN.ASSET.CLASS
LMM.CHARGE.CONDITIONS
CURRENCY
MI.COF.TYPE
CATEGORY
COUNTRY
CUSTOMER.STATUS
GEOGRAPHIC.BLOCK

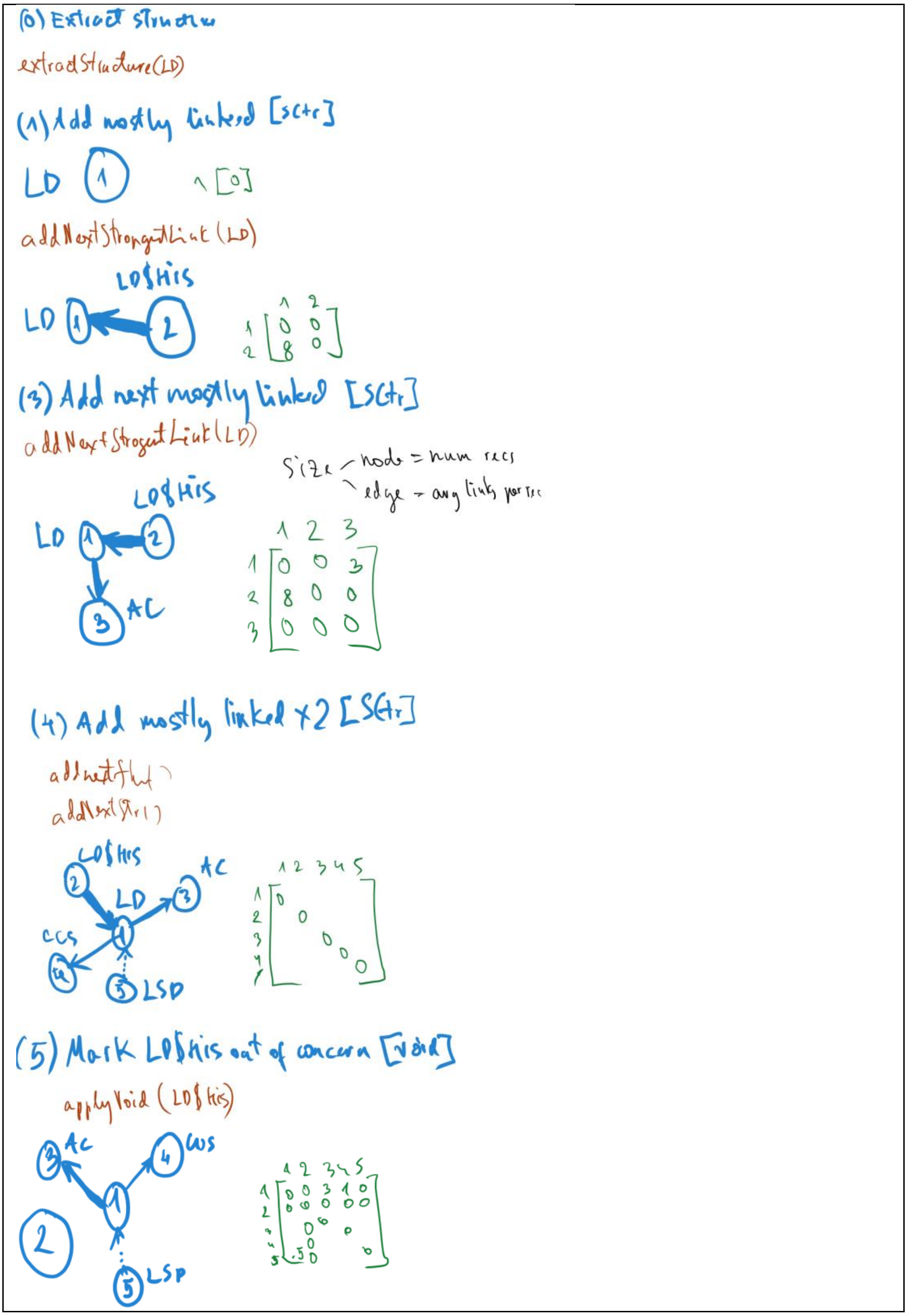


Figure 24. LD module analysis visualization

Figure 25. AA module analysis visualization

- Liveliness assessment

5. CONFIRMATORY SURVEY

This chapter describes design and conduction of the study for assessing liveliness of various structures discussed earlier

5.1 Survey Design

- Sampling size determination
- Alexander’s sample repeating (1x)
- Control question about familiarity with Alexander’s work

5.1.1 <u>Photo Evaluation</u>

Ask to assess liveliness of selected photos from overview section

5.1.2 <u>Narrative Evaluation</u>

- Positive vision (shorten)

> Somebody deeply involved feels the need for change. She convinces others on the same. Appropriate technology will be selected with a long-term view and partnership with technology company commenced. Development organization will be formed from balanced amount of carefully selected internal and external people, following the organizational culture with transparency, community of trust, commitment. Volatility of the environment and viability of the enterprise and its information system will be assessed. In accordance to the value analysis a development strategy will be elaborated for a particular case. It will be carried out by everybody giving their best in specialization area and contributing to the whole. After transformation the change will be evaluated and lessons learned from the experience

- Different vision (polish)

> Open the door to salesmen and let them entertain you. With other important people initiate the change. Look for cheapest resources available, keep them competing and tell them what they have to do. Keep good track of costs, let accountants and business people to decide. Expect a system to be delivered, which within next few years will justify the development costs. After few more years decommission the system and start all over again

5.1.3 <u>Abstract Structure Evaluation</u>

Abstract structures generated in previous chapter

5.1.4 <u>Abstract Sequence Evaluation</u>

Abstract sequence generated in previous chapter

5.1.5 <u>Case Study Analysis Evaluation</u>

- Comparing sequential description with traditional figure + text

5.2 Survey Conduction

❖ Randomized positions

Figure 26. Survey form

5.3 Survey Analysis

❖ Proof of the usefulness of the analysis

❖ Accuracy of learning algorithms where available

```
#number of participants
np <- 30

#number of questions
nq <- 10

#simulate responses as random binomials
resp <- matrix(data=rbinom(n=np*nq, size=1, prob=0.7), nrow=np)

colnames(resp) <- sapply(1:ncol(resp), function(x) paste0('q', x))
rownames(resp) <- sapply(1:nrow(resp), function(x) paste0('p', x))

#First 3 responses (success means agreement in liveliness assessment)
head(resp, 3)

##    q1 q2 q3 q4 q5 q6 q7 q8 q9 q10
## p1  1  1  1  0  1  1  0  0  1   0
## p2  1  1  1  0  1  1  1  1  1   1
## p3  1  0  1  1  0  0  1  1  1   1

#Overall detection
binom.test(sum(resp), length(resp))

##
##  Exact binomial test
##
## data:  sum(resp) and length(resp)
## number of successes = 207, number of trials = 300, p-value =
## 4.014e-11
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6343061 0.7418943
## sample estimates:
## probability of success
##                   0.69

#Individual question usability for analysis
apply(resp, 2, function(x) binom.test(sum(x), length(x))$p.value)

##          q1          q2          q3          q4          q5          q6
## 0.042773945 0.098737147 0.005222879 0.016124802 0.098737147 0.200488422
##          q7          q8          q9         q10
## 0.016124802 0.042773945 0.042773945 0.016124802
```

CONCLUSIONS

Main Contributions

Future Research

REFERENCES

ALEXANDER, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press

ALEXANDER, C. et al. (1977). *A Pattern Language*. Oxford University Press

ALEXANDER, C. (1979). *The Timeless Way of Building*. Oxford University Press

ALEXANDER, C. (2002). *The Nature of Order. Book One: The Phenomenon of Life*. The Center for Environmental Structure

ALEXANDER, C. (2002a). *The Nature of Order. Book Two: The Process of Creating Life*. The Center for Environmental Structure

ALEXANDER, C. (2004). *The Nature of Order. Book Four: The Luminous Ground*. The Center for Environmental Structure

ALEXANDER, C. (2005). *The Nature of Order. Book Three: A Vision of the Living World*. The Center for Environmental Structure

ALEXANDER, C. (2012). *The Battle for the Life and Beauty of the Earth: A Struggle Between Two World-Systems*. The Center for Environmental Structure

ASHAR, H., LANE-MAHER, M. (2004). Success and Spirituality in the New Business Paradigm. *Journal of Management Inquiry*, Sep 2004, 13, 3, pg. 249

AUBIN, J.-P., BAYEN, A. M., SAINT-PIERRE, P. (2011). *Viability Theory: New Directions*. Springer

BEER, S. (1981). *Brain of the Firm: the Managerial Cybernetics of Organization*. Wiley, Chichester

BIFFL, S. et al. (Eds.) (2006). *Value-Based Software Engineering*. Springer-Verlag, Berlin

BOULDING, K. E. (1961). *The Image: Knowledge in Life and Society*. Ann Arbor Paperback. University of Michigan Press

BOOCH, G., RUMBAUGH, J., JACOBSON, I. (2005). *The Unified Modeling Language User Guide (2nd Edition)*. Addison-Wesley Professional

BRAND, S. (1994). *How Buildings Learn: What Happens After They're Built*. Penguin Books, New York

BRAND, S. (2000). *Clock Of The Long Now: Time And Responsibility: The Ideas Behind The World's Slowest Computer*. Basic Books

BROOKS, F. (1995). The Mythical Man-Month

BROWN, W. et al. (1998). *Anti-Patterns: Refactoring Software, Architectures, and Projects in Crisis*. John Wiley & Sons, New York

BROWN, W. et al. (1999). *Anti-Patterns and Patterns in Software Configuration Management*. John Wiley & Sons, New York

BROWN, W. et al. (2000). *Anti-Patterns in Project Management*. John Wiley & Sons, New York

CAMPBELL, C. T. (2013). *Whole: Rethinking the Science of Nutrition*. BenBella Books

CASAKIN, H. P. (2007). Factors of metaphors in design problem-solving: Implications for design creativity. *International Journal of Design*, 1(2), 21-33

CHO, A. (2010). *The Jazz Process. Collaboration, Innovation, and Agility*. Addison-Wesley

COCKBURN (200X). Agile Software Development

COLLINS, J., PORRAS, J. I. (2004*). Built to Last*. HarperCollins, New York

COPLIEN, J., HARRISON N. B. (2004). *Organizational Patterns of Agile Software Development*. Prentice-Hall, Upper Saddle River

COWAN, F. S, ALLEN, J. K., MISTREE, F. (2006). Functional modelling in engineering design: a perspectival approach featuring living systems theory. *Systems Research and Behavioral Science. Special Issue: James Grier Miller's Living Systems Theory (LST). Volume 23, Issue 3, pages 365–381, May/June 2006*

CSIKSZENTMIHALYI, M., (1991). *Flow: The Psychology of Optimal Experience*. HarperCollins, New York

CSIKSZENTMIHALYI, M. (1996). *Creativity*. HarperCollins, New York

DAVENPORT, T. H., PRUSAK, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston

DIETZ, J. (2006). *Enterprise Ontology: Theory and Methodology*. Springer, Berlin

FAUCONNIER, G., TURNER, M. (2002). *The way we think: conceptual blending and the mind's hidden complexities*. Basic Books

FOWLER, M., (1996). *Analysis Patterns: Reusable Object Models*. Addison-Wesley, Menlo Park

GAMMA, E. et al. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley

GATES, B. (1999). *Business at the Speed of Thought. Warner Books*. New York

DE GEUS, A. (1997). *The Living Company*. Harvard Business School Press, Boston

DEMARCO, T., LISTER, T. (1999). *Peopleware. Productive Projects and Teams. 2nd ed*. Dorset House Publishing, New York

GOERTZEL, B. (2006). *The Hidden Pattern*. BrownWalker Press

HAY, D. (1995). *Data Model Patterns: Conventions of Thought*. Dorset House Publishing, New York

HAY, D. (2006). *Data Model Patterns: A Metadata Map*. Elsevier Morgan Kaufmann, Amsterdam

HOLMBERG, S. C. (2008). A New Era for Living Systems Theory. In R. Trappel (Ed), Cybernetics and Systems 2008. Vienna: Austrian Society for Cybernetic Studies, Vol 1: 179 – 184

VAN HEE, K.M., (1994). *Information Systems Engineering. A Formal Approach. Cambridge University Press*, New York

HOOKER, J. (2004). Is design theory possible?, *Journal of Information Technology Theory and Application 6* (2004, no. 2), pp. 73-82.

HOVERD, T., STEPNEY, S. (2010). Formalising Harmony Seeking Rules of Morphogenesis. *ALIFE*, pp. 386-393

KAUFFMANN, S. (2008). *Reinventing the Sacred*. Basic Books, New York

KAUTZ, K. et al. (2009). The Role of Improvisation in Information Systems Development. Systems Engineering for the Living Enterprise. *IADIS International Conference Information Systems 2009*, pp. 195-204

IIVARI, J., HIRSCHHEIM, R., KLEIN, H. K. (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. *Information Systems Research*. Vol. 9, No. 2, Jun 1998

KNUTH, D. E. (1984). Literate Programming. *The Computer Journal (British Computer Society).* 27 (2): 97–111

KORTEN, D. C. (1999). *The Post-Corporate World: Life After Capitalism*. Berrett-Koehler Publishers, San Francisco

KÖVECSES, Z. (2010). *Metaphor: A Practical Introduction, 2nd Edition*. Oxford University Press

LAPLANTE, P., NEILL, C. (2006). *Antipatterns: Identification, Refactoring, and Management*. Taylor & Francis Group

LAKOFF, G., JOHNSON, M. (2003). *Metaphors We Live By*. University of Chicago Press

LUHMANN, N. (1995). *Social Systems*. Stanford University Press, Stanford

MAXWELL, N. (2004). *Is Science Neurotic?* Imperial College Press, London

MCCONNELL, S. (2004). *Code Complete: A Practical Handbook of Software Construction. 2nd Edition*. Microsoft Press, Redmond

McCracken, G. (2010). Chief Culture Officer. Basic Books

MCDAVID, D. (2005). Systems Engineering for the Living Enterprise. *18th International Conference on Systems Engineering*, Aug 2005, pp. 244-249

MILLER, J. G. (1978). *Living Systems*. McGraw-Hill, New York

MORGAN, G. (1997). *Images of Organization*. 2nd Revised Edition. Sage Publications

NYSTROM, R. (2014). *Game Programming Patterns*. Genever Benning

PAGE, S. (2011). *Diversity and Complexity*. Princeton University Press

PAPOWS, J. (2010). *Glitch: The Hidden Impact of Faulty Software*. Prentice Hall

PARNAS, D. L. (1998). Software Engineering Programmes Are Not Computer Science Programmes. *Annals of Software Engineering*. Vol. 6, 1-4, pp. 19-37

PENG, R. D., MATSUI, E. (2015). *The Art of Data Science*. Skybrude Consulting, LLC

PENKER, M., ERIKSSON, H. (2000). *Business Modeling With UML: Business Patterns at Work*. John Wiley & Sons, New York

PRONK, A., KOMAN, H., LUIJKS, S. (2008). Mathematical Beauty in Architecture. *In C Sequin & R Sarhangi (Eds.), Mathematical Connections in Art, Music, and Science Leeuwarden: Tarquin*

QUILLIEN, J., ROSTAL, P., WEST, D. (2009). Agile, Anthropology, and Alexander's Architecture: An Essay in Three Voices. *Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications*, pp. 529-546

SAMUELSON, K. (2006). The System Symbols, from Deep-Insights to Universal Applications. *Systems Research and Behavioral Science 23*. pp. 349-363

SIMON, H. A. (1996). *The Sciences of the Artificial*. Cambridge, MA: MIT Press

SEIBEL, P. (2009). *Coders at Work*. Apress

SCHWANINGER, M. (2006). Theories of Viability: a Comparison. *Systems Research and Behavioral Science 23*, pp. 337-347

TAYNTOR, C. B. (2005). *Succesful Package Software Implementation*. CRC Press

ULRICH, K. T. (2011). *Design: Creation of Artifacts in Society*. University of Pennsylvania

SPINELLIS, D., GOUSIOS, G. (eds.) (2009). *Beautiful Architecture*. O'Reilly Media

WAGUESPACK, J. L. (2010). *Thriving Systems Theory and Metaphor-Driven Modeling*. Springer-Verlag, London

Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, 59 (10)

WITTGENSTEIN, L. (2010). *Tractatus Logico-Philosophicus*. Cosimo Classics

WITTGENSTEIN, L. (2009). *Philosophical Investigations.* Wiley-Blackwell

APPENDICES

A1. Common Functions

```
library(igraph)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#CENTER transformations

addCenter <- function(ang, center) {

  if (!is.element(center, vertex.attributes(ang)$name)) {
    ang <- ang + vertices(center)
  }

  linkList <- obs[grep(paste0(center, '>'), obs$object), 'value']
  for (i in 1:length(linkList)) {
    if (!is.element(linkList[i], vertex.attributes(ang)$name)) {
      ang <- ang + vertices(linkList[i])
    }
    ang <- ang + edge(center, linkList[i])
  }

  ang
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#VOID transformations

voidCenter <- function(ang, center) {
  delete.vertices(ang, center)
}
```

CURRICULUM VITAE

Personal Data

Name	Einar Polis
Date of Birth	12.09.1971
Citizenship	Estonian

Contact Information

e-mail	einarpo@gmail.com

Education

Period	Institution	Speciality / Grade
09/2002 to date	Tallinn University of Technology	Information Technology, doctoral studies
06/1997 – 06/2002	Tallinn University of Technology	Information Technology, masters studies
09/1989 – 05/1995	Tallinn University of Technology	Automated Control Systems engineer

Languages

Language	Level
Estonian	Mother tongue
English	Fluent
Russian	Average
Finnish	Average
French	Basic skills

Professional Employment

Period	Institution	Position
11.2009 to date	SOFGEN Services Ltd	Senior Technical Consultant
06.2005 – 03.2009	Bishop Cavanagh Ltd	Senior Technical Consultant
04.2000 – 05.2005	TietoEnator Estonia	Key Customer Manager
05.1999 – 02.2000	Estonian Telephone Company	Systems Analyst
09.1994 – 06.1996	Estonian Academy of Security Sciences	Lecturer
110.1993 – 04.1999	Estonian National Customs Board	Chief Expert

Scientific Work

Conference Articles

POLIS, E. (2007). Enlivenment of Information Systems. *Proceedings of the 10th Symposium on Programming Languages and Software Tools*. June 14-16, Dobogókö, Hungary

POLIS, E. (2009). Enlivenment of Enterprises. *Proceedings of the IADIS International Conference Information Systems 2009.* February 25-27, Barcelona, Spain

POLIS, E. (2010). Assessment of Value Analysis on an Example of Core Banking System Replacement Projects. *Proceedings of the Ninth International Baltic Conference Baltic DB&IS 2010*. July 5-7, Riga, Latvia

Chapters in Books

POLIS, E. (2011). Value and Viability Considerations in Information Systems Development. *Selected Papers of Baltic DB&IS 2010. Volume of book series Frontiers of AI and Applications.* IOS Press

Articles in Popular Press

POLIS, E. (2005a). Heatujuliselt maailmalõpule vastu. EE Homme 8 (17) /Facing the End of World in a Good Spirit/

POLIS, E. (2005b). Kontoritarkvara ärkab ellu. EE Homme 9 (18) /Office Software Becomes Alive/

POLIS, E. (2006a). Talletame failid aastatuhandeiks. EE Homme 1 (19) /Let's Store Files for Millenniums/

POLIS, E. (2006b). Arhitekt Alexander tõestas jumala. EE Homme 6 (24) /Architect Alexander proved the God/

POLIS, E. (2006c). Leiutaja saatus. HEI 2 /A Fate of an Inventor/

POLIS, E. (2006d). Elusettevõtluse perspektiivid. HEI nr 3 /Perspectives of Living Entrepreneurship/

Defended Theses

2002 Master's Thesis

1994 Graduate Engineer Thesis

Research Interest

X

ELULOOKIRJELDUS

Isikuandmed

Nimi	Einar Polis
Sünniaeg	12.09.1971
Kodakondsus	Eesti

Kontaktandmed

e-post	einarpo@gmail.com

Hariduskäik

Periood	**Institutsioon**	**Speciality / Grade**
09/2002 tänaseni	Tallinna Tehnikaülikool	Infotehnoloogia, doktoriõpe
09/1997 – 06/2002	Tallinna Tehnikaülikool	Infotehnoloogia, magistriõpe
09/1989 – 05/1995	Tallinna Tehnikaülikool	Automatiseeritud juhtimissüsteemid

Keeleoskus

Keel	**Tase**
Eesti	Emakeel
Inglise	Kõrgtase
Vene	Hea
Soome	Hea
Prantsuse	Algtase

Teenistuskäik

Periood	**Institutsioon**	**Ametikoht**
11.2009 tänini	SOFGEN Services Ltd	Konsultant
06.2005 – 03.2009	Bishop Cavanagh Ltd	Konsultant
04.2000 – 05.2005	TietoEnator Eesti	Kliendisuhete juht
05.1999 – 02.2000	Eesti Telefon	Süsteemianalüütik
09.1994 – 06.1996	Riigikaitse Akadeemia	Lektor
11.1993 – 04.1999	Riigi Tolliamet	Peaspetsialist

Teadustöö

Konverentsiartiklid

POLIS, E. (2007). Enlivenment of Information Systems. *Proceedings of the 10th Symposium on Programming Languages and Software Tools.* June 14-16, Dobogókö, Hungary

POLIS, E. (2009). Enlivenment of Enterprises. *Proceedings of the IADIS International Conference Information Systems 2009.* February 25-27, Barcelona, Spain

POLIS, E. (2010). Assessment of Value Analysis on an Example of Core Banking System Replacement Projects. *Proceedings of the Ninth International Baltic Conference Baltic DB&IS 2010*. July 5-7, Riga, Latvia

Raamatupeatükid

POLIS, E. (2011). Value and Viability Considerations in Information Systems Development. *Selected Papers of Baltic DB&IS 2010. Volume of book series Frontiers of AI and Applications.* IOS Press

Populaarteaduslikud artiklid

POLIS, E. (2005a). Heatujuliselt maailmalõpule vastu. EE Homme 8 (17)

POLIS, E. (2005b). Kontoritarkvara ärkab ellu. EE Homme 9 (18)

POLIS, E. (2006a). Talletame failid aastatuhandeiks. EE Homme 1 (19)

POLIS, E. (2006b). Arhitekt Alexander tõestas jumala. EE Homme 6 (24

POLIS, E. (2006c). Leiutaja saatus. HEI 2

POLIS, E. (2006d). Elusettevõtluse perspektiivid. HEI nr 3

Kaitstud lõputööd

2002 Magistritöö

1994 Diplomitöö

Teadustöö põhisuunad

X

LIST OF PUBLICATIONS

Conference Articles

POLIS, E. (2007). Enlivenment of Information Systems. *Proceedings of the 10th Symposium on Programming Languages and Software Tools.* June 14-16, Dobogókö, Hungary

POLIS, E. (2009). Enlivenment of Enterprises. *Proceedings of the IADIS International Conference Information Systems 2009.* February 25-27, Barcelona, Spain

POLIS, E. (2010). Assessment of Value Analysis on an Example of Core Banking System Replacement Projects. *Proceedings of the Ninth International Baltic Conference Baltic DB&IS 2010*. July 5-7, Riga, Latvia

Chapters in Books

POLIS, E. (2011). Value and Viability Considerations in Information Systems Development. In Barzdins, J., Kirikova, M. Eds. *Databases and Information Systems VI: Selected Papers of the Ninth International Baltic Conference, DB&IS 2010.* Amsterdam: IOS Press, pp. 257-270