Tomasz Estkowski

Towards a Generative Design System Based on Evolutionary Computing

ISBN: 978-82-547-0248-2

CON-TEXT
Thesis #59

A doctoral thesis submitted to:
The Oslo School of Architecture and Design

PUBLISHER:
The Oslo School of Architecture and Design

COVER:
Tomasz Estkowski

PRINTED BY:
Unipub AS

To my family

TABLE OF CONTENTS:

ACKNOWLEDGEMENTS

First of all, I thank my research supervisor, Professor Birger Sevaldson, for invaluable support and advice during the planning and development of this PhD. I enjoyed the cooperation with him very much!

I also thank my joint supervisor, Dr Martina Keitsch, for assistance and constructive critiques of my research. She discussed the thesis with me many times. She gave me very inspiring suggestions and helped to clarify my ideas.

I thank Professor Halina Dunin-Woyseth, Head of the PhD program at the Oslo School of Architecture and Design. I am grateful for her constant help, concern and positive attitude. I would not have started my PhD journey without her encouragement.

Special thanks go to Professor Nikos Salingaros of the University of Texas, who provided many relevant reading suggestions, and whose very positive comments on a first draft of the dissertation gave me much motivation in the last, most difficult phase of the research.

I am grateful to Professor Jim Tørresen for giving me the opportunity to present my project at his research group at the Department of Informatics, University of Oslo. This gave me the opportunity to get interdisciplinary feedback. I am also indebted to the Oslo School of Architecture and Design for awarding me a concluding grant, and to the library staff and the school's administration for their professional help.

Finally, I thank my wife Beata and my sons Mateusz, Damian and Konrad, for their patience and understanding throughout my studies. This dissertation is dedicated to them.

SUMMARY

This thesis proposes elements and principles for a digital *generative design system*, which involves *Evolutionary Computing*. The functioning of the system is based on an analogy between *a design process* and a *process of natural evolution*. The design system based on natural evolution seems especially appropriate in the early stage of a design process, where architects explore design possibilities by testing a number of different versions of design. Proposed generative design system is meant to support architects by generating different versions of *building models*, whose automatic development is guided by *design objectives* inputted by a system's user. Evolutionary Computation has been selected as the guiding programming approach for this project because of its high generative and creative potential, and its flexibility in terms of application.

Another key framework of this thesis is *sustainable architecture*. Here, sustainable architecture is understood first as a balanced integrative approach, which intends to comprehend architecture in its totality (in contrast to other systems, which focus on one or two architectural features). This property of sustainable architecture makes it suitable as a default system of values, which sets an unbiased direction for automated design development.

This thesis investigates four prototypes of a generative design system, concluding that they are fragmented. The main claim of the thesis is that a design system, in order to function, has to take into consideration a wide variety of design aspects (social, environment, economic), as defined in the guidelines of sustainable development.

A practical application of the principles developed and discussed in this thesis may be a computer-based system or *CAAD* (Computer Aided Architectural Design) system. Such a system would function as a creative design assistant during the conceptual stages of architectural design. This thesis, however, does not provide an operative CAAD system, but just the theoretical ground for it.

SECTION I – INTRODUCTION

1. RESEARCH SUBJECT AND RESEARCH QUESTIONS

My thesis discusses a *generative design system* – a computer program which can creatively facilitate architectural design by providing design proposals in form of digital building models. Throughout the thesis, I develop elements and principles of such a generative design system. In addition, I make a review of four examples of generative design systems, to investigate why they did not find their broader application in architectural practice. The main intention of the thesis is firstly to discuss and develop the elements and the principles of a generative design system in different contexts and secondly to look at them from different perspectives, in order to offer the reader a new understanding of the topic of computer-aided creative design.

The term generative design system has been used in many contexts and has many meanings. The common denominator for all generative systems is a capacity to generate or refine a building model *automatically*, based only on generally formulated properties of the intended building. Examples of such systems go back to 1960s and include systems that can:

- synthesise optimal room layout for a hospital, given the movement pattern of the hospital (developed in mid-1960s by Whitehead and Elders),
- create forms of buildings, based on a few simple geometric rules or transformation rules (for example technique called *shape grammar* developed by Stiny and Gips in 1971),
- create sophisticated building forms using existing computer aided design systems (such as Rhinoceros, MicroStation, CATIA, 3D Studio Max, Maya and so forth); a system's function that simulates physical force is used to create a shape of a building (e.g. the 'frozen waves' of Bernard Franken's BMW Pavilion in Germany),
- optimize position of façade openings in order to improve energy performance of a building at the same time providing good interior

illumination (system presented by Luisa Caldas and Leslie Norfold in 2001);

More detailed description of some of these systems is included in section II.

In spite of many approaches for creating a generative design system, it seems that its implementation to common architectural practice failed. There is a lack of a digital creative design assistance, which would take advantage of the state-of-the-art digital technology and which at the same time would fit well with the *actual* design practice. Judging from my professional experience, such assistance could be especially useful in the early stages of the design process, where an architect explores potentials of a building site, testing different building variants and adjusting assumed design objectives. Although a significant change has taken place in some areas of commercial Computer Aided Architectural Design (CAAD) systems (such as user interface, sophistication of the drawing tools, automation of the documentation process, or development of Building Information Model technology), the *creative* design systems are still only a subject of research. It seems that they lack a better adjustment to the specific nature of architectural practice.

One objective of this thesis is to develop and discuss elements and principles for a generative design system, which would be *operative* in the design practice. I assume that a viable generative design system has to be able to produce *integrated* and *comprehensive* designs, that takes into account heterogeneous, often conflicting architectural aspects.

My proposition for how the system should function is as follows. It should be a 'design assistant' during the early, conceptual stages of a design process. It should be able to generate many variants of building models, evaluate them and select one or a few that meet the *design objectives* best. The design objectives should be inputted into the system by a user, at the outset of a generation process, as a set of properties of the intended building. They could include for example building size, number of floors, functional program, energy consumptions or daylight illumination in selected spaces. Specification of these characteristics should be optional, and a user should decide which ones should be specified.

From my own architectural experience I know that the early phases of design involve a careful examination of a number of different properties of a building model, and their relations. Such an examination might include for example how a building's form affects distribution of functions, or how a building's volume defines a layout of shadows. The architect's task is to integrate all these aspect in one, logical, functional and elegant form, which would possibly fit with the client's intentions. In many cases, it is a laborious, in fact very mathematical work. It often involves examination of a large number of different building variants. Architects, relying on their intuition and experience, consider only seemingly most promising alternatives of building models. Here comes the benefit of the generative system. It could generate and analyze a large number of different building models very quickly, evaluate them and provide a few that 'best' fit with the design objectives. The system's user can use these generated models as:

- a starting point for a further careful design refinement,
- a source of information of what is possible in a given design context, or
- a source of inspiration, when a generated shape would be so unexpected that it would trigger a new idea of how the problem could be approached.

The generative design system is not supposed to replace the architect. It seems that the system will always be deficient, for at least three reasons:

- A building is a very complex entity and its mathematical representation (its model) is always reductive – a representation is a simplification of a phenomenon. Thus, selecting a particular way of representation over another reduces the number of potential building solutions that can be represented in that other way. For example, in recent Building Information Model (BIM) systems, not every conceived building can be digitally represented – it especially applies to complex forms of buildings. So, even though a generative design system could facilitate design and help to find interesting solutions, it might as well have a negative effect on innovative solutions – those which go beyond the predefined, generic structure of a building model;
- There are many unquantifiable building characteristics, such as aesthetic properties, which cannot be fully expressed numerically and processed algorithmically;
- Only a human user can fully comprehend a design situation in its socio-cultural context. Thus, only a human user can define design objectives that are adequate to the design context.

THE ELEMENTS AND THE PRINCIPLES OF THE GENERATIVE DESIGN SYSTEM

The main research question is "What are the elements and the principles of a viable generative design system?" Additional question might be posed:

- Considering the system as a creative design assistant, how exactly could the system function?
- At which stage of a design process can the system be applied and to which purposes precisely?

In the following paragraphs, I suggest the elements and the principles that the viable generative design system should include. This list is the hypothesis, which is supported by arguments and developed throughout the thesis. The complete list of principles reappears in the concluding section, accompanied by a comprehensive scheme that shows the overall functioning of the generative design system.

The principles of the generative design system are grouped around and discussed in connection to its main elements. These elements are:

- the *building model* (a representation of an intended building)
- the *building characteristics* (properties of the building model)
- the *building environment* (design objectives)
- the *evolutionary algorithm* (algorithm which generates building models, based on the principle of evolution by natural selection)

In addition, I assume that working of a system should be based on an actual design process. Thus, I discuss a design process and recognize its two general features, *adaptation* and *iteration*. I use these two features as the principles for constructing the generative design system.

The distinction of the elements from the principles is made only for convention and it is not strict. For example, I listed the evolutionary algorithm as an element of the system (because the system consists of the evolutionary algorithm), but it can be also understood as its principle, because it defines the way the system works.

In my thesis, most of the classifications and concretizations should be taken as *pragmatic* proposals, rather than *normative* declarations. My intention is not to make absolute statements of how the system has to be constructed, but rather to examine methodically the theme of a generative design in different contexts in order to provide a *well-informed* starting point for its realisation.

PRINCIPLES RELATED TO THE DESIGN PROCESS

The way the generative design system functions, should be adjusted to the actual design process. This is why I dedicated one section to an analysis of a design process. The following are two principles for the generative design system:

1. The functioning of the generative design system should involve *adaptation* of a building model. One can understand a design process as a gradual and non-linear transformation of a building model towards increasing fitness with the design objectives.
2. The generative design should involve *iteration*. By iteration in a design process I understand a cyclical redefinition of design objectives, caused by the knowledge gained from the design process. This knowledge includes information about achievability and pertinence of the assumed design objectives.

Accordingly, the functioning of the generative design system should be based on a more general process of iteration and a more specific process of adaptation. The iterative character of a generative design process implies that the design system should function as an exploratory tool rather than a solver for a well-specific problem. By differentiating input data and then, studying the output, the user would acquire information about achievability of differently formulated design objectives, which in turn would support the design process.

PRINCIPLES RELATED TO THE BUILDING MODEL

In most cases, I use the term *building model* to refer to a digital representation of an intended building, whose elements are interrelated. I propose the following principles for a building model:

1. The generative design system can create building models only in a predefined generic structure. Within this structure only a limited number of buildings can be represented. Therefore, this structure should be possibly flexible in order to make generation of very diverse building models possible.
2. A building model should consist of elements which resemble real architectural components. Practically, information should be attached to mathematical objects representing building elements. This information would indicate a kind of represented object and its properties.
3. A building model should have a hierarchic structure. In an actual building, the relations among its elements are not distributed homogenously, but there are areas of higher and lower density of relations. My proposition is that a construction of a building model should include this property. This property would speed up evolution of building models.

PRINCIPLES RELATED TO THE BUILDING ENVIRONMENT AND TO THE BUILDING CHARACTERISTICS

By the term *building environment* I understand a sum of factors that affect a building model, such as building's site (shape of terrain, size of a plot), building codes, mater plan regulations, user's intentions and architectural qualities. I use a word 'environment' in a specific context of the generative design system, where development of a building model occurs in an evolutionary way. There is an analogy between natural selection process and the functioning of the generative design system: as an individual organism evolves to fit the natural environment, so the building model evolves to fit the building environment.

The term *building characteristics* denotes properties of a building model that are implemented in the generative design system. Practically, the design system would need a specific mathematical function for each implemented building characteristic.

Below is a list of principles for the building environment and the building characteristics. I put the principles for both elements in one list, because they are linked – the building environment is defined by the building characteristics. An architect conceiving a building, considers a number of its different properties simultaneously and 'integrates' them in one form. Thus, the common aim of the listed principles is to generate the integrated and comprehensive building model.

1. The building environment should be *inclusive* and *balanced*. Consequently, the generative design system should be able to consider a number of very different building characteristics, reflecting a typical ability of an architect in an actual design process.
2. Instead of including one or a few building characteristics and simulating them to a high degree (*low resolution* and *high depth* of a building environment), it would be desirable to include many aspects and simulate them in less-details (high *resolution* and low *depth* of a building environment).
3. The building environment should offer a default parameter for each implemented building characteristic, in order not to necessitate a user to define all the building characteristics (because there are many of them). The default set of parameters should be based on sustainable architecture.

PRINCIPLES RELATED TO THE EVOLUTIONARY ALGORITHM

The generative design system proposed in my thesis is based on a programming technique called Evolutionary Computing (EC). This technique is applicable to diverse types of problems (so far it has been applied in art, biology, chemistry, economics, engineering, genetics, operations research, physics and robotics). EC is based on an analogy with the mechanism of biological evolution (natural selection and reproduction). The analogy is constructed as follows:

- an environment is identified with a problem specification;
- an individual in a population is identified with a candidate solution;
- a fitness of an individual is identified with a quality of a solution

I assume that the approach offered by EC is especially relevant for generating building models. The EC metaphor extended to building model generation looks as follows. The problem is specified by design objectives, candidate solutions are building models and quality of a solution is a quality of an individual model. To put it differently, the algorithm searches for a building model, which best fits the design objectives (Figure 1).

EVOLUTION	PROBLEM SOLVING	ARCHITECTURAL DESIGN
environment	problem	building environment (design objectives)
individual	candidate solution	building model
fitness	quality of solution	fitness of building model

Figure 1. The Evolutionary Computing metaphor and its extension to architectural design[1].

Below is a list of the techniques of Evolutionary Computing and the way they can be applied to the generative design system. These techniques can be understood as principles for the generative design system. These techniques are explained and discussed more in-details in section VIII.

---

[1] The figures and diagrams are developed and drawn by me, except the following: figure 2 (Weinberg, 2001), figure 6 (Butters, 2004), figure 11 (Eiben and Smith, 2007), figure 21 and 22 (Caldas and Norfold, 2001), figure 24 and 25 (de Vries et al., 2004), figure 27 (O'Reilly and Ramachandran, 1999), figure 35 (Brawne, 1992).

1. The evolutionary algorithm should apply *mutation rate control.* It is more desirable to have a large mutation rate at the beginning of the generation process and then to focus on refining the solutions that have been found so far. Due to large mutation rates, a large part of the search space could be examined at the beginning of the search process, increasing probability of finding good solutions.
2. The algorithm should use *parameter adjustment*. The building model's development could be improved by application of a dynamic adjustment of such parameters as mutation rate control, probability of mutation and crossover or a level of decomposition of a building model. Practically, these parameters would be subject of the evolutionary process of building models.
3. The evolutionary algorithm should apply a *parallel-terraced scan*. This technique optimally allocates the available searching resources in order to find the best balance between exploration and exploitation of a search space. Thus, at the initial phase of the search (generation) process, building models should be very different from each other, exploring possibly large space of design versions. Successively, the algorithm should focus on refining a few best-fitted building models.
4. The algorithm should employ *penalty functions* – a method for eliminating deficient (for example spatially inconsistent) building models from evolving population.
5. The algorithm should use a *tournament selection*. It is a method of *relative* evaluation of building models. Instead of evaluating a building in absolute terms (for example by points) the algorithm should indicate which model is 'better' in a direct comparison ('tournament') concerning selected building characteristic.
6. A *user intervention* in the process of building models evaluation should be allowed. A user should be able to monitor the adaptation process and intervene in defined periods to support the automatic evaluation of building models.
7. *Multimodal problems*, *multi-objective problems* and the *Pareto front* are tactics from which the generative design system could probably benefit most. These tactics involve evolution that considers many different objectives (often opposing each other) at the same time. Design problems are a class of multi-objective problems, because they involve several design objectives (the building environment is inclusive).

2. RESEARCH STRATEGIES AND METHODS

Any study of architecture is limited to a range of problems and “even the sum of such studies is unlikely to reveal the totality of architecture as it is used, sensed and understood in the everyday environment” (Brawne, 1992: 42). Nonetheless, there are attempts to systematize architectural research strategies and methods (Groat and Wang, 2002). Most commonly, research in the field of architecture is based on the strategies from the social sciences (Stake, 1995, 2006; Creswell, 2003; Yin, 1994). On the most general level, Creswell (2003) identifies two major groups of research strategies within the social sciences:

- quantitative strategies (experiments, surveys, etc.)
- qualitative strategies (case studies, narratives, grounded theory, ethnographies, etc.)

This thesis uses qualitative strategies. As Creswell emphasizes, qualitative research is essentially interpretive – the researcher analyses and interprets data, develops descriptions of the processes studied, draws conclusions about the meaning of the phenomena explored, and eventually states the lessons learnt and indicates the further questions to be addressed. In qualitative studies, the phenomena studied are approached holistically – such studies propose broad, panoramic views rather than micro-analyses (Creswell, 2003: 182). According to Creswell, qualitative research may be characterized as ‘emergent’, in contrast to a strictly prefigured quantitative inquiry. This means that the research questions are often restated in the course of research, as the inquirer gets a better understanding of the research problem. This ‘unfolding’ character of qualitative research makes difficult a precise definition of methods at the initial stage. Typically, multiple methods and complex reasoning are employed. Especially, the reasoning process may be described as:

- multifaceted (i.e., both inductive and deductive);
- iterative (moving back and forth from data collection/analysis to the reformulation of a research problem);
- simultaneous (consists of collecting, analysing and discussing data) (Creswell, 2003: 182-183).

1.1. TRANSDISCIPLINARITY AND RESEARCH VALIDITY

One of the features of architectural research is its inter- or transdisciplinary character. The challenges of working ‘out of a discipline’ or between disciplines have been studied in literature on transdisciplinary research, for example in Dunin-Woyseth and Nielsen (2004).

Consequently, this thesis involves interdisciplinary features, which comes with several challenges. Developing a digital design tool, which fully employs the possibilities of a state-of-the-art computing, requires interdisciplinary cooperation between architects and software engineers. It seems that sometimes a lack of understanding between the profession of architecture and computer science is indeed a difficulty. As Sevaldson observes, “Software engineers have their own distinct design methods and production culture, normally detached from insight into how designers work” (Sevaldson, 2005: 39). Architecture is associated to a high degree with the domain of art, where intuition, ingenuity and nonverbal imagination play a significant role. At the same time, architectural design - unlike painting, music or sculpture - requires a very rational and tangible approach, because it creates not only objects of contemplation and reflection but also functional objects constrained by real life requirements. On the other hand, computer science is based on a strict, formal approach, where clear definition of a problem and often mathematically advanced algorithms are essential. Because of this divergence, it happens that software engineers do not fully respond to architects’ needs, while simultaneously architects do not embrace the actual potential that contemporary computing techniques offer[2]. Kostas Terzidis expresses this concern as follows: “very few designers have the knowledge to understand the computational mechanism involved in a CAD system, or, reversely, very few CAD developers are also equally accomplished designers” (Terzidis, 2006: 41). The CAD technologies are not well understood in the design profession as a whole. Especially in architecture, an appreciation of what are these technologies offer is underdeveloped.

The subject of the thesis combines a number of diverse sub-areas of research, such as sustainable architecture, evolutionary algorithms, theory of systems, creativity in design, and others. Addressing these different research

---

[2] My presentation of this project at the Department of Computer Science, University of Oslo, seems to confirm the hypothesis that the computer scientists have an imperfect comprehension of the authentic architectural problems. The software engineers, participating in the presentation though very competent in their research fields, used to reduce the architectural problems to artistic study of a building form, assuming that a level of complexity and novelty of a form is a measure of its perfection. (ref. appendix - an interdisciplinary discussion of this project)

areas is a methodological challenge. In order to solve a specific research assignment, one has to collect the relevant knowledge from disparate research areas and then combine them while meeting scientific standards. However, when knowledge is used outside of its disciplinary context, it cannot be validated with the scientific methods characteristic to the discipline it is taken from. The following quotation reflects the problem:

> In the [academic discipline] the context and research is defined in relation to cognitive and social norms of academic science, which has tended to produce knowledge in absence of any practical goal. In Mode 2 the knowledge is intended to be useful. (Dunin-Woyseth and Nielsen, 2004: 32)

The authors compare the research within academic disciplines with the so-called 'mode 2 knowledge production'. Because of its distinctive nature, transdisciplinary research in architecture is often referred to as mode 2 knowledge production. While the traditional approach prevails in academic science (especially in the natural science), mode 2 is present in problem solving oriented research. Further, the authors characterize mode 2 knowledge production as follows:

> Mode 2 knowledge is created in broader, transdisciplinary social and economic contexts; in non-hierarchical, heterogeneously organized forma, which are essentially flexible and transient. It involves close interaction of many actors throughout the process of knowledge production, which thereby also becomes more socially accountable and more reflexive. (Dunin-Woyseth and Nielsen, 2004: 31)

The above quotation suggests that knowledge reliability in mode 2 is achieved through interaction of many actors and confrontation of many viewpoints. Their opinions contextualize the research content and verify it on different levels. In the context of a generative design system, the validity of the system would correspond to how helpful it is in a regular process of architectural design. Because there is no possibility to test the design system's performance in real design circumstances (due to the conceptual profile of the project) one has to test the principles of the system. The proposal here is to discuss these principles and the role of the system in a context of a typical design process, based on the author's own professional experience and the relevant literature on the topic.

Furthermore, as the project is a conceptual software development, it needed the expertise of computer science professionals. The main assumptions of the project, with a focus on application of Evolutionary Computation, were discussed at a seminar at the Department of Computer Science at the University of Oslo. The findings of the discussion are reported in the appendix.

Finally, the question remains: How knowledge from the different disciplines could be brought together and organised under a common denominator, in a way that keeps the scientific validity and gain expected relevance. In their discussion of a transdisciplinary research Dunin-Woyseth and Nilsen emphasize the practical approach, strongly oriented on problem solving. A researcher's attention should be focused on a problem area and knowledge should be applied on a temporary basis. In this context, a significant challenge of this study was to identify disciplines and areas that could fruitfully contribute to the research problem. I anticipated which areas of knowledge could be relevant to the project, changing my focus on the basis of the research problems, as they emerged in contexts of application.

1.2. LOGICAL ARGUMENTATION

The basic research technique of this project is *logical argumentation* as defined in Groat and Wang (2002), and it is supported by an *analysis* of selected generative design system prototypes. Logical argumentation can be classified as a qualitative research strategy following Creswell (2003).

According to the *Architectural Research Methods* (Groat and Wang, 2002) the research strategy of *logical argumentation* consists in "an ability to give logical order to a set of previously disparate factors (…) to frame logical conceptual systems that, once framed, interconnect previously unknown or unappreciated factors in relevant ways" (Groat and Wang, 2002: 301-302) .

The authors present a diagram, which illustrates different kinds of systems based on logical argumentation. On the one end of the diagram there are formal-mathematical systems, on the other end there are systems which address cultural issues. The formal-mathematical systems are ready to be digitally implemented (shape-grammar is given here as an example), while the culture oriented systems aim at conceptualising a worldview by displaying it in a logically consistent, systematic and clarified way (the authors propose the term *treatise* to denote them and give Vitruvius's *Ten Books of Architecture* as an example). Finally, between the two extremes, there are systems combining formal-mathematical and cultural-discursive properties. These systems use equations and principle-based analyses to cast light on social-cultural values. The authors provide Bill Hillier's and Julienne Hanson's *The Social Logic of Space* as an example, where an abstract map and a set of numerical figures are used as an analytic tool to display correspondence between social behaviour and space adjacencies. According to this diagram, my research is positioned between the poles, as it is neither a pure formal-mathematical system ready to be applied in a computer, nor a direct cultural discourse, but features both elements. On the one hand this thesis attempts to address the full spectrum of architectural design, i.e. both quantitative and qualitative aspects, on the other hand it attempts to organise them systematically, to make them applicable to the computational processing.

Logical argumentation is recognizable by a number of traits. The first trait is a *broad explanatory applicability* – "[the thesis] tends to have as a research outcome the framing of a conceptual system that has wide explanatory applicability" (Groat and Wang, 2002: 308). Here, the broad explanatory applicability can be understood on two levels.

On the first level, it can be referred to as the research outcome. The principles of the generative design system are meant to be universal - the field of its application is not limited to architecture, but (slightly modified) can be expanded to address design problems in general. Because of the use of Evolutionary Computing, the software would be open to effortless modification and further development. In addition, the way the software would be used is open. Because the problems the system deals with are 'wicked', it can be used as a design problem solver, a design optimisation tool, or a design exploration assistant. Alternatively, the forms it would generate may be used as a source of inspiration.

On the second level, the discussions undertaken here can be applied beyond the CAAD domain. The recognition of a formal structure of a building model appeals to a general systems theory. In this context, a building model is identified with a system and a design process is recognized as a process of adaptation, where a building model adapts to a given building environment (design requirements and design intentions). The analogy between a design process and a process of adaptation elaborated throughout section VIII, can find its application outside the software development activity. In other words, the research contributes to a broader body of knowledge about an architectural design process. In this light, the research findings can contribute to a more informed design practice.

Paradigmatic innovation is another trait of logical argumentation. Paradigmatic innovation connects disparate factors into novel, unified frameworks that “have significant and sometimes novel explanatory power” (Groat and Wang, 2002: 309). The contribution of this research is that it brings together heterogeneous aspects of architectural design and proposes a consistent framework for organising and processing them. The aspects (grouped in social, economic and ecological requirements) are recognised and gathered with the help of a ‘characteristics diagram’. The proposal of interrelating, evaluating and finally processing them mathematically is studied in this thesis in the perspective of Evolutionary Computing.

The next trait of logical argumentation is *a priori argumentation*. A priori argumentation takes place when “any specific instance of a thing is only ratification of principles that the logical system in question has already identified as the enabling conditions for that thing” (Groat and Wang, 2002: 309). The research findings are presented as a set of normative principles for a generative design system. They can be considered as a theoretical basis for the realisation of a generative design system. In other words, any specific instance of a generative design system can be seen as a kind of ‘ratification’ of these principles. Specifically in section IX, the principles are used for critical investigation of the prototypes of the generative design system. The investigation undertaken identifies the limitations of these prototypes.

The last trait of logical argumentation is its *testability* – the possibility to evaluate the validity or reliability of the theory. In the case of this research, whose aim is to formulate the design system principles, the theory *testability* must correspond to the *testability* of the formulated principles. As mentioned in the previous paragraph about research reliability, the principles are to a certain extent testable through professional review by practitioners and academics. The professional skills and academic experience can assess how the projected digital tool can possibly facilitate a design process in its conceptual phases. It is important to emphasise here that the validity of a qualitative research should not be evaluated in terms of its conformity to a set of methods. It is rather “a status given by a reader who is convinced that the research made responsible judgements and exercised care in the production of the study” (Polkinghorne, 2006: 76).

1.3. THE ANALYSIS OF PROTOTYPES

Section IX contains an analysis of four prototypical generative design systems. The examination is intended to enrich or validate the elements and principles of the generative design system, which are constructed throughout sections IV-VIII. The analysis focuses on:

- the building model: components and structure;
- the building environment: constituents and structure.

The elements and principles are a conceptual framework for explaining, describing or evaluating any realisation of a generative design system. The concluding paragraphs of the section IX answer the questions: which elements of the generative design system are already implemented in the software studied and to what extent? Which elements are missing, and why? Furthermore, the section identifies the differences and similarities between the prototypes from the perspective of the presumed principles. More specifically, the study consists of:

1. An identification of the shortcomings of the existing prototypes, and a formulation of how a given prototype could be improved and benefit from the principles worked out in this project.
2. An identification of the advantages and benefits of the existing prototypes. The lessons learned here led to an improvement of the suggested principles.

Information on the design system prototypes has been found in 4 research articles:

- “Architectural Constraints in a Generative Design System: Interpreting Energy Consumption Levels” (Caldas and Norfold, 2001),
- “Dutch Dwellings” (de Vries, 2004),
- “Evolution as a Design Strategy for Nonlinear Architecture: Generative Modeling of 3-D Surfaces.” (O'Reilly and Ramachandran, 1999),
- “Performance-Based Design Evolution: The Use of Genetic Algorithms and CFD” (Malkavi and Srinivasan, 2003).

The articles were selected from the following databases:

- Cumulative Index of Computer Aided Architectural Design Internet Database (CUMINCAD)[3]

---

[3] Ref. (http://cumincad.scix.net/cgi-bin/works/Home)

- Conference Papers Database of The Generative Art International Conference in Milan, Italy[4]

The key-phrases used in the search of articles were*: evolutionary computing, evolutionary algorithms, genetic algorithms, evolutionary architecture, generative design, generative system, constrained based design*. The presence of a generation method based on Evolutionary Computing was especially desirable. The search focus was on the authors who are most frequently cited or referred in the field-related papers and books. In cases when the key-phrase search resulted in hundred or more articles (for example in the case of the phrase *evolutionary algorithms* in CUMINCAD), the pre-selection was done according to the following criteria:

- The title of the article (containing one or more of the key-phrases),
- The author of the article (his/ her popularity),
- The abstract of the article (whether it contained some of the key-phrases).

The final selection of an article was done in regard to how closely the article content reflected the thesis interest.

3. EPISTEMOLOGICAL ASPECT OF THE THESIS

This research refers to computer science in its attempt to find a practical ‘method’ or a ‘system’ for design. Thus, it might seem that it situates itself within the project of positivism. Positivism aims at “(…) building up from simple elements to all higher forms that would, by virtue of the systematic constructional program itself, guarantee the exclusion of the decorative, mystical, or metaphysical” (Galison, 1990: 710). A very good illustration of the positivist attitude is the following statement by Theo van Doesburg:

> The work of art must be entirely conceived and formed by the mind before its execution. It must receive nothing from nature’s given forms or from sensuality or from sentimentality. We wish to exclude lyricism,

---

[4] Ref. (http://www.generativeart.com/)

> dramaticism, symbolism, etc. (…) Technique must be mechanical (…) (Van Deosburg, 1931, in: Osborne, 1979: 128)

However, there are at least two reasons for which the methods and the propositions of the thesis do not constitute a positivist approach:

1. The thesis acknowledges that there are design aspects that are difficult or impossible to describe in formal terms (or algorithmically). Consequently, it suggests the intervention of a human operator to supplement the system. Only a human agent can provide the system with artistic taste, sense of genius loci, sensitivity and imagination. In this sense, the system is a tool and cannot 'replace' the designer; it can only assist the designer and enhance his or her capacities.
2. The fundamental mechanism of the design generation is based on a black box principle, which involves stochastic processes. In this process, the system transforms a building model many times, gradually approaching the satisfying solution (in terms of design requirements and intentions). This fact makes the building model by no means entirely conceived and formed by the mind before its execution, as postulated by the positivists. On the contrary, the generated model is only partly envisaged and controlled by the designer (indirect design) and it is hardly ever an optimal solution. Far from the positivists' ideas, the thesis postulates that the architectural design process is to some extent circular and consists in solving 'ill-structured' problems (i.e., the solution points back at the problem initially formulated, calling for its reformulation or adjustment).

In this context, even though the thesis is an attempt to address the rational arguments about the design process, it admits that there is a significant area of architectural design that positivists would call 'mystical' or 'metaphysical'; this area evades the rational formulation (and it evades algorithmic formulation). The automation of the design process proposed here is limited and it takes place under the initiative and control of a designer. The automation of the building model generation proposed does not harm the integrity of the design process by neglecting or simplifying the subtle design aspects. The essential assumption is that the design process should be approached holistically. This requires that the computable aspects do not prevail in design and that they are supplemented and counterbalanced by the remaining, complementary aspects. Therefore, the overall methodological way of approaching problems in this thesis is close to a post-positivistic approach.

Postpositivism “acknowledges the importance of socio-cultural studies in its analysis of the built environment, thus encourages a more dynamic and holistic view of design” (Sirowy, 2010: 52). Specifically, this thesis continues and extends discussion in areas such as space syntax (elaborated on in the late 1970s by Bill Hillier and Julienne Hanson) or Christopher Alexander’s concept of the ‘pattern language’ (Alexander, 1968). Both these concepts are an attempt at rationally investigating architectural design (in terms of applying scientific methods) while at the same time being sceptical about dogmatic and direct application of the scientific methods. In natural science, the shift from positivism to postpositivism is associated with Popper’s idea of falsification, which posits that if one cannot prove that a proposition is true, one can prove that a proposition is false. However, we can use a theory as long as it is not falsified. Scientific progress consists of constant verifications of theories based on new evidence. This concept fits very well with the idea of architectural design advocated in this thesis, which indicates that design comes about in an iterative process of ‘fine-tuning’ the initial design intentions based on progressing knowledge on specific design circumstances.

4. THESIS’ OUTLINE

The thesis consists of ten sections. At the end of section IV, V, VII and VIII there is a list of principles that relate to the theme of the respective section.

In the first section, I describe what a generative design system is and give a few examples of such systems. I maintain, that these systems found little application in an actual design practice. I formulate my research question, that is: “What are elements and principles of a generative design system that would find a broader application in a typical design process?” I propose to answer the question at the very beginning of the thesis, with a list of elements and principles for such a generative design system. The central body of the thesis (sections IV-VIII) contains arguments in support of the proposed elements and principles.

The second section introduces the reader into a theme of CAAD systems. In this section I follow the development of CAAD systems and I make their classification. My purpose is to position a generative design system in a broad research field of CAAD systems.

The third section is a theoretical one, in a sense that it does not directly refer to the development of elements and principles of a generative design system. Its role is to prepare a background for such a development, exploring *Systems Theory*. I look at different definitions and properties of a system, in order to use them specifically for the development of a generative design system.

One of my postulates is that a generative design system should be well-adjusted to the actual design process. Thus, the fourth section examines a *design process*. The examination is based on my experience as a professional architect and on the review of related literature. I recognize an *adaptive* and an *iterative* character of a design process. My assumption here is that adaptation and iteration should constitute a framework in which a generative design system should operate.

In the fifth section, I focus on important element of a generative design system – a *building model*. I assume that a building model should be conceptualized as a *system* that *adapts* to a set of design objectives. I use definitions and properties of a system described in the section III, and apply them to a building model. I claim here, that a building model for a generative design system should be approached holistically and its general structure should be hierarchic.

In the sixth section I focus on *building characteristics* – properties of a building model. I argue that they should play an essential role in generation of a building model. Specifically, a user of a generative design system should control the development of a building model through its characteristics rather than directly through its elements.

In addition, in this section I give and discuss a few examples of building characteristics. For instance, I examine how building aesthetics can be approached algorithmically and how this approach can be applied to a generative design system.

The seventh section contains a central element of the generative design system – a *building environment*. The building environment embraces and defines all factors that affect a building model during its development.

Firstly, I postulate that a building environment needs to be inclusive and balanced. I propose to base it on building characteristics related to *sustainable architecture*, because it is a comprehensive and holistic approach.

Secondly, I provide an outline for a generic building environment, showing, which building characteristics it should include. The generic building environment should consist of building characteristics that include different *design constraints* (site constraints, building codes, master plan regulations, client intentions and architectural qualities) and that address *design domains* of ecology, economy and society.

Finally in the seventh section, I show how a generated building model can be evaluated. My proposal is that the similarity between characteristics of a given building model and characteristics of a building environment would define the fitness of a building model. Such evaluation should be made automatically by a generative design system and is essential for the automatic generation of intended building models.

In the eighth section I discuss another important element of a generative design system, the *evolutionary algorithm*. This algorithm connects a building model, building characteristics and a building environment into one system, and so, it defines the way the generative design system works. Specifically, the algorithm simulates the mechanisms of natural evolution and thus it can be used to execute the development of a digital building model, which evolves to fit with a building environment.

I begin section VIII with an introduction to Evolutionary Computing – a general approach for problems solving. I explain its essential mechanisms (generation, selection, modification, multiplication) and show how EC is applied in a broad array of disciplines. Then I suggest how EC can be applied in a generative design system, specifically. My main contribution in this section is, that I show how selected techniques of EC can be applied to the generative design system. Lastly in this section, I reason why exactly the evolutionary algorithm is relevant for the generative design system.

In the ninth section I make an analysis of selected, realized generative design systems in order to *support* my assumption about the need for an inclusive building environment. I review four prototypes, attempting to answer why these design systems did not find its application in an actual design process. In my analysis, I use the concepts developed and discussed in the previous sections, such as: a building model, building characteristics, a building environment and the evolutionary algorithm. The central finding of the analysis confirms my hypothesis that a building environment of a viable generative design system has to be possibly inclusive and balanced.

Section X is the concluding one. It brings together the finding of the thesis, shows contribution to the existing body of knowledge, identifies strengths and limitation of the thesis and suggests directions for further research.

The beginning of this section contains a synthetic description of the thesis' findings. The description is accompanied with a scheme, which shows the connected elements and principles of the generative design system. The scheme is indented to support the textual explanation of the overall functioning of the system. Then, I go into details, describing the elements and the principles of the generative design system that were proposed as a hypothesis in the introduction and were developed and supported throughout the thesis. In addition I show how a designer, a client and the nature can benefit from a generative design system.

This section contains also a more abstract reflection over a generative system and its functioning. I distinguish between systems that are black boxes and white boxes, postulating that a generative design system should work mainly as a black box. I propose the term 'indirect design' for describing the way the generative design system would work. The indirect design is different than a usual design since it focuses on performance of a building model rather than on its structure. The indirect design consists in manipulating building characteristics rather than building elements.

SECTION II – RESEARCH POSITION AND THE STATE-OF-THE-ART

The purpose of this section is to introduce the reader into a theme of CAAD systems and to position the generative design system in the broader research field. This section consists of two parts.

In the first part, I follow the development of CAAD systems and try to identify their limitations. I recognize that the trend in development of CAAD systems in the last decades was to advance *technically* the production, modification and reproduction of drawings. Compared to this trend, the initial development of CAAD systems was more ambitious. The aim was not as much at improvement of the drawing techniques, but at rationalizing and automating a design process itself.

In the second part, I discern four groups of CAAD systems based on their focus:

1. drawing and modeling tools,
2. organization of a building model,
3. capacity to evaluate building models and
4. capacity to generate building models.

The generative design system proposed here belongs to the fourth group, as it focuses on automatic generation of building models. I give examples of design systems that aim at design automation. I also refer to developments in artificial intelligence – approaches to digital creativity that combine randomness with defined rules. These approaches include: evolutionary computing, expert systems, case-based reasoning, artificial neural networks and fuzzy logic. I suggest that the generative design system that is developed in my thesis should apply evolutionary algorithm. An adequate computer application would support design exploration and accordingly, lead to a more informed design practice.

1. DEVELOPMENT OF CAAD SYSTEMS AND THEIR LIMITATIONS

In architecture, Computer Aided Design (CAD) systems have been used for decades. CAD refers to the software that uses computational technique and that is applied in design. With a CAD system, a user is capable to create and manipulate digital geometric objects displayed on a screen. The objects (e.g. geometric figures) are kept in a database and they constitute a digital building model. A set of digital tools provided by the software helps the user to manipulate the objects.

Computer Aided Architectural Design (CAAD) systems are a special kind of CAD systems, which – because of customised set of modelling tools and representation methods (3D visualisation) – are suitable especially for architectural purposes. CAAD systems help in the production of architectural drawings at several stages and without using paper, drawing boards, triangle, scale, pencil or pen. The drawings are displayed on a computer screen and they can be discussed and manipulated, or they can be printed on paper and used as production drawings.

Considering these capacities of traditional CAAD system one can conclude that these systems are essentially focused on *technical* improvement of the production, modification and reproduction of drawings. In this light, the early CAD systems were more ambitious. They were aiming not as much at improving the drawing techniques, but at rationalizing and automating a design process itself.

Attempts to rationalize the design process were made already in the 1930s. The idea to standardize building components made possible manufacturing of components in predetermined ways in factories. The Hertfordshire County Council's development of prefabricated classrooms in the 1930 is an example. The components themselves were predictable and amenable to incorporation within a computer-based system. They could be assembled in many different ways. The idea was further developed and in the 1960s it was implemented on IBM computers, on which a user could manipulate predefined building components on a computer screen. In addition, the computer program was able to *analyse* the performance of the constructed building. In the 1960s and 1970s, a series of computer-aided systems were developed. Some of them were:

- HARNESS - for automated design of hospitals,
- OXSYS – in which a manual outline of design was processed by a combination of manual and automated methods,
- EdCAAD – for optimising the floor layout of the houses and then their distribution on the site through calculation of road alignments

> and carrying out a range of environmental analyses (Day, 1997: 35 – 37).

The idea of rationalizing design can be traced back to Design Methods research, which in turn was based on the earlier development of Operational Research techniques. One of the representatives of Design Methods research was Christopher Jones. His aim was not to replace the traditional methods of design (i.e., those built on knowledge, experience and intuition), but to enhance them by codifying the design aspects which could be expressed in formalised way in diagrams, charts and lists. He divided design actions in three components: *analysis* – the description of the problem, *synthesis* – the generation and manipulation of solutions, and *evaluation* – the testing of the resultant solutions. It is interesting how similar his conceptualisation was to the approach of Evolutionary Computing, which consists first in the definition of the building environment (analysis) and then in the cyclic generation, modification (synthesis) and evaluation of candidate solutions.

The second representative of Design Method – Bruce Archer – elaborated the method further. He developed a system of seven main components containing 200 activities. Nonetheless, these attempts to rationalize design turned out not to be very successful in practice. Design Methods were since then gradually replaced by computer applications – CAAD systems. Conferences on the use of computers in design – like for example one held in MIT at the end of 1960s - successively replaced discussions on Design Methods, though the central concepts of Design Methods research were retained.

Nowadays, CAAD systems are present in every architectural office. The computer replaced totally the earlier design environment, which was dominated by the drawing board, pencils and rulers. Even though, it seems that CAAD systems' potential in aiding design in the areas where it can be rationalized – as Christopher Jones pointed out – is underdeveloped. Alan Day puts it in this way:

> One might imagine that one is witnessing the culmination of a revolution in the way buildings are designed and constructed. However, in reality, this is not so. For all the gloss of new technology nothing has really changed other than the automation of tasks that were once carried out by hand. Rather like the carpenter's electric drill which has replaced the brace and bit, the computer allows work to be completed more quickly, turning capital investments into increased

> efficiency, but the revolutionary aspects of the technology are currently under-exploited with only the immediate efficiency gains being realized. (Day, 1997: 35)

The success and omnipresence of the digital technologies in design – though at the same time underdevelopment of their potential – was also emphasised by Sevaldson (2005). He claims that even though the creative techniques provided by computer technology led to a more complex and inclusive design process, the potential of many digital techniques (for example digital modelling) is far from being fully explored (Sevaldson, 2005: 348). A lack of a digital assistance adequate to the state-of-the-art digital technology is especially evident in "(…) early stages of the design process; the explorative phases before the constraints of realisation start to narrow down the options" (Sevaldson, 2005: 9). Woodward identifies the same problem, pointing at a lack of conceptual digital tool:

> No currently commercially available program provides a substitute for or supersedes the use of paper and pencil for sketching architectural ideas and diagrams, or for the technique of design in which a plan, for example, is encouraged to emerge from a skein of lines by progressively firmer drawing, erasing and redrawing. (Woodward, 1997: 89)

Even though a change has taken place in some areas of CAAD systems, such as in user interface, sophistication of the drawing tools, or higher automation of the documentation process, the creative aspect of design has remained untouched. The current software has still far to go when it comes to the creative user – there is not as much development in this area as one could expect.

2. CLASSIFICATION OF CAAD SYSTEMS

The following classification of state-of-the-art CAAD systems will make a context, in which I will position the generative design system developed in this thesis. The systems available nowadays can be classified in many ways. According to Christopher Jones (1992), they can be classified in terms of stages of a design process they address:

- design problem analysis systems (formulating design intentions);
- design synthesis systems (producing a building model using a specified criteria) or
- design evaluation systems.

Many CAAD systems combine two or all the three areas. Schodek (2005) proposes to distinguish design systems (used for drawing and modelling buildings) from analysis systems (used for evaluating buildings). Again, it is difficult to find a well-defined boundary between these two groups, because CAAD systems usually join analysis and synthesis and other functions. So, the classification I propose below is based on the *focus* of a given CAAD system. CAAD systems can focus on:

1. Drawing and modelling tools.
2. Organisation of a building model.
3. Design analysis/ evaluation tools (Jones' evaluation systems, Schodek's analysis systems).
4. Design generation tools (Jones' synthesis systems).

DRAWING/ MODELLING TOOLS

The simplest digital drawing tool allows drawing primitive geometric figures such as lines, rectangles, ellipses, polygons, arcs, curves, etc. Each of these objects has a set of attributes, such as type, position, size, line thickness, pattern and colour of fill. To create an object, the user specifies its spatial position in the Cartesian co-ordinate system and its parameters – for example, in the case of a circle, the position of its centre and its radius.

More advanced tools allow creating complex objects (‘symbols’, ‘blocks’, ‘compounds’ etc.). Here, simple objects (lines, circles etc.) are combined into a group (for example a door or a window) which makes them more manageable. In addition, a user can use ‘libraries’ of such ready-drawn complex objects created and provided by different manufacturers.

The modification tools can transform objects. Selected elements of a drawing can be easily copied and re-used. Shapes can be filled with hatches or patterns, to signify their different functions. The systems provide a facility to dimension the drawing easily, to make elegant and editable notes, captions and titles, to cut an adequate fragment of a drawing (make a ‘view port’, ‘sheet layer’ etc.), and to print it out in a relevant scale. The more sophisticated tools allow users to:

- Construct and draw an object in many alternate ways, e.g., a line as a tangent to a circle; a line at the mid-point and perpendicular or parallel to another line; a circle set out from its centre or in relation to another object.
- Quickly draw one- or two-dimensional arrays of regularly occurring objects, e.g., setting out lines, a grid of columns, using rectangular or polar coordinates.
- Set up ‘associative’ dimensions, that is, dimensions which refer to particular objects and which change automatically when the sizes of the objects are changed.
- Automatically add straight or curved chamfers to the corners of objects.
- Use Boolean operations on pairs or groups of objects, in order to add or subtract objects.

Beside these drawing tools, there is a toolset especially designed for modelling 3-dimensional objects. A basic toolset allows:

- extruding 2D geometric figure to give it ‘depth’;
- lathing a profile to produce objects of rotation;
- sweeping – rotating a profile while applying other transformations;
- 'lofting' to produce a surface;
- offering 'primitive' shapes such as boxes, pyramids, spheres, cylinders and cones;
- editing pre-existing shapes by moving their vertices (Woodward, 1997: 107);

So far, three kinds of objects were mentioned:

1. Simple 2D objects, like lines, rectangles, circles and others;
2. Assemblies of 2D elementary objects, which make complex objects like windows or doors (‘symbols’, ‘blocks’, ‘compounds’ etc.);
3. Simple 3D objects, like faces, boxes, pyramids, spheres, cylinders and cones.

It seems natural that the simple 3D objects can be assembled similarly to the 2D elementary objects, making 3D ‘compounds’. But unlike 2D assemblages, the 3D assemblages can contain information about their volume and - referring to real building materials – about their weight, or cost. Therefore, such 3D assemblages can denote real architectural components in a more comprehensive way than the 2D assemblages. One can say that such assemblages have ‘semantic content’ attached to them (Schodek, 2005: 185). Examples include many architectural, engineering and construction objects: walls, windows, doors, staircases, columns, slabs, roofs, ramps, beams, bolts, steel angles and the like.

One of the benefits of complex 3D objects is that they have adequate modification tools for each of the complex object. They can be modified by the parameters that describe them. For example, a staircase can be determined by parameters describing the size and number of steps.

Unlike basic modification tools, which are able to resize an object only proportionally, the modification tools of complex objects enable the user to modify the objects indirectly, by editing the properties (or parameters) of the specified object. For example, if a staircase were a complex object, changing the number of steps (that is, changing the properties) would automatically affect the height and the length of the staircase.

One could go a step further and consider yet more extensive assemblage – a group of all complex objects, a new ‘super-complex’ object. Such a ‘super-complex’ object would represent the whole building model. Now, modifying the parameters of such object would affect the properties of the building model itself. The modelling process would change fundamentally – instead of drawing each element of a model separately, one would rather form the model dynamically by manipulating its general parameters. The constituents of the model would be rearranged automatically in function of the parameters.

Such a dynamical modelling is possible under at least one condition: the relationships between all the compound-objects of the model should be recognized and defined. Then, the whole design could be not only parametrically ‘driven’, but selected properties of the design could be automatically specified[5].

---

[5] GenerativeComponents, CATIA, SolidWorks, Pro/ ENGINEER, Unigraphics are examples of CAD systems that include parametric modelling, dimensionally driven design, feature-based design, assembly modelling, and similar techniques. This type of modelling consists first in establishing relationships amongst model components and then manipulating their parameters either directly or with the help of algorithms (when a more complex form is being modelled). These programs often combine parametric modelling with analytical tools for design evaluation (Schodek, 2005: x). The possibility of achieving an indirect control over a building model by manipulation of design intentions and design requirements is one of the key concerns of this thesis.

ORGANISATION OF A BUILDING MODEL

A conventional CAAD system consists of a digital building model and a set of digital tools, which enables creating and manipulating the model. A basic building model is a repository of lines, circles, polygons and other geometric figures, and a number of other types of objects, such as parametric curves (splines) or symbols. Such CAAD systems can provide only very abstract information about a model. For example, one can get information about a number of blue polygons, a number of 0,5mm thick lines, and the like. However, this information says nothing about the actual building, and particularly about how the building elements are constructed, what materials are used or how they are fixed to each other. The geometric figures defining the building do not refer to any architectural or physical components unless a user interprets them. In some cases, this fact leads to a confusion or ambiguity, where the meaning of a certain line, a rectangle or a circle is not clear to those who did not draw them (especially to non-architects).

On the other hand, such an abstract representation might be advantageous, especially when an ambiguity or ‘openness’ in interpretation is desirable, such as in the early stages of design, when the concept gradually develops.

In architectural practice, actual 3D building model is often described through sets of 2D drawings. Architects draw plans of a building model (horizontal sections) and the selected vertical sections based on these plans, which together give a full description of the model. Every time that a change is made in the building model however – it is necessary to revise all the sections and plans, which include the new element. In other words, the architect must check manually the consistency of all the drawings – that is, check whether they give an unambiguous description of a 3D building model.

The situation is different when working with software that “is inherently based in three-dimensional models, from which two-dimensional views and other information may be derived” (Schodek, 2005: 5). In this case, the architect works in 3D space, manipulating 3D objects. The revision of 2D ‘views’, arranged earlier by a user, is automated and so their consistency is always secured.

There are three types of 3D models: wireframe models (created from points and lines), surface models (where space is enclosed by surfaces that can be manipulated) and solid models. The solid models are “essentially volumetric models, where complex curved lines and bounding surfaces that define model configurations are numerically defined in an exacting sense” (Schodek, 2005: 5). Numerical definitions of objects in the case of solid models make possible intersections of their complex geometries. Moreover, solid models allow derivation of volumes of objects (or assemblies of objects), as well as more sophisticated data, such as centres of gravity, masses of the objects or other information.

Today, many CAAD applications operate on sophisticated 3D digital models, called Building Information Models (BIMs)[6]. In contrast to a typical digital model, which is a repository of *geometric figures*, basic constituents of BIM are *building components*, which are *logical* assemblies of 3D geometric figures accompanied by information (for example, a geometry of a door with specification of the materials used and manufacturer's details). Operating on building components makes possible the derivation of various kinds of broad information about the building model, such as the key spatial relationships or relevant quantities and properties, which help to analyse the building. BIM is an attempt to make a full digital description of a building.

ANALYSIS/ EVALUATION TOOLS

The third group of CAAD systems are programs capable to analyse selected characteristics of a digital building model. Woodward (1997: 32) gives a list of areas, where such software finds application:

- foundations, piling, hydraulics;
- structural analysis, wind loads, steel, concrete, timber, masonry
- environmental analysis, space loads, condensation, lighting, noise, fire, ventilation;
- energy consumption and monitoring;
- mechanical services, plant, ductwork, lifts;
- piped services;
- electrical services.

Moreover, there are CAAD systems that target at ecological analysis of design. I will give three examples of such systems.

ArchiCAD is the first example. It requires external, ecologically profiled addition, a program called EcoDesigner[7]. This addition is integrated in the ArchiCAD design environment and can evaluate energy performance of the building model.

Another example is Autodesk Ecotect Analysis[8], which is software targeting at sustainable design. The program offers a broad range of

---

[6] The commonly used software with applied BIM includes: Autodesk's Architectural Desktop and Revit, Bentley's Microstation, Nemetschek's Allplan, and Graphisoft's ArchiCAD (Schodek, 2005: 184).

7 (ref. http://www.graphisoft.com/products/ecodesigner/)

8 (ref. http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=12602821)

simulations and energy analysis, such as energy and water consumption or carbon emission, integrated with tools that visualise the building's performance. Specifically, the program is capable to perform:

- whole-building energy analysis (total energy use and carbon emissions of a building model);
- thermal performance analysis (heating and cooling);
- water usage and cost evaluation;
- solar radiation analysis (visualise solar radiation on windows and surfaces);
- day lighting analysis (luminance levels at any point in the building model);
- shadows and reflections analyses.

The last example is IES (Integrated Environmental Solutions)[9]. IES offers a range of analysis software that allows architects to test different design options (in terms of energy efficiency, low-carbon technologies, occupant comfort, light levels, airflow, etc.) and draw conclusions on what the best solutions are.

Analysis software is more often present in CAE (Computer Aided Engineering) systems than in CAAD systems. In engineering applications, a technical evaluation of a design solution is essential. Each model is subjected to a number of technical tests, because it is to be manufactured in many copies (often hundreds of thousands or more, such as in the case of cell-phone models).

The case of buildings is essentially different. Usually, a building model is to be built only once. Therefore – for economic reasons – resources allocated to perform overall technical tests for a building are smaller. But even more importantly, the technical performance of a building – except its structure, which must be correct to preserve the building from collapsing – is not as important as its 'social' functions, such as arrangement of spaces, visual qualities, sense of privacy, belonging and the like.

Furthermore, there is a difference between testing the ergonomic performance of an automobile and that of a building. In the former case, the test is relatively easy to perform, because there is a 'template' for design; for example, there is a steering wheel in front of the driver and a gear stick on his or her side. The position of these elements can change only slightly in different car models (e.g., the gear stick cannot be 2m far from a driver). In the case of a building, testing ergonomics is much more difficult to perform, even within a single category of building. The layout of one flat can differ drastically from another one. So what should the ergonomic performance of a

---

[9] (ref. http://www.iesve.com/software)

flat take into account? One might test a relation between key elements of a kitchen, such as a fridge, an oven and a working space. But in many cases, the particular shape of the flat, or culturally-specific ways of arranging kitchen, or a particular need of a client, might be more important than the 'solutions' considered as ergonomic or 'standard'. The point here is that, it is much more difficult to implement the evaluation of different aspects of building performance (in CAAD systems) than the evaluation of performances of objects such as cars (in CAE systems), because in the former case it is more challenging to determine criteria for an optimal performance.

DESIGN GENERATION TOOLS

The last group of CAAD tools contains systems that deal with the automation of design generation. These systems support the user by generating design proposals automatically.

Surprisingly, a mainstream of the early digital design systems used to be quite ambitious and they aimed at supporting designers in the *conceptual* stage of design. The first example might be the program BAID (Basic Architectural Investigation and Design), developed in 1972 and aimed at housing layout studies. The program randomly generated site layouts, which satisfied initially defined conditions (constraints), e.g. minimum amount of daylight and sunlight in each room combined with required amount of privacy (Day, 1997: 39).

Worth mentioning is a program created in the mid-1960s by Whitehead and Elders focused on a design synthesis for the layout of a single-storey hospital. A list of desired connections between rooms in a hospital was given as an input. Then, the program was supposed to optimise the connections. The preferred connections could be specified either by authoritative principles or by investigating the movement patterns of the hospital.

In 1971 Stiny and Gips presented a more general approach to the same area of generative design. This approached is called *shape grammar*. According to the authors, the system capable to produce new forms should consist of a set of transformation rules (applicable for a shape) and a generation engine that would apply these rules. In a generation process, these transformation rules are applied to an initial shape that changes, eventually becoming a new shape. The transformation is monitored by the system and the process can be stopped when a set of conditions is fulfilled. Shape grammar is a universal, formalised system, appropriate for the generation of architectural objects. It was used to study existing architectural objects like Palladian villas and Victorian windows (Mitchell, 1990) and for creating new designs like Alvaro Siza's Malagueira housing project (Duarte, 2001). A significant achievement of shape grammar is the introduction of the shape transformation method[10]. This method could find its application in the Evolutionary Computing approach to design, particularly when the evolved object undergoes a transformation.

A recent popular approach to generative design excels in the generation of sophisticated forms. This approach does not require particular software that specialise in form generation, but it takes advantage of existing, advanced systems for shapes modelling (such as Form-Z, Rhinoceros, MicroStation, CATIA, 3D Studio Max, Maya and so forth):

> (…) there has been an interest on the part of some more speculatively oriented architects in using one or more of external factors as a “direct generator” of a building's shape, or in using some reference metaphor (e.g. the “frozen waves” of Bernard Franken's BMW Pavilion in Germany). In these situations, the curved surfaces assume (automatically within an appropriate computational environment) shapes in response to a prescribed forcing function of one type or another. (Schodek, 2005: 51)

The drawing or modelling tools that these programs offer are used in creative, novel ways (and not necessarily for the purposes these tools were designed for), which leads to the generation of unexpected forms.

A variant of this approach is the parametric shape variation. In this case, a designer first defines a shape with a set of parameters and relations between these parameters, and then – by modifying some of the parameters – he or she generates a spectrum of forms. Unlike in ‘traditional’ parametric design, in which parameters describe some desirable and ‘reasonable’ relations between the elements of a model, in parametric shape variation the parameters as well as the relations are used freely, because the only purpose is to generate interesting or inspiring form.

Yet another approach is growth/ repetition algorithms (cellular automata). Here a simple set of rules for a form growth (an algorithm) can generate very complex forms. Manipulating the algorithm leads to form variations.

Many of these approaches focus on form development and the search for exciting artistic expression. Therefore, they do not need an elaborated conceptualisation of a building model (such as in the case of BIM) and as a

---

[10] Application of shape grammars in CAAD: (Dounas and Anastasios, 2006; Chen, 2007; Mei, McKay et al., 2006)

result they might lack analysis module. Unfortunately, their focus is narrow (forms generation) and they serve more as generators of artistic forms than comprehensive generative design systems.

Kostas Terzidis (2006) proposes yet another approach to the generative design. Terzidis argues that the creative process of design must be based on *computation* rather than computerization. Computerization is a relatively static process of inputting predefined data and it is characteristic of CAAD systems that merely automate the drawing tools. On the other hand, computation takes advantage of a machine's capacity to analyse a very large amount of data. This computational power makes it possible to control (e.g., by means of analysing, constructing and visualising) architectural objects that go beyond human conceptualization.

Moreover, if algorithms should produce unpredictable results then one should let randomness affect the design. To a certain degree, the algorithm should be a kind of a 'black box' – it should collect a set of design intentions as input, process them, and eventually provide an unexpected design as output. The problem discussed above leads to further questions: Should computational creativity be controlled? If yes, to what extent? How to attain control over computational generation process that is beyond human comprehension?

Another approach to digital creativity represent design systems, which apply recent developments in Artificial Intelligence, such as: expert systems, systems based on case-based reasoning, systems based on artificial neural networks, and systems based on fuzzy logic.

An expert system is software that simulates the performance of a human expert. It is 'equipped' with a database containing formalised knowledge on a specific domain. The expert systems have the ability to extend the knowledge base by attaining and codifying knowledge of human experts (optionally, the expert systems might attain knowledge through a learning component). The expert systems function in a context of 'real-world' problem solving, as users' assistants or as components of larger information systems[11].

A case-based reasoning is another approach. It consists of solving new problems using knowledge acquired in previous similar situations. The idea of case-based reasoning is not only a technique associated with computer science, but it finds its broader application, for instance in everyday human problem solving (human experience is built on past cases). The case-based reasoning is related to a prototype theory, which posits that some members of

---

[11] For an application of expert system in CAAD, see: (Oxman, 1988; Rosenman, Radford et al., 1990; Coyne and Newton, 1989; Calvo, 1993). For an application of case-based reasoning in CAAD, see: (Lee, 2003).

a given category (of objects or problems) are more central, typical or frequent than others. Thus, solutions of problems that are less usual can be based on solutions of typical (prototypical) problems.

Computational models of networks, based on biological neural networks, are called artificial neural networks. Reflecting the neural structure of the human brain, where neurons are interconnected with axons and dendrites, the artificial neural networks consist of a structure of interconnected elements that processes information. A learning phase plays a substantial role in the artificial neural networks. In this phase, the network undergoes an adaptation or 'setting up' process, in order to make it apt to a planned purpose. Artificial neural networks may be applied, amongst others, to data processing and pattern recognition[12].

A fuzzy logic is a multi-valued logic; it is based on fuzzy set theory. Different from a classical predicate logic, which is based on either true or false statements, fuzzy-logic bases reasoning on the evaluation of a statement. The degree of truth in fuzzy logic can vary from 0 to 1; it is therefore a continuous rather than a dichotomous measure[13].

Finally, this fourth group of CAAD systems include the generative design system proposed by me. The generative design system developed in this thesis involves evolutionary algorithms that combine randomness with deterministic methods. Evolutionary algorithms offer a number of programming tactics that are apt to design. I argue for appropriateness of evolutionary algorithms to design in section VIII.4.

The whole generative design system that I develop works essentially as a black box, though it includes elements of a white box. A user controls design development by its characteristics that are inputted into the system at the outset of the generation process. A product of the generative design system is a building model that fits to a possibly high degree with the inputted set of characteristics. A compressed explanation of how the generative design system works is included in section X.1. The discussion about the system in the context of the concept of a black box is included in section X.2.

---

[12] For an application of artificial neural network in CAAD, see: (Coyne and Newton 1989; Huang 2000; Flood and Christophilos, 1996; O'Neill and Michael,1992)

[13] For an application of fuzzy logic in CAAD, see: (Chen and Chiu, 2006)

SECTION III – SYSTEMS THEORY AND COMPLEXITY SCIENCE

This section is an introduction to systems theory, and to the more recent research domain of complexity science. It gives an overview of different definitions and properties of a system. These definitions and properties will be applied in the later sections of the thesis for the development of elements and principles of the generative design system. Systems theory offers methods, which are relevant for approaching architectural design. Specifically, the problems of architectural design are similar to problems of systems theory in the following areas:

- expertise form various disciplines is involved (interdisciplinarity),
- there is a number of objectives that might rule out each other,
- it is impossible to use linear approach for finding a solution (very large solution space) and
- a solution might partly define a problem (feedback).

I use the notion *system* in two contexts that should not be confused. Firstly, the term can refer to the generative design system as a whole – a computer program for generating building models. Secondly, in more specific meaning, the term system refers to one element of the generative design system – a building model.

By a building model I understand a representation of an intended building. The thesis postulates that a building model should be understood as *a complex system that adapts to design objectives (a building environment)*. A similarity of a building model and a system as defined in this section is evident especially in two regards.

- A building model is an *integrated entity* and it has to be approached accordingly (one property of a system is irreducibility). One cannot design a function of a building, its acoustic comfort, its construction and its form separately and then put them together in one building model, because each aspect affects others.
- A building model develops in a certain direction, dictated by design objectives. It is a *goal-directed* entity. Even if design objectives are redefined in a process and the direction of a building model development shifts, still the new design objectives define the direction of the model's development.

In addition, this section discusses a difference between natural and artificial systems. While a natural system is an effect of natural laws, an artificial system is an effect of deliberate human actions. However, the latter is subjected to natural laws too. Its development is a combination of an intention (a human-defined purpose) and a requirement (subordination to laws of physics).

Finally, the section investigates the environment of a system: it begins with a definition of the environment, it discusses environments of natural systems and artificial systems and it examines a system's boundary. Final paragraphs show how different definitions of a system's boundary can be applied to different elements of the generative design system.

1. WHAT IS SYSTEMS THEORY

Ludwig von Bertalanffy identified two essential features of systems theory. Firstly, it is transdisciplinary; secondly, it is a science of systems – indecomposable entities or 'wholes'. According to Bertalanffy, there are general aspects, correspondences and isomorphisms that are common to very different systems. Even though such disciplines as physics, biology or psychology are 'encapsulated' in the boundaries of their domains, and communication between them is limited, independently of each other, similar concepts and problems appeared. Examining these concepts, problems and correspondences is a subject of general systems theory (Bertalanffy, 1968: xx). The problems that appeared in one discipline have led to patterns of solutions that were applicable in other disciplines. The transdisciplinary character of system science finally found its more precise, mathematical formulation in computer science. According to Weinberg (2001: 46), the systems theory focuses on three activities:

1. General systems thinking, which is about methods and approaches;
2. General systems application;
3. General systems research, which is about creating new laws and refining old ones.

SYSTEMS THINKING

The systems thinking is a framework of thinking or an approach to thinking about systems. It is based on the principle that systems are wholes and that their elements should be examined in terms of their relationships with each other.

There are two central approaches within the systems theory: *soft systems* and *hard systems*. It is worthwhile to introduce these approaches, because architecture design includes elements of them both. A generative design system specifically, should be able to tackle problems typical for both soft and hard systems.

Soft systems are difficult to quantify, and they are often associated with social systems. The methodology of soft systems is aimed at providing a framework for tackling real-world situations, the ‘messy’ situations involving psychological, social and cultural elements – so called ‘wicked’ problems. Hard systems on the other hand involve well-defined problems – these that have a single, optimum solution, that can be approached with analytical methods and these, in which technical factors tend to dominate. They are easier to quantify and their methodologies often use computational examination methods such as simulations, systems modelling, systems analysis or optimisation methods.

The essential difference between soft systems and hard systems lies in the definition of a system. The hard systems’ approach is ontological – it defines systems as objects existing in the real world or bounded entities with a physical existence. The soft system’s approach is epistemological – it understands systems as mental constructs. In this approach, a system is conceptualised from many perspectives, each of them providing a different understanding of it (e.g., the same man can be considered as a freedom fighter or as a terrorist). Soft systems investigate problems that are complex, i.e., where nonlinear relationships, feedback loops, hierarchies and emergent properties have to be taken into account. The soft system’s method postulates understanding of a system, by iterative learning process.

The elements of these two approaches are discussed more in-details in a number of places in the thesis, for example in the contexts of a design process in section IV, or in the context of building characteristics in section VI.

One of methods applied within system thinking is system dynamics. System dynamics is universal approach, i.e., it is applicable to both hard and soft systems. Sterman (2000) defines system dynamics as follows:

> System dynamics is a method to enhance learning in complex systems. Just as an airline uses flight simulators to help pilots learn, system dynamics is, partly, a method for developing management flight simulators, often computer simulation models, to help us learn about dynamic complexity, understand the sources of policy resistance, and design more effective policies. (Sterman, 2000: 4)

According to system dynamics, the complexity of the real-world systems exceeds human capacity to anticipate the implications of these systems. Human beings operate within a complex network of positive and negative feedbacks, which are extremely difficult to model mentally. In examining complex systems, one needs to take into account phenomena such as time-delays, poor reasoning skills, defensive reactions and the like. Because mental simulations of complex behaviour are unreliable, more formal simulations, especially computer simulations, are useful.

SYSTEMS APPLICATION

Although systems theory was initially aimed at embracing a wide array of disciplines, it has a tendency to 'dissolve' when it is used in the context of a specific application. Mechanisms investigated by the systems theory are abstractions, and it is as such that they can be studied. Applied to specific problems, the mechanisms have to be carefully adjusted.

For example, the notion of natural evolution in its abstract, simplified form can be applicable to a large number of phenomena in many disciplines. Broadly understood, the concept of evolution can be applied to language (e.g., how spellings have changed over time), society and culture (e.g., from simple to complex kinship systems), individual human beings (e.g., developmental changes). When the notion of evolution is applied to stellar development for instance, it both reduces its meaning (as it does not involve natural selection) and it extends it to a specific content (the description of a star's life phases). The phases in which, through gradual warming, a star become a red giant, a white dwarf and a black hole are not constrained by natural selection, but by the initial mass of a star. Alternatively, applying the laws of biological evolution to society and culture, one can predict that stronger, more flexible or more fitted cultures survive. But the risk is that such a statement oversimplifies the intricacy of cultural phenomena.

The generalized mechanism of natural evolution is applied also in Evolutionary Computing as a general problem solving mechanism. In section VIII I applied this mechanism to the development of a building model.

SYSTEMS RESEARCH

General system research initially aimed at finding a central, unified theory (a general system theory), which could explain the behaviour of all systems in different disciplines. Nowadays, the aim of the general system theory is probably less ambitious. General system research transformed into field called complexity science, whose research interests, though interdisciplinary and broad, focus on mathematics and natural science.

Systems research – in terms of developing transdisciplinary principles or laws – is intertwined with systems application. Systems research affects the investigation of specific phenomena and reversely the specific application modifies the understanding and formulation of general mechanisms.

For example, the observation of a natural population growth led to a development of a mathematical model called 'logistic map'. This mathematical model was intensely examined, and resulted in the elaboration of interesting, new properties of systems, such as 'deterministic chaos'[14]. The general mechanism of deterministic chaos can be in turn applied to other phenomena, such as dynamics of weather (a popular 'butterfly effect'), economic and social systems dynamics, or bouncing ball dynamics.

Evolutionary Computing is an example of natural selection understood in the abstract sense. This computing technique arose from examination of biological phenomena – first represented mathematically, and then applied to computer science. As such, the mechanisms of EC are the subject of distinctive research. At the same time, EC is a method for solving actual problems and have specific applications.

---

[14] One of these remarkable properties of the 'logistic map' equation is that even though it is simply formulated and deterministic, there is certain range of parameters for which the equation produces series of 'random' numbers. Moreover, the equation yields different strings of random numbers for only slightly different initial parameters. This property has been called a 'deterministic chaos' (Mitchell, 2009). In fact, whatever the initial parameters were, the generated string of numbers was calculable (deterministic), but even a miniscule change in the initial parameters would result in a significant change in the generated numbers (chaos).

2. DEFINITIONS OF A SYSTEM

SYSTEM AS A SET OF RELATED ELEMENTS

When speaking of groups of elements, Bertalanffy makes three distinctions: first, concerning the number of elements; second, concerning their species; and third, concerning their relations. Bertalanffy calls the characteristics of the first two groups *summative*, and the characteristics of the third group *constitutive*. The constitutive characteristics "are those which are dependent on the specific relations within the complex; for understanding such characteristics we therefore must know not only the parts, but also relations" (Bertalanffy 1968: 54). The characteristics cannot be derived from a set of isolated elements, but they are an effect of a whole group of interacting elements (the complex). System here is defined as an assembly of such related elements.

If one assumed that a building is a system, then each of its elements should be a 'constitutive' element. Because the elements are purposefully composed, they are related to each other and to the whole building and their characteristics come from these relations.

SYSTEM AS AN ORGANISED COMPLEXITY

Weinberg (2001: 19) proposes another definition. He divides objects of examination into three groups, or regions. The first region ('organised simplicity') contains objects that he calls machines – they are organised in a way that is accessible to analytical procedures (the so-called scientific method initiated by Galileo and Descartes). Representatives of this group are artefacts or objects created by humans, such as a car, a mobile phone,[15] or models of some natural phenomena (e.g., a model of the solar system).

The second region ('unorganised complexity') contains objects or phenomena that are relatively unorganised and that include a vast number of elements (aggregates), like unorganized group of people. This property makes them very intricate subjects for examination, and they cannot be examined with analytical procedures, but with statistical methods.

Lastly, and most importantly, there is an intermediary region ('organised complexity'), which consists of number of objects that are organised, but whose organisation is too complex for analytical procedures. Weinberg calls

---

[15] The problem is more complicated. A car for example, cannot be reduced to an object, which can be examined only through analytical procedures. It consists of both quantitative characteristics (consumption of fuel, power of engine and the like) and qualitative characteristics (level of comfort, visual quality). The latter characteristics are relative to the car's purpose and to some extend they depend on human judgment.

these objects *systems* and they include for example the atmosphere or the living organisms (figure 2).

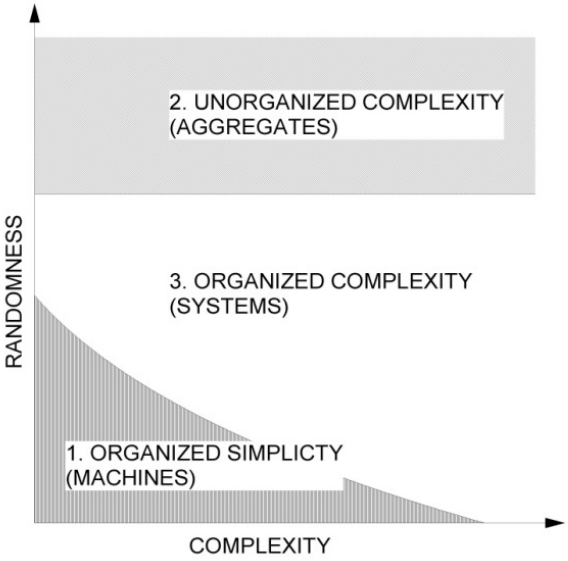


Figure 2. Weinberg's classification of objects and phenomena. The vertical axis (randomness) reflects the number of relations between objects – the more random an object, the less relations between its elements. The horizontal axis reflects the complexity of relations (after Weinberg, 2001).

In Weinberg's classification, the interdisciplinary character of the systems theory is apparent. When it comes to the first region, the machines are products of combined knowledge from different disciplines. Similarly, the representatives of the second region, the aggregates, can be found both in the social sciences and in the natural sciences.

The difference between Bertalanffy's and Weinberg's definitions of a system is evident. Bertalanffy's definition is broader, because the only criterion for a system is that its elements are related. For Bertalanffy, a system is everything that makes a whole including objects from Weinberg's first region, i.e. machines and mechanisms. Weinberg's systems include only objects that are too complex for the analytical procedures. The examples of Weinberg's systems include natural systems (such as the atmosphere), biological systems (such as living organisms), or men-made conceptual systems (such as economic systems or legal systems).

Weinberg emphasizes the potential lack of procedures applicable to systems. The application of the acknowledged methods, such as statistical and analytical procedures is inadequate to organized complexities. In the statistical procedures, simplicity is achieved by operating on averages, while in analytical procedures, the number of relevant parts of the system is purposefully lessened. Both procedures result in oversimplification of the examined phenomenon.

The lack of a relevant approach to complex systems was a major reason for initiating systems research. The careful examination of systems gave rise to the development of new methods, which are alternative to the traditional, analytical procedures of natural science. This thesis assumes, that a building model is such an organized complexity (a system in Weinberg terms) and thus, methods developed by systems research find application in the generative design system.

SYSTEM AS A WHOLE

Laszlo (1996) differentiates between two fundamental modes of thinking in the European tradition: atomistic (rigorous) and holistic (speculative). In the early scientific thinking, holistic thinking prevailed. The holistic thinking then gave way to more rigorous, empirically testable knowledge. But the latter, though rigorous, were fragmentary (atomistic) and it lost its original coherence. The two modes of thinking alternated in the past and eventually transformed into today's mode of thinking which is "rigorous, yet holistic".

In this context, Laszlo mentions an early definition of systems as conceptual entities, which we used to organise our knowledge about the world. According to Laszlo, a system used to be something that exists in our mind and not in the world, such as a 'theological system' or a 'system of logic'. Nowadays we call systems many things whose existence is independent of our thinking – a political system, an economic system, a social system, an ecological system, a biological system, an astronomical system or a computer system.

Laszlo defines systems by distinguishing '*wholes*' and '*heaps*'.

> 'Wholes' and 'heaps' are not mysterious metaphysical notions but clearly, even mathematically, definable states of complex entities. The decisive difference is that wholes are not the simple sum of their parts, and heaps are. Take, for example, a pile of rubbish. Adding another can or removing a pop bottle makes only a

> quantitative difference to the pile – it becomes that much bigger or smaller. No other characteristic of it changes. (Laszlo, 1996: 25)

In contrast to a heap, a whole is a structured relation between two or more elements. The characteristics of the whole cannot be derived from the characteristics of its parts. The characteristics of the whole come about with a particular and exact relation (organisation) between the parts. Laszlo gives examples of wholes:

- Friendship, were the relation of two persons is something more than the sum of their separate relations;
- An atom, which properties are not reducible to the sum of the properties of its constituents.

Unlike the whole, the heap is an assembly or mass of unrelated elements. The heaps are similar to the objects from Weinberg's second region – unorganised complexities, or aggregates. Referring to the Bertalanffy's distinction into the summative and the constitutive characteristics, a stone taken away from a pile of stones would not change the constitutive characteristic of a pile, because it was not 'in relation' with other stones and had no 'function'. It would change the summative characteristic of the pile – its weight, but it would not change its constitutive characteristic – the pile would still be a mass of unrelated stones.

SYSTEM AS A MODEL

Bertalanffy offers also a definition of a system that is narrower that "elements in relation". According to this alternative definition, a system may be understood as a model – a representation of some universal traits of a class of natural phenomena (Bertalanffy, 1968: 251). Such a system would be a conceptual construct, serving either as an explanation of a phenomenon (e.g., a model of a small community) or as a method of predicting the 'behaviour' of a phenomenon (e.g. a model of a hurricane, a model of a solar system). Compared to the earlier definitions, this one narrows down the range of potential systems. Firstly, it limits systems to conceptual constructs – especially those that are defined by the scientific method. Secondly, this approach excludes artefacts, such as cars, washing machines or computers, which Weinberg calls machines.

For Bertalanffy, a model involves an act of abstraction (selective removal of irrelevant elements or factors), and an act of interpretation (situating elements of observed phenomena into a network of known phenomena). For example, a model of an atom is meant to be a representation of a real phenomenon. But it is as well a construction, which is apparent when one considers the development of the model of atom. Bohr's model depicts an atom as consisting of electrons circulating around the nucleus. In this model, the reference to the conventional model of a solar system is apparent, and the only difference is that the gravitational forces were replaced by electrostatic forces. The conventional model of a phenomenon (the solar system) has been applied to another phenomenon (the atom).

Models of artefacts are not considered by Bertalanffy. But they too combine abstraction (as they consist of generic building elements) with construction (as they combine these elements in new ways). A new design of a car is a new configuration of the generic components, such as elements of engine, wheels, bumper, seats, doors, windshield, radiator, which in turn are assemblies of smaller, more basic elements. Nonetheless, there is a fundamental difference between the models of artefacts and the scientific models. The models of artefacts do not serve as an explanation of a phenomenon or as a method of predicting the 'behaviour' of a phenomenon. They are creations, not representations – they are meant to guide making of a new physical object, not to represent an existing one. A building model is an example of such an artefact. It consists of building elements, that are representations of real entities, but their configuration is new.

SYSTEM AS A GOAL-DIRECTED IDENTITY

A teleological definition of a system focuses on two aspects: the preservation of the system's identity (despite transformations), and its goal-directedness (Mitchell, 2009: 297). Weinberg (2009: 251) also claims that "the permanence of the relations among component parts", is an essential feature of a system. The identity of biological organisms is protected by sophisticated mechanisms, specifically "the elaborate arrangements are made to protect the germ plasm from such change [mutation] and to nullify its effect if it does occur" (Weinberg, 2009: 251).

The evolution of organisms can be understood as a progressive change, where each subsequent state of the organism is an effect of all the previous, less successful states. Such understood evolution implies continuity of an organism and thus, preservation of its identity.

A design process, as a rule, is a goal-oriented process. The goal is specified at the outset of the design process by design objectives. The design objectives, as long as they are relevant and followed, define design's identity. Each modification of a form of developed building model is a record, or a memory, of a series of earlier states, in a general development tendency towards the design objectives.

3. DEFINITION OF A COMPLEX SYSTEM

As the systems theory gave place to complexity science, the term 'system' transformed into 'complex system'. Researchers of complex systems acknowledge that there is not one science of complexity, but different 'sciences' that all have their own notion of a complex system and complexity (Mitchell, 2009). Therefore, there is no unequivocal definition of a complex system and there is no established method for measuring the level of the system's complexity. In spite of these different approaches and definitions, Mitchell discerns some common properties of complex systems:

- Complex collective behaviour (a high number of components triggers complex patterns of behaviour);
- Signalling and information processing (systems produce and use signals from internal and external environments) ;
- Adaptation (systems change their behaviour to improve their chances of survival through, e.g., evolutionary process).

Furthermore, Mitchell defines a complex system as "(…) a system in which large networks of components with no central control and simple rules of operation give rise to complex collective behaviour, sophisticated information processing, and adaptation via learning or evolution" (Mitchell, 2009: 13).

A crucial feature of a complex system is the difficulty to predict its behaviour. A *chaotic* behaviour of complex systems consists in the fact that even minuscule uncertainties in the initial state of a system (measurement of initial positions and momentum of its elements) can result in large errors in predicting its subsequent states. This phenomenon is known as "sensitive dependence on initial conditions" (Mitchell, 2009: 20). Understanding of the mechanism of chaos, 'buried' the hope of perfect prediction of complex systems' behaviour.

Sensitive dependence on initial conditions is not necessarily an effect of a large number of elements and connections. Complexity can also emerge from simple rules. Sensitive dependence on initial conditions takes place even when investigating three simple bodies: "to determine, using Newton's laws, the long-term motions of three masses exerting gravitational forces on one another" (Mitchell, 2009: 21). Newton solved the two-body problem, but the three-body problem turned out to be much harder.

Some scientists regard the complexity of a system as a key factor in its description. Seth Lloyd (2001) considers three different questions in defining a complex system:

- How hard is it to describe?
- How hard is it to create?
- What is its degree of organisation?

The first question is challenging, because random or chaotic systems are the most difficult to create or describe. Indeed, the lack of regularity or recognizable pattern makes the description of such systems difficult. To overcome this difficulty, physicist Murray Gell-Mann proposed an alternative definition of complexity. He developed a measure called 'effective complexity', which fits well our intuitive understanding of complexity. A basic assumption of effective complexity is that complexity is always a combination of regularity and randomness. Gell-Mann (1995) gives as an example the DNA of a living organism, in which regularities coexist with randomness (called 'junk DNA'). According to Gell-Mann, the effective complexity can be measured in two steps. First, by figuring out the best description of the regularities of a given system; second, by defining the amount of information contained in this description. The complexity of a system would then be the amount of information contained in the description of the system's regularities.

Yet another definition of complexity was proposed by Herbert Simon. In Simon's definition, complexity is related to the hierarchic degree of a system. Hierarchy is a universal, common feature of all complex systems. He defines hierarchy as a composition of interrelated sub-groups within a larger entity (Simon, 1962: 468). Simon speaks also of a 'span of control'. In formal human organisations, for example, a span of control is specified by a number of subordinates who report directly to the manager. The term span of a system denotes the subsystems that compose the system as a whole.

The hierarchical construction of a system is significant, because it gives a system an *advantage* in the evolution process. An organism (or an artificial system) undergoes permanent reconfigurations in the process. Stable subsystems facilitate a faster reconfiguration of the organism, because the organism does not need to be reconstructed from its basic elements: "The time required for the evolution of a complex form from simple elements depends critically on the numbers and distribution of potential intermediate stable forms" (Simon, 1962: 471). In natural evolution, the genetic modification involves the subsystems of an organism (cells, tissues, organs) rather than its basic elements (organic compounds). Applying this principle to car design one can say that a blueprint of a car is developed hierarchically, at the level of its components, and that the designer goes down to its basic elements and redefines them only if needed. Even if a single component is redesigned, the operation is carried on roughly within the boundaries of this component (subsystem). For example, designing a seat of a car, there is no need to go much beyond the boundaries of this particular seat (as long as the size of the seat is correct). One can say that because the plan of a car is conceptually divided into subsystems, it is easier to manipulate the whole system during a design process. An adjustment in a subsystem does not require a total decomposition of all the elements, but rather decomposition of the subsystem in question. In sum, even though a system is a whole, the density of relations between its elements is not distributed homogeneously.

Daniel McShea (2001) elaborated Simon's idea further. McShea observed that the complexity of organisms increases over time in an evolutionary process. He proposed a scale of hierarchy as a measure of the level of an organisms' complexity. In this context, he coined the term *nestedness*: "a higher-level entity contains as parts entities from the next lower level." Each level is more complex that the previous one.

Both Simon's and McShea's ideas are relevant to the approach presented in this thesis. Hierarchy as an essential feature of a complex system will be further discussed in section V.

In his paper *Life and Complexity in Architecture from a Thermodynamic Analogy*, Nikos Salingaros (1997) presents an interesting approach to complexity. This approach is worth mentioning because it addresses the context of a building form. Salingaros proposes two building qualities, with which he defines complexity: a temperature T, and a harmony H. In broad terms, "The architectural temperature T is defined as a degree of detail, curvature and colour in architectural form, whereas the architectural harmony H measures the degree of visual coherence and internal symmetry in the visual structure" (Salingaros, 1997: 87). Temperature and harmony are related, so that usually more harmony results in 'lower' temperature. For example, reducing colour differences in certain areas increases the building's harmony while at the same time it reduces the contrasts and thus the temperature. Salingaros applies the two terms to define complexity in the following formula:

$C = T \times (10-H)$, where T and H are between 0 and 10

This formal definition of 'C' reflects an intuitive understanding of a building's complexity, or a quality that "arouses a viewer's interest (…) the inverse measure of how boring a building is" (Salingaros, 1997: 99). Salingaros speaks of two types of complexity: disorganised complexity and organised complexity, where C is a measure of the 'disorganised complexity'. The notion differs from a common understanding of complexity as a disorganised variety. Biological organisms, for example, are highly complex but they are not disorganised.

Salingaros' notion of complexity is derived from yet another key term in his model: 'life'. The quality of 'life' of a building is formulated as a product of a building's temperature and its harmony. The two notions of complexity and life seem to be very relevant to the algorithmic description of an architectural form and are discussed further in the context of building characteristics in section VI.

4. IRREDUCIBILITY OF A SYSTEM

A distinctive feature of a system is its irreducibility. Scientific disciplines such as chemistry or biology isolate the elements of the investigated complexity (chemical compounds, enzymes, cells, etc.), "expecting that, by putting them together again, conceptually or experimentally, the whole of system would result and be intelligible" (Bertalanffy, 1968: xix). This approach might be considered as reductive, because isolating elements of a whole deprives the whole its essential properties, which arise precisely from the relations of the elements. Again, the systems theory suggests a different approach to object examination – the emphasis moves from the elements to their interrelations, for example from a cell's structure to the interplay of enzymes in a cell.

Bertalanffy asserts that to understand a system, one must accept that it is irreducible, and that this irreducibility applies to systems regardless of the discipline. He claims that the relations between the system's elements require extra attention from the researcher, in addition to the 'standard' procedure of investigating the parts in isolation.

Weinberg distinguished systems from mechanisms because analytical procedures do not apply to systems. Bertalanffy makes a similar point on the irreducibility of systems. Because the analytic approach of natural science can only operate on a limited number of variables at the same time, it needs to 'decompose' complex objects (systems) before being able to draw conclusions:

> The system problem is essentially the problem of the limitations of analytical procedures in science. This

> used to be expressed by half-metaphysical statements, such as emergent evolution or 'the whole is more than a sum of its part', but has a clear operational meaning. 'Analytical procedure' means that an entity investigated be resolved into, and hence can be constituted or reconstituted from, the parts put together, these procedures being understood both in their material and conceptual sense. (Bertalanffy, 1968: 18)

But, what exactly makes a system 'examinable' with the analytical procedures? Bertalanffy specifies two conditions under which analytical methods can be applied. First, the interactions between parts of a system must be weak enough to be neglected. "Only under this condition, can the parts be 'worked out' actually, logically, and mathematically, and then be 'put together'" (Bertalanffy, 1968: 19). Secondly, relation describing the behaviour of a part must be linear. Only then an equation describing the behaviour of the whole is of the same form as the equations describing the behaviour of the parts.

Classical science is successful within the range of systems with linear dependencies between elements. One can assume that the more elements there are in a system, the more probable it is that their behaviour is not linear. For example, natural sciences, such as physics, tend to reduce the number of elements of investigated system to the relevant ones, which makes the analytical method applicable. The situation is more difficult in biology and even more in sociology, because the number of relevant elements in these systems is very large and the relations cannot be simplified – the system cannot be divided into examinable parts.

In order to use the analytical procedures in biological and social science, one has to neglect or minimize relations. One has to 'hide' the integrity of a system and exaggerate the apparent independence of some of its parts. This method makes relations only 'provisionally' linear:

> We divided the body into its organs, the skeleton into its bones, as in very much the same fashion we make a subjective analysis of the mind, according to the teaching of psychology, into component factors: but we know very well, that judgment and knowledge, courage or gentleness, love or fear, have no separate existence, but are somehow mere manifestations, or imaginary coefficients, of a most complex integral. (Weinberg, 2001: 21)

In the area of design, natural sciences blend with social sciences (or hard systems with soft systems). In particular, a design of a building incorporates elements of knowledge including physics (structure calculations, thermal insulation, energy-performance, acoustics and the like), economics (cost of construction, cost of maintenance), and social sciences (impact on local community, users' response, adequateness to the intended function, contribution to the local identity, and even a potential effect on criminal activity). The method of 'provisional' simplification of relations is typically used in design when a part of a building is 'extracted' and redesigned. Redesigning a part of a building does not necessarily mean envisioning all the implications of the intervention on the whole building. It is only gradually (when the change is carefully examined in the context of the whole building) that the consequences are recognized and that one can realise the outcome of the change.

Buildings are wholes and to some extent they are irreducible. Architects often refer to a building's form in its totality, and derive its global characteristics by using adjectives such as dynamic, static, open, closed, coherent, random, ordered, intrinsic, simple, etc. An attempt for careful examination of the individual building parts would not say much about the overall building shape. Two static forms put together could make a dynamic form as well as a static one. A designer cannot judge the effect of the whole assembly just by examining its individual parts.

Concluding, a system is more than the linear sum of its elements, and it is very difficult or often impossible to understand a system by examining its parts individually. The system's properties are difficult to anticipate, and they are a subject of observation and explanation rather than anticipation. One understands the properties of a system gradually, through the observation of the performance of the whole system.

Considering this, I suggest a generative, incremental approach to building model generation. Properties of a generated building model should be carefully examined and relevant changes should be introduced gradually, in an adaptive way.

5. NATURAL AND ARTIFICIAL SYSTEMS

According to Simon, a fundamental difference between artificial and natural systems is that natural systems are not designed by human beings and therefore are an effect of natural laws (which would link them to inevitability), whereas artificial systems are an effect of deliberate human actions (which would link them to possibility) (Simon, 1981: xi). This does not mean that artificial systems are not subordinate to natural laws, but that their construction is constrained by factors such as human conception, ingenuity and – above all – human needs and desires. Artificial systems therefore have a teleological character, and it is humans who define their purposes. This purpose is the rationale for constructing an artefact. Because artefacts satisfy human desires, they are adapted to man's goals and purposes and "as man's aims change, so too do his artefacts" (Simon, 1981: 6).

Design is an activity dealing with synthesis, i.e., composing objects that have specific desired properties. In contrast, natural science is an activity concerned with analysis, i.e., making account of existing natural phenomena or non-artificial objects. According to Simon, "The engineer, and more generally the designer, is concerned with how thighs *ought* to be – how they ought to be in order to *attain goals*, and to *function*. " (Simon, 1981: 7)

Synthesis is making wholes from elements and by investigating these constructed wholes, a designer can say if they work according to his or her intentions and desires. The purpose of the artificial system constrains the way it is constructed. For example, a car is intended to be a means of transportation and it is this purpose that 'governs' its construction. The elements of an artificial system are put together in such a way as to 'utilize' the natural laws and make them work for human purposes.

On the other hand, natural science conceptually deconstructs existing wholes – natural systems – and by investigating their elements, it tries to understand these wholes. It does not try to make use of natural laws, but it tries to derive them from examining (manipulating) natural systems.

In the context of the generative design system, a development of a building model would have to be activated by a 'purpose', or a 'target' defined by a user. At the same time, the development would be constrained by feasibility of a generated building model. So, the generation process would combine 'possibility' (human defined purpose) with 'necessity' (subordination to natural laws).

6. THE ENVIRONMENT OF A SYSTEM

The following paragraphs explore the notion of environment in the context of Systems Theory. First, I define environment of a system in general terms. Then I examine two types of systems – natural systems and artificial systems, trying to identify what their environments consist of. Based on this examination, I make an important distinction: the environment for natural systems development is a *physical* reality, while the environment for artificial systems development it is a *mental* reality. This conclusion has a significant implication for the generative design system: it is a building *model* that adapts, not a *physical* building.

Finally, I discuss a boundary of a system – a space between an 'inside' and an 'outside' of a system. My intention is to outline a scope of the generative design system and its elements. I propose three definitions of a boundary and I show how they can be applied to the generative design system.

DEFINING THE ENVIRONMENT OF A SYSTEM

Generally speaking, the environment of a system is coterminous with the reality outside the system. The definition of a system as a set of interconnected elements can be supplemented in a following way: "A system may be defined as a set of elements standing in interrelation among themselves *and* with environment" (Williamson et al., 2003: 82). This extended definition incorporates connections between the elements of a system with the reality outside – the environment. But even though the elements of the system interact with the environment, their internal connections are stronger.

One can discuss three kinds of systems in terms of their interaction with the environment: open systems, isolated systems and closed systems. The definition of a system by Williamson et al., quoted above, refers to the open systems, i.e., systems that interact with their environments (by exchanging energy, matter or information with the environment). Conversely, the isolated systems do not exchange energy, matter or information with the outside world, and thus, the elements of the isolated system do not interact with the environment. The isolated systems do not exist in physical reality (except the universe itself); they are only useful concepts, approximations of real phenomena. An example of such a conceptual isolated system can be a model of our solar system.

The notion of a closed system appears in thermodynamics and it denotes a system that exchanges energy but not matter with the environment. One example is a fluid compressed by a piston in a cylinder. There is always the same amount of fluid, but heat (energy) is exchanged across the boundary of the cylinder.

Throughout the whole thesis, only open systems are discussed. Both a building model and a whole generative design system are open systems, because in both cases there is an exchange of information between these systems and their environments. A building model exchanges information with design objectives defined by a user (which refers to the concept of adaptation). A generative design system exchanges information every time a user redefines design objectives.

ENVIRONMENTS OF ARTIFICIAL AND NATURAL SYSTEMS

Natural adaptive systems (for instance living creatures) exist in certain physical environments. Their interaction with the outside world consists in exchanging information through their sensors. They either passively receive impulses from the outside world, or they act and then read the response from the outside. In this process of information exchange, the environment affects the organism, causing the organism to adjust some of its features, or its behaviour, in order to become fitter.

There is yet another, more fundamental way the living organisms interact with the environment. This kind of interaction affects generations of individuals, and it results in significant structural transformations of the organisms. In this case, the exchange of information occurs through a phenomenon called natural selection. Natural selection makes that the better-fitted organisms live longer and thus can proliferate more successfully in their environments, which leads to survival and development of their species. The development can occur, because the offspring of the better-fitted organisms are always slightly different from their parents and it can happen that some of them are even better fitted than they parents. Because all the offspring are subjected to the process of natural selection, only the best fitted survive and proliferate giving birth to still more fitted individuals. If the procedure lasts long enough, the structure of the organisms can change profoundly (Darwin, 1859). In shorter time-spans, the offspring preserve the basic features of their predecessors[16].

---

[16] This process is often referred to as autopoiesis (as opposed to allopoiesis). The process of autopoiesis (self-production) happens in the systems that maintain their integrity despite the fact that their constituents are continuously exchanged with the outside environment. In other words, an autopoietic system is “a circular organization which secures the production or maintenance of the components that specify it in such a manner

When it comes to artificial systems, or specifically human produced objects – they exist in the physical environment as well. Some artificial systems are equipped with sensors, imitating the architecture and behaviour of natural systems. These artificial systems – like robots – are capable to exchange information with the environment. Although there are no artificial systems that undergo a natural selection process in a physical environment, one can speak of a similar phenomenon. Considering cars, for example, the evolution of their form is apparent. The form of a car responds to the actual (but not physical like in the case of living organisms) demands: economic (e.g., efficient combustion), aesthetic, functional and other. Some of these demands alter with time and the shape and construction of a car transform to meet the new expectations:

> Under pressures from within and without, the 'systems' change with the times. Those that do not are left as ossified relics of the past. Inputs from within and without call forth innovations, and the innovative system produces new kinds of inputs (…) (Laszlo, 1972: 65)

What Laszlo calls 'inputs from without' and later 'innovation call' is a pressure of the environment on the system to transform. Simon puts it in a very similar way: "The artificial world is centred precisely on this interface between the inner and the outer environments; it is concerned with attaining goals by adapting the former to the latter" (Simon, 1981: 132). The artificial systems change in parallel to socio-political or cultural transformations, which initiate new social standards and new expectations – generally meaning new values. These values set the direction of systems' transformation:

> Values are goals which behavior strives to realize. Any activity which is oriented toward accomplishment of some end is value-oriented activity. (…) There is nothing in the sphere of culture which would exempt us from the realm of values – no facts floating around, ready to be grasped without valuations and expectations. (Laszlo, 1996: 79)

---

that the product of their functioning is the very same organization that produces them" (Maturana and Varela, 1980: 48). See also: http://www.christianhubert.com/writings/Autopoesis.html

That said, in what kind of space do artificial systems evolve? It seems that the transformations of the artificial systems occur in the mind of a designer and that it is a mental construct. A physical artefact is a realisation of a mental model, and it is actually the model that undergoes transformation. The environment for this evolving model consists of socio-cultural demands, intentions and expectations. Accordingly, the artificial system is a physical realisation of a mental model, evolving in a mental environment.

When an artefact of everyday use no longer meets the expectations, it is replaced by a new, more adjusted artefact. For example, when a given model of a car consumes too much fuel, produces too much pollution, or becomes old-fashioned, it is redesigned. It does not transform itself ‘automatically’, i.e., without human deliberate action, in a way living organisms do. It is a designer, who makes a plan of a new model, hoping that the new one would better fit the new expectations. The environment for developing the plan of a car is based on the expectations of a certain group of clients, and their expectations in turn are defined by a broader socio-cultural context.

To give another example, originally computers were very big machines, which were operated by a group of specialists. They gradually developed into smaller objects, which can be operated by everyone intuitively. The demand for miniaturisation and intuitive software of a global, socio-cultural character was the environment in which the new models of computers developed.

The case of building is similar. A physical building – like other artefacts – can change only to a very limited extent. It is primarily a plan of a building, a building model, which undergoes transformations.

Nonetheless, compared to other artefacts, a building is a special case. It is not an object of mass production, where one car model is manufactured in thousands of copies; most buildings have its own respective models. The form and function of a specific building differs from that of another one because - putting aside factors such as global trends or collective needs of a class of clients – there are other, more specific factors affecting the building. One detached house can be very different from another one in the same neighbourhood for a number of reasons: a different shape of terrain, different pattern of adjacent buildings, different budgets of the owners, their different aesthetic tastes, needs, desires and the like. A building (contrasting to a car or a computer) is ‘attached’ to one physical place and – because of its large size – it has a great impact on the site, often defining the site.

In conclusion, the mechanism of building models’ development is in one regard similar to that of other artefacts: it relies on a deliberate action of human beings. A conscious human designer eventually decides (though affected by the ‘pressures’ of the environment) which direction the model of an artefact will take. On the contrary, the natural systems development happens without human intervention, and it involves a 'blind' mechanism of natural selection. The environment of change for artificial systems is a *mental space*, while the environment of change for natural system is a *physical space*. In both cases change is caused by pressures from the environments:

- In the case of artificial systems, the environment is defined by a socio-cultural space (in the case of buildings, the environment is defined more individually by a specific physical site and user intentions)
- In the case of natural systems – the environment is defined by a physical space (the mechanism of development is not caused by deliberate and conscious human action, but by a 'blind' process of eliminating less fitted organism).

BOUNDARIES OF A SYSTEM

A system is a set of interconnected elements, which is discernible from its surroundings. The strongly related elements that make a system can be referred to as an 'inside', while the environment of the system can be referred to as an 'outside'. The inside is separated from the outside – there is a boundary between the system and its environment. According to Cabrera (2006), a boundary is not itself an object, but it is rather a *distinction* between the object and what is not. "These two states are called identity and other. One could alternatively think of these states as thing and not-thing, here and there, this and that, us and them, in and out, internalities and externalities" (Cabrera, 2006: 12).

Weinberg, in a similar manner, points at a fact that the metaphors of 'inside' and 'outside', as well as 'part' or 'thing', 'object', etc., are closely allied to our experience of physical space, and thus, to our experience of 'boundaries'. It is our mental mechanism that distinguishes 'objects' from their 'outside', thus recognising 'boundaries'.

Considering physical artefacts, one realises that their boundaries can be defined intuitively as the surface of their outer elements. These objects exist in a physical space and their borders are physical. When it comes to natural systems, in many cases their boundaries are easy to define. For example, it is easy to distinguish a body of an animal from its surroundings. The distinction becomes more difficult when it comes to groups of animals (herds) or groups of plants (forests). Another example of a natural system where it is difficult to define physical boundary is a star, because it is made out of amorphous plasma (ionized gas) that is in constant physical transformation.

In these 'difficult' cases, a boundary – instead of being a conceptual separation of the inside from the outside – can be something like a belt, which incorporates both the elements of the system and some elements of the environment. As Weinberg remarks, such a belt could connect the system to the environment rather than separating them.

> The problem here is that a ‘boundary’ may not be infinitely thin, precisely so it can partake of both system and environment. Rather that separating, such a boundary connects. In order to make it perfectly clear that we are not talking about a perfectly thin, perfectly separating line or surface, system thinkers use the term ‘interface’ to describe that part of the world that, like the two-faced god Janus, looks both inside and outside at the same time. ‘Interface’ is a more useful word than ‘boundary’, for it reminds us to pay attention to the connection, and not just the separation, between system and environment. (Weinberg, 2001: 147)

Another view of boundaries is proposed by the so-called ‘Minimal Cell Model’, where a single cell of a living organism is modelled. The technique used in this case consists in a gradual exclusion of cell components. Each time a component is removed, one examines whether the cell can still perform its function. In other words, this technique keeps only these elements of the system that are necessary for the system to function (Cabrera, 2006).

Finally, there are systems that do not exist in the physical space: conceptual systems. A plan of construction of an artificial system is a mental construct. It is a system of interconnected representations, conceptualisations and images referred to as a physical space – a model of an artefact. In this case, the boundary can be understood in three ways:

- ‘Physically’ – in an intuitive way, like in the case of the physical objects, i.e., the outer surfaces of constitutive elements of the object are their boundaries. A model of a car would have boundaries defined by its physical realisation, though the former is a mental construct. The boundaries of a car model would dynamically follow the alterations of the model during the design process.
- ‘Functionally’ – as in the ‘Minimal Cell Model’, i.e., as embracing only the elements of the system that are necessary to its function. Unlike the cells, models of artefacts are somehow difficult to examine following principles of functionality. The reason for this is that the *necessary* functions of an artefact are not as easy to determine as a function of a cell. For example, is it sufficient that a

car transports people or should it provide comfort and safety for the passengers?

- ‘Inclusively’ – in a complex, comprehensive way, where the physical and functional boundaries would only be a point of departure for a much more comprehensive definition. The definition would include a net of references for each element of a designed artefact. In a mental model, the elements of an artefact do not exist independently as abstract entities, but they are defined through a net of references to a broader, ‘existential’ space (function in society, symbolic connotations). In other words, they are symbols, attaining a meaning and thus boundaries through an act of interpretation.

In a context of a generative design system, it seems reasonable to apply the first definition to a building model. According to this definition the boundary is defined by the surface of the outer components of the system. The boundaries of a building model would be then delineated by the outer walls of the building, optionally by the outer surface of balconies, terraces and the like[17].

The second, ‘functional’ definition seems suitable to the *generation* of a building model, where building characteristics play important role. The building characteristics reflect functional aspects of a building – they describe its ‘behaviour’. Thus, a definition of building characteristics would require algorithms and data that go beyond a simple definition of a building model. For example specification of energy consumption level of a building model would require algorithms that are based on laws of thermodynamics.

Finally, the third definition addresses a *user* of a generative design system. A user would be inputting design objectives to the system, embracing a design problem in its depth and extensity. A user would be a kind of a ‘link’ to the reality outside the system, exceeding largely its boundaries and making it an open system.

---

[17] A question of how ‘thin’ should be a physical boundary of a building model could be a subject of further consideration. Should elements like porches or entrances be considered as elements of a building model or as its environment? Or should they be considered as a ‘belt’ or an ‘interface’ between a building and environment? A boundary of a building model can go beyond surface of its external elements and it can embrace its close surrounding (surroundings within the boundaries of a property). However, usually the connections between a building and its direct surroundings are weaker than these between the constitutive elements of the building. Indeed, often in a design process, landscape architects design the surroundings relatively independently to the architects designing the building (though obviously in connection to the main elements of the building).

SECTION IV –ARCHITECTURAL DESIGN PROCESS

One postulate of this thesis is that the generative design system, should be well-adjusted to the actual design process. Following this suggestion, this section examines the design process from the perspective of constructing the generative design system. The examination is based on related literature review and on my experience as a professional architect. In this section I recognize general settings in which the generative design system should operate.

The section consists of two parts. The first one acknowledges the *complexity* of a design process. This complexity arises mainly from the fact that designers deal with the wicked problems. The phrase 'wicked problems' were originally used in the context of social planning and it used to describe problems that were very difficult to solve, because they addressed complex social interdependencies. In this section I argue, that the term fits architectural problems as well. The generative design system should be able to deal with this kind of problems. The wicked problems can be roughly characterized by:

- incomplete or general formulation of design objectives;
- the fact, that different parts of solution might be in contradiction to each other (for example cost of a building and quality of building materials);
- the fact, that apparent solution might reveal new problems, which leads to reformulation of the original design objectives (circular definition of a problem).

In the second part of the section I make an attempt to find a pattern in a design process, despite its complexity. I find two inherent features of design process, which I call *adaptation* and *iteration*. Assuming that design objectives are well-defined, one can understand a design process as a gradual transformation of a building model towards increasing fitness with the design objectives. I call this process adaptation. The term seems adequate, because a building model is being approached in a non-linear way and this process is more similar to development of an adapting system rather than to a deterministic, mathematical solution-finding process. Iteration on the other hand stems from a wicked nature of architectural problems. It consists in reappearing need for reformulation of the initial design objectives, based on conclusions derived from a developed solution. Developed solution often reveals new problems, that were not apparent at the initial phase of a design process (for example, that two design objectives rule out each other), which requires a reformulation of the initial design objectives.

I postulate, that the generative design system should take into consideration both adaptation and iteration. It should:

- be able to generate design by adaptation and
- it should allow a user to reiterate design objectives based on information derived from the generated design.

1. COMPLEXITY IN ARCHITECTURAL DESIGN

'WICKED' PROBLEMS IN PLANNING

Planning problems and mechanisms are similar in many aspects to the problems and mechanisms present in architecture. Rittel and Webber (1973) discussed a potential crisis in social policy caused by the nature of problems occurring in planning. The authors pointed out that first, as the society developed, and as the planers' aim was to provide basic facilities, such as schools, hospitals, highways and the like, the planning problems were easily definable. Planning was at that time a process of solution-finding and the planers' challenge used to be efficiency of solutions. The only requirement for the well-defined planning objectives was to assign technicians with skills for accomplishing them.

Later, society became more complex and pluralistic and its problems became more ambiguous. The planers did not only have to solve well-formulated problems, but they had to formulate the problems:

> By now we are all beginning to realize that one of the most intractable problems is that of defining problems (of knowing what distinguishes an observed condition from a desired condition) and of locating problems (finding where in the complex casual networks the trouble really lies.) (Rittel and Webber, 1973: 159)

The complexity of problems related to planning is well encapsulated in the term ‘policy resistance’. It denotes the fact that long-term effects of political actions are difficult to anticipate. Policy for reduction of street crime is a good example:

> What should we do to reduce street crime? Should we disarm the police, as they do it in England, since even criminals are less likely to shoot unarmed men? Or repeal the laws that define crime, such as those that make marijuana use a criminal act or those that make car theft a criminal act? (…) Try moral rearmament and substitute ethical self-control for police and court crime? (Rittel and Webber, 1973: 164)

Another feature of the wicked problems in planning is a transitory definition of planning objectives as they “rely upon elusive political judgment” (Rittel and Webber, 1973: 160). In a democratic society, public policy plays a crucial role and planning objectives are defined by political decisions. In this light, the wicked problems in planning are very different than the *tame* problems present in mathematics, natural sciences and partly in engineering. The latter are well-defined and one can clearly determine whether they are solved or not.

Summing up, a complex socio-political network in which a planner operates hardly ever provides straight answers. There are at least two features of the wicked nature of planning problems:

- It is difficult to formulate solutions, because of the complexity of a socio-cultural network planers operate in. This leads to the inability to predict long-term effects of decisions; the identification of the source of the problem is essential but highly complicated.
- A definition of planning objectives is provisional as it relies on the political decision, and it involves different opinions and interests.

INCREASING COMPLEXITY IN DESIGN PROFESSIONS

Problems and mechanisms that take place in planning are similar to those in architecture. Evidently, in both planning professions and design professions difficulty of problems increases. According to Simon (1957: 198), the capacity of human mind for formulating and solving problems is insufficient to the size and complexity of the design problems that involve human interactions in the real word. A dozen years later, Alexander (1970) notes that: “Today more and more design problems are reaching insoluble levels of complexity. (…) these problems have a background of needs and activities which is becoming too complex to grasp intuitively”. The building forms around us seem arbitrary and they lack clarity and organisation, because “their design has often taxed their designer’s cognitive capacity well beyond the limit” (Alexander, 1970: 5). Finally, Schön (1982: 14) speaks in similar manner: “In such fields as medicine, management, and engineering, for example, leading professionals speak of a new awareness of a complexity which resists the skills and techniques of traditional expertise.”

DIFFICULTY IN FINDING SOLUTIONS FOR WELL-DEFINED DESIGN OBJECTIVES

In architectural design, it is difficult to find a solution (a building model), which meets the design objectives, even if the objectives are clearly specified. This is because the architect operates in a complicated network of independencies and it is difficult to anticipate consequences of some design decisions.

Compared to natural science, the design activity is surely a different kind. The design process cannot be investigated directly with the analytical methods used in the natural sciences, and its outcome cannot be envisioned analytically. Rather, the designer’s method is to ‘probe’ the possibilities, to explore the ‘infinite expanding web’ of design choices. The designer responds to the changes in the network and simultaneously triggers new changes, in a dynamic and circular process of ‘reflection-in-action’ (Schön, 1982).

System dynamics offers a detailed account for the difficulty of anticipating the implications of design decisions. In terms of system dynamics, apparently unanticipated behaviour of a system arises from multifaceted interactions between positive (self-reinforcing) and negative (self-correcting) feedbacks, which in combination with time delay cause nonlinearity (Sterman, 2000: 12). Trivial examples of this include: the bigger the living room, the smaller the other rooms; the larger house, the more expensive; the more windows, the weaker the insulation; etc. The number of inter-relations is so large that it makes virtually impossible to assess all the consequences of all the possible design decisions[18].

---

[18] As stated by system dynamics researchers, the difficulty in anticipating effects of the design decisions is caused by the fact that designers do not use proper formal models for simulating the phenomena that design includes. According to Sterman (2000: 28), the designers “ignore feedbacks, multiple interconnections, nonlinearities, time delays, and other elements of dynamic complexity”.

CONFLICTS BETWEEN DESIGN OBJECTIVES

Even assuming, that the designer had methods to develop a building model that would meet the design objectives, still, the wicked nature of design problems could manifests itself right from the start of the design process, because different design objective might contradict each other or even rule out each other. Williamson et al. (2003: 127) formulate it as follows: “A building is always full of compromises, the result of juggling and trying to make compatible the diverse objectives of its creation.” The role of the designer is to prioritize the divergent design objectives and this prioritizing is a part of the design process.

The design objectives can be roughly divided into:

- Design requirements, which are relatively constant and defined independently from an individual design process (for example site constraints and building codes);
- Design intentions, which depend on the individual design process and which are defined in connection with a particular design context.

Master plan regulations are a special kind of design objective. They sometimes (especially in small projects) act as a design requirement, and then they are something given and constant. In other times (often in large projects) they act as a design intentions and it is a client and an architect that define them.

Normally, the master plan regulations are prepared by a local planning office and they reflect the spatial development policy of the local community, acting as legal regulations. In this respect, they are similar to the building codes – a design has to follow them. Specifically, master plan regulations include a set of more or less general guidelines for the size of a building, its aesthetic expression and principles for shaping surroundings of the building, such as the height of a building, the admissible building line, the number of floors, the function of the building, the footprint of a building, and the required green area.

Although usually a design should fit the framework approved by the master plan regulations, if there is a good reason, the regulations can be changed. The quality of the architectural proposal can be a decisive factor in renegotiating the master plan regulations. A good design can be used as guidelines to adjust the existing regulations.

Here is an example of a potential conflict. A developer may want to maximize the use of his plot by building a massive, high structure. In contrast, the municipal policy may favour scattered, low structures, separated with green belts with playgrounds and green squares. As this is the case, the developer and the locals have to negotiate and find a solution that is acceptable for all.

Another type of conflict arises when client's intentions can negatively affect architectural qualities. The architect's concerns might differ substantially from those of the client in respect to many building characteristics. The client's wish for a particular building type or design can have many different reasons. From the architect's point of view, the client's intention might sound unconvincing or even naïve. In other instances, the client might deliberately aim to minimize the building cost, thus compromising the architectural qualities. On the other hand, the architect's proposal might seem arbitrary or simply too costly to the client.

Furthermore, regardless of the discrepancies between the client and the architect, there are design intentions, which fundamentally are in opposition. Reduction of energy consumption for cooling for instance might be obtained at the expense of users' discomfort. The 'green' material might be applied at the expense of increased maintenance requirements (Williamson et al., 2003: 101). Especially the low energy consumption is used to compromise other design aspects: thicker insulation increases the building cost, smaller widows provide less daylight and energy efficient compact form of a building might worsen its visual qualities.

A conflict can also occur between the design requirements (as site constraints or building codes) and the design intentions (as client intentions or architectural qualities). This kind of conflict happens when – in order to satisfy the building codes - one compromises design solutions. For example, a certain number of floors in a housing project requires a certain number of elevators. The initially intended eight floors of a residential building might have to be reduced to seven floors because the eighth floor would require an extra elevator, exceeding the project's cost.

DIFFICULTY IN DEFINING DESIGN OBJECTIVES

The definition of design objectives is problematic not only because they can oppose each other, but for yet other reasons.

Brawne (1992: 34) says that it is very difficult to determine whether a building in its totality can be accepted or refuted. For example, even though architect's premises about how a building should work, how it should look like or how which functions should be allocated, appeared to be mistaken, it does not imply a total 'denial' of the building. There is no "guarantee that the design will be only what we intended it to be, for the reasons we intended (…) We will always be startled by the appearance of unintended consequences and other unpleasant surprises" (Nelson and Stolterman, 2003: 10). These statements do not refuse the reliability of an architect's expertise and solution, but they make it more relative.

Because the building model is only an analogue of a real object, the final product rarely becomes what was expected. The complete building is both less and more than had been anticipated in a design process. The expectations are rarely entirely fulfilled, because "however experienced and skilful we may be, we are not dealing with the real object but with some analogue, so reality has inevitably some surprises" (Brawne, 1992: 210-211).

Moreover, the quality of architectural solution is relative to time and to the transforming surroundings. Buildings can be adapted to new functions.

> Even a causal survey of the existing stock of buildings in any city would show that houses are being used as offices, schools or workshops, that warehouses become studios, that palaces may be art galleries, that chapels are used as club rooms, that an urban open space can be a car park one day a week and a market on another or, as in Singapore, an open-air restaurant in the evening. (Brawne, 1992: 40)

Hensel (2013) speaks in a similar manner, claiming, that designers do not have a certainty about adequateness of anticipated design objectives, since "future building activity might affect or perhaps even make redundant the performative capacity of the articulated envelope by changing key contextual aspects (…)".

Given that, it seems that more important than finding a solution to a well-formulated set of design objectives is to define correctly design objectives (design problem), considering a broad design context. The role of the architect is both to find means to achieve ends (finding solutions) and to define the ends to be achieved (formulate problems).

In the real-world practice, the problem is not 'given' to architects, but needs to be constructed from situations that can be puzzling or uncertain. Often the design problems are not defined at all: there are no design constraints or clearly formulated design intentions. For example, the building site may not be regulated, and the client may let the designer free to decide. There are practically no constraints except a building's type, and the design is supposed to be at the same time a solution for and a formulation of design objectives. This case is theoretical, but in practice it is not rare that the constraints are negligible. In such situations the architects have to 'construct' the design objectives.

When there are no constraints 'from above', the guidelines for formulating the design objectives should be possibly universal and generic. Usually, architects follow a current stylistic trend or their own artistic preferences. The possibilities are many and a building can:

- be cheap, efficient and functional, and play a social role (as the Bauhaus school postulated in the 1930s);
- be eclectic, ornamental, rich in recognizable meanings (postmodern architecture);
- be sophisticated and oriented on visual effect (deconstructive architecture);
- be balanced with respect to the environment (sustainable architecture), and so on.

The architect's preferences in selecting the style or his or her philosophical principles are only a first step towards concretization of the design objectives. It continues when architects decide on the design priority; be it an attractive shape, efficiency in site's utilization or relevant building materials. Furthermore, the design might be based on a rigid set of modules, and subordinate the form and the function to the logic of the modules, or it might be based on the 'organic' shape of capricious curves. Obviously, the 'design' of design objectives is a part of the process. The design process is not only an attempt to find a solution for an actual, 'existing' problem, but it is a creative act of 'imposition' of an order.

The design objectives can also be re-constructed. It often happens that a designer encounters situations, which exceed his or her ordinary means of conceptualisation. In such a situation, the designer might have to "construct a new way of setting the problem – a new frame, which, in what I shall call a 'frame experiment', he tries to impose in the situation" (Schön, 1982: 62). Again, Schön underlines the active role of the designer, who does not simply solve a given problem by applying previous knowledge and experience, but uses his inventiveness to re-conceptualise the problem or re-frame it. By assigning new meanings to the situation, he or she alters the comprehension of the situation, focusing on the elements that usually evade his or her attention. For example, instead of prioritizing the geometries of the building, the designer might focus on processes in and around the building, and adjust the geometries to the processes. In such situations design objectives are not something given at the outset of the design process. They are rather derivatives of a reconceptualization of the initial design problem.

Because the definition of design objectives is itself problematic and transient, one might conclude that there are no absolute criteria for judging a design. The quality of a design depends strongly on the broadly defined design context: “a building, a garden, a chair – cannot be rated as the best solution, either in an absolute sense or in the sense of a set of unique conditions, but only as the most preferred in those circumstances” (Brawne, 1992: 72). So, a solution for wicked problems can only be ‘acceptable’ or ‘sufficient’, relatively to the design circumstances.

SATISFYING SOLUTIONS

A design can often be improved if: more time and considerations were invested in the design process; the complex network of choices was examined more profoundly; and if the positive and the negative loops were studied more carefully (following the system dynamics’ terminology). Therefore, the design process strongly depends on the resources available in particular circumstances. Even a design that is far from being perfect might be accepted if the resources were used up.

> The planner terminates work on a wicked problem, not for reasons inherent in the ‘logic’ of the problem. He stops for considerations that are external to the problem: he runs out of time, or money, or patience. He finally says, ‘That’s good enough,’ or ‘This is the best I can do within the limitations of the project.’ Or ‘I like this solution’, etc. (Rittel and Webber, 1973: 162)

The design solutions are not true-of-false but rather relatively good-or-bad. There is not one perfect solution, which would be undeniably correct and accepted by all. Simon proposes the term ‘satisficing’ (combining the meaning of satisfy with suffice) to denote a solution that is good relatively to a particular context.

The solution in architecture emerges from a process of replacing poor solutions with better ones. This process is coordinated by an architect, but it involves many actors who revise their own judgment about the solution. The design is assessed by the architect, by the client or group of consultants, by the local community (including owners of the neighbouring properties) and eventually, by anyone who passes the constructed building. So, there is not one, best solution for a wicked problem, but the “assessments of proposed solutions are expressed as ‘good’ or ‘bad’ or, more likely, as ‘better or worse’ or ‘satisfying’ or ‘good enough’ “ (Rittel and Webber, 1973: 163).

SUMMARY

This section attempted to identify the causes of the complexity of architectural design. Firstly, finding a solution even for strictly defined design objectives is not a linear, straightforward process. The designer moves in a complex network of design choices, being only partly able to anticipate the consequences of his or her choices to the eventual design. This difficulty might lead to a situation where the final product is not the same as the one intended.

Secondly, the architect often has to formulate the design objectives (or in more general terms – to construct the design problem), for the following reasons:

1. The design objectives can oppose or rule out each other, and the designer has to prioritize them, formulating a new list of design objectives.
2. Because of a multiplicity of point of views, values and preconceptions, the specification of design objectives is subjective and can be seen differently by different stakeholders. Moreover, the relevance of anticipated design objectives might change over time. Therefore the design objectives are to some extent a matter of temporary construction.
3. Sometimes designer alone has to define design objectives both on a general level of architectural values and on a more practical level of a particular design solution. The architect is supposed to 'impose' an order where there is a lack of constraints.
4. The untypical design situations require novel approaches. The design problem needs to be re-framed and the design objectives re-formulated.
5. The fact that a number of possible choices and number of interrelated consequences of the choices are huge or practically infinite makes the anticipation of the implications of design choices very difficult. One design decision cuts off a number of alternative ones, and a designer is only partly capable to envision the eventual consequences of a path of choices he has taken. This difficulty in anticipation of the outcome of the initial problem formulation leads to a need for an eventual redefinition of the originally formulated design objectives.

These arguments suggest that it is difficult to speak of the quality of building models in absolute terms. Architectural designs are rather satisfactory (or 'satisficing', as Simon calls them) and cannot be judged out of their context.

2. DESIGN PROCESS AS ADAPTATION AND ITERATION

The previous section exposed complexity of architectural problems, which

- are difficult to solve because they exist in a complex network of possible design choices, and their consequences are only partly predictable;
- are difficult to define because there are different perspectives from which the relevance of the problems (design objectives) can be assessed, and their definition is always relative to a particular context.

This section is an attempt to find a pattern in the design process in spite of its complexity. Basing on both theory and my professional experience I recognize that there are two distinguishable features of a design process – its *adaptive* and its *iterative* character.

The adaptive character addresses a solution refinement. It takes into account the fact that a building model transforms gradually, towards the design objectives. The iterative character consists in a cyclical refining of the problem definition. It underlines the exploratory and circular nature of the design process, which repeatedly redefines the initial design objectives: "(…) design problems do not come fully pre-defined, but rather need to be explored by an iterative approach in which initial understandings of the problem and means of addressing it are refined" (Williamson et al., 2003: 101). Thus adaptation takes place within a loop of iteration. The following paragraphs discuss the two mechanisms in details.

ADAPTATION IN THE DESIGN PROCESS

In terms of systems science, a complex system evolves towards a better fitness with its environment. The process is called adaptation and it "involves progressive modification of some structure or structures" (Holland, 1975: 3). In a design context, a building model progressively changes the configuration of its building elements in order to fit the building environment. In the following quotation, Kolarevic and Malkavi (2005: 91) recognize the adaptive character of design: "(…) design is a goal-oriented decision-making model, where goals are defined by desired performance values". Here, a model adapts to the 'desired performance values', the latter being an equivalent of the building environment or design objectives.

The simplest definition of adaptation in the design context is a positivistic one, where designing consists of solving a well-defined problem: "Given the separation of means from ends, instrumental problem solving can be seen as a technical procedure to be measured by its effectiveness in achieving a pre-established objective" (Schön, 1982: 165). Holland points out that the adaptation of even a well-defined problem turns out to be difficult because of its complexity: "Basically, adaptive processes are optimisation processes, but it is difficult to subject them to unified study, because the structures being modified are complex and their performance is uncertain" (Holland, 1975: v). Certainly, the process is goal oriented, but the goal is approached not in a linear way but in a complex and uncertain one.

When it comes to the investigation method, in a positivistic sense, the adaptation of design should be analysed using 'analytical procedures', i.e., by applying problem-specific knowledge (for example calculus). Unfortunately, because of the intricacy of the task (many objectives have to be considered at the same time), there is no adequate analytical procedure and the only traditional way the problem can be approached is either a random search or an enumerative method.

The enumerative method consists in a systematic examination of all the different solutions for the problem, one by one. The random method is essentially similar, as it too examines solutions one by one, though the order of examination is not specified. The drawback of both methods is that "the order in which they test structures is unaffected by the outcome of previous tests" (Holland, 1975: 16). The efficiency of both methods in solving complex problems is similarly low. Even assuming that these methods could be applied to architectural design, they would be extremely unproductive.

Simply because it takes too much time to test all the possible forms, the complex adaptive systems use combination of stochastic and deterministic methods, adjusting dynamically the search focus. Living organisms, for example, test how their bodies react to the environment, and use the *results* of these tests to guide restructuring of the bodies.

Actually, in a design process the designer proceeds in a similar way. The designer probes the consequences of different variants of building models and adjusts his or her action to the obtained information. The designer's action is never enumerative; it is always partly stochastic partly based on intuition and experience. The designer does not follow any linear procedure, but incorporates a 'reflection in action'– he or she reads feedbacks and responds to it.

In contrast, the ‘romantic’ image of architects presents them as visionaries – architects would be inspired by a beautiful visual experience or some intuitive insight, and would then come up with an ingenious design. But “A man who sets out to achieve [design] adaptation in a single leap is not unlike the child who shakes his glass-topped puzzle fretfully, expecting at one shake to arrange the bits inside correctly” (Alexander, 1970: 59). Because of the high number of factors that have to be taken into account simultaneously, the designer’s chances of conceiving a good design in one conceptual leap are very small. Although it is true that architects often get inspired, a great deal of time and effort must be dedicated to a careful and laborious development of the initial idea. The alternative versions of the design have to be examined and tested. A good design never results from one, flawless, linear process.

The development of a building model is a gradual process, where solutions are being gradually ‘discovered’, after a laborious series of examination of different variants of buildings and building parts. Each design decision is a ‘what if’ speculation – it is a process, in which “designers come to understand the possibilities and scope of a problem through a circle of making proposals and reflecting on their implications.” (Williamson et al., 2003: 66).

The idea that a design should be generated in steps or in a process of constant reflections on the implications is nicely expressed in Alexander’s approach to design. He is sceptical to designing a building or a plan of a whole neighbourhood completely, and using the drawings as instructions from which to build. According to Alexander (2004: 176) “instead of using plans, designs (…) we must use generative processes.” Alexander makes a distinction between fabricated plans and generated plans. In the former group, a plan is designed ‘at-once’ in an office and the process is centralised – it does not involve the residents. In the latter group, the plan is not an effect of an ‘immediate’ decision of designers, but is flexible enough to let the residents gradually modify small fragments of the plan. In this concept, actions of the residents can be compared to the feedback information of adaptive systems. The generated plan ‘emerges’ as an effect of the decentralised process. The advantage of generated plans is that they are less error-prone than fabricated plans. That is because generated plans proceed step-by-step, at every step responding to the immediately apparent implications of its own development, whereas fabricated plans are designed ‘at-once’ as complete designs and they lack the explorative feature of the generated plans.

That said, if the consequences of each design decision are carefully considered, a generative approach to design is achievable in offices. For example, Alexander mentions sophisticated computer techniques that apply evolutionary optimisation to find an appropriate design form in a generative way.

> The very high-speed trains developed by the French, were given their shape gradually, not in the field, but in computer simulations. (…) This process [based on evolutionary computing], similar to organic adaptation, allows shape to be generated by a dynamic process in an evolutionary way. (Alexander, 2004: 152)

As a design process goes on, the design decisions must be expressed in spatial (architectonic) terms and their implications must be tested through simulations or 'thought experiments'. Based on these tests, the decisions are re-thought and revised. In this light, the design is a process of making small choices. It can be metaphorically illustrated as climbing a tree, starting from its trunk, selecting thicker branches, moving upwards, then choosing thinner branches, finally going towards branchlets and twigs. Each bifurcation is a choice-point.

Schön speaks in a similar manner. He uses the metaphor of moving in a complex net of possible design actions, where each move is to some extent determined by the preceding one. For example a decision to locate the main entrance for a shopping centre would 'automatically' constrain where the car parking can be. "In this sense, there is a literal logic of design, a pattern of 'if... then' propositions that relates the cumulative sequence of prior moves to the choices now confronting the designer" (Schön, 1982: 99).

On this 'tree' of design actions, each implication of a significant design decision requires a building model evaluation, and evaluations of different degrees and scopes take place at a number of design steps. The evaluations in turn involve mental or digital simulations of 'what-if' scenarios. So the careful and comprehensive analysis accompanies design process.

On the one hand, the implications of the design actions define the subsequent actions; on the other hand making actions shapes the situation. In this 'dialog', the designer achieves a better comprehension of how the design intentions can be optimally achieved (e.g., which actions will lead to the realisation of the design objectives or whether they can be achieved at all). In this process, a designer moves back and forth, learning the design situation.

> [A designer] must adopt a kind of double vision. He must act in accordance with the view he has adopted, but he must recognize that he can always break it open later, indeed, must break it open later in order to make

> new sense of his transaction with the situation. This becomes more difficult to do as the process continues. His choices become more committing. (Schön, 1982: 164)

Concluding, the method that designers use in the design process is in many ways equivalent to adaptation of complex systems. It combines the random search with deterministic methods. To large extent it involves testing implications of hypothetic design choices. Based on the feedback information, the designer adjusts his or her design decisions, aiming at building models that meet the design objectives.

ITERATION IN THE DESIGN PROCESS

The adaptation process might end up in a building model that for some reasons is not satisfactory. Specifically, the following questions can be posed: Which of the initial design intentions were achieved, and which were not (or only partly) achieved? Why were some of them achieved and others not? The unsatisfactory solution might require a reassessment of the design objectives as initially defined. Such a reconsideration of the problem I call iteration.

The mechanism of iteration can be illustrated by the following example. The initial client's intention is to build 100 apartments on a certain property, where the master plan regulations permits a building footprint of about 500m^2 and a maximum building height of 16 meters. Suppose that all attempts to develop such a building model fail, because a regulated utmost volume of the building would not contain 100 apartments. Even though the client's intentions seemed feasible at the beginning of the design process, after trying a few versions of a building model (adaptation), it has become apparent that the initial design intentions need to be compromised. The client may either accept reduction of apartments, or try to renegotiate the master plan regulations. This simplified situation might need only one iterative step, while more complex situations would require a series of iterations.

The initial definition of design objectives does not have to be very precise, achievable or even relevant. In most cases, they are revised many times in the subsequent iterative steps. The first approach to design and the first assumptions play the role of an initiation for exploring the design situation. The eventual formulation of design objectives is being achieved only in the design process and gradually. It is an iterative process of progressively better understanding of the design situation in its broad context. The situation reveals its potential step by step, and constrained by provisional design intentions it ‘informs’ how much it can ‘tolerate’. In each following iterative step, the design intentions are being adjusted and become more appropriate to the design situation. By repeatedly probing the design situation, the designer approximates the possibly best problem formulation and, eventually, possibly best solution.

In other words, the design process consists in continuous attempts to modify assumptions for a building model, and thus reshaping the whole design situation. It is a dynamic process of ‘mutual pressure’:

> The inquirer's relation to this situation is transactional. He shapes the situation, but in conversation with it, so that his own models and appreciations are also shaped by the situation. The phenomena that he seeks to understand are partly of his own making; he is in the situation that he seeks to understand. (Schön, 1982: 150)

Frequent redefinitions of the design intentions are therefore a means for exploring the situation. The design situation can also be described metaphorically as a ‘struggle’ between what is to be ‘created’ and what is ‘discovered’. The design intentions are thus a part of a design process.

One can assume that a building model is a system that adapts, and that design objectives are an environment with which the building model attempts to fit. The argumentation above emphasizes, that not only environment affects a system, but – to some extent – a system affects environment too. A situation of mutual dependency of a system and its environment is not only characteristic of design situations but it is known in social science. Here is an example provided by Sterman that shows how a change in the system triggers change in the environment, and vice versa:

> The system reacts to your solution: As your sales rise, competitors cut prices, and sales fall again. Yesterday’s solution becomes today’s problem. We are not puppets masters influencing a system out there – we are embedded in the system. The puppet master’s movements respond to the position of the marionette on the string. This is feedback: The results of our actions define the situation we face in the future. The new situation alters our assessment of the problem and the decisions we take tomorrow. (Sterman, 2000: 10)

A similar mechanism of ‘mutual pressure’ is also at work in biological environments. Biological interaction occurs all the time (competition, antagonism, ecological facilitation, symbiosis and the like). The evolution of life forms shows that organisms constitute the environment required for the appearance of new organisms.

The environment and the system affect each other in a similar way as do the design objectives and the building model. After each adaptation cycle, the design objectives might need to be reformulated, following new information about what is actually feasible and what is actually desirable.

PRINCIPLES FOR THE GENERATIVE DESIGN PROCESS

Concluding, the design process is based on two mechanisms: adaptation and iteration. The generative design system should include both. The adaptation phase is a goal-oriented mechanism in which the building model (the solution) is being refined to meet the design objectives (the environment). The adaptation process can be automated in the generative design system. The design objectives would be defined by a user and the algorithm based on adaptation should generate a building model.

On the other hand, the iterative process is less a solution-finding or optimisation process than a process of acquiring knowledge about the design situation. The generated building model itself is not as important as the lessons one can learn from its development. Therefore, the iterative process of cyclical redefinition of the design objectives (the building environment) depends on human assessment of the broader design context. Only the user can embrace the design situation in its totality and understand the meaning of potential discrepancies between a generated building model and the building environment, and therefore take an appropriate measure.

Figure 3 brings together the elements of the generative design process discussed so far. In particular, the adaptive and iterative character of the design process is incorporated into the scheme:

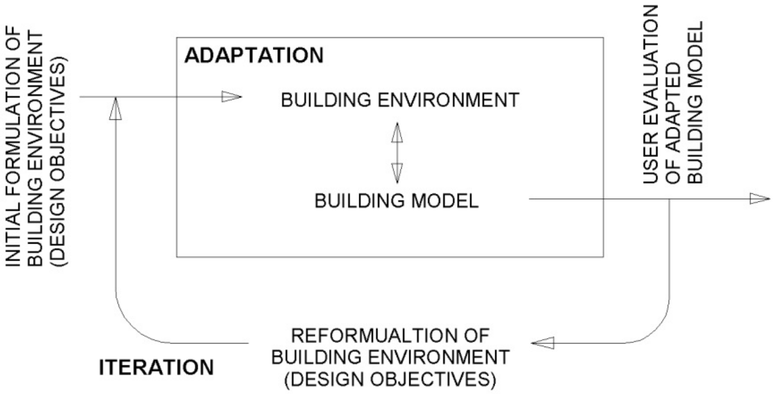


Figure 3. The generative design process.

SECTION V – THE BUILDING MODEL

In this section I focus on the first element of the generative design system, the object of a design process – a *building model*. This section consists of three parts.

In the first part, I discuss two meanings of the term building model. Firstly, it can be understood as a *mental* building model - an idea of the intended building. Secondly, it can be understood as a *digital* building model – a representation of the intended building, stored in computer's memory. Normally, a digital building model is entered into a CAAD system by a user and it is being modified in a design process, following the development of a mental building model. CAAD systems provide a set of modelling tools for modification of a digital building model. More advanced CAAD systems allow for its analysis and evaluations. In each case though, it is a user who manages the model's development.

It is different in the case of the generative design system. Here, a building model does not need to be *directly* controlled by a system's user. In the second part of this section, I discuss a digital building whose generation and remodelling can be to large extent automated. I start with the assumption that such a building model has to be conceptualized as a *system*, which *adapts* to a set of design objectives. In section III I introduce a number of definitions and properties of a system, for example that a system is an integrated and a goal-directed entity. In this section, I apply these definitions and properties to develop the notion of the adapting building model.

Finally, in the third part of this section I argue for a hierarchic structure of the adapting building model, saying, that some building elements should be interconnected stronger than others. This would improve the adaptation, because a modification of the building model would involve only reconfiguration of the groups of building elements not decomposition of the entire model.

1. A MENTAL AND A DIGITAL BUILDING MODEL

A MENTAL BUILDING MODEL

The major part of the architect's work takes place before the actual construction starts and it is a mental process of conceiving a building that is to be constructed. The architect works on an idea or a representation of a building – a *mental building model*. This model is a logical assemblage of building elements that are interrelated in a specific way and thus, the mental model might be thought of as a system.

Systems can be divided in two groups: natural systems and artificial systems, depending on whether they owe its existence to conscious human planning. Building models are artificial systems because they result from human planning and decision-making. But a mental building model is a plan of yet another artificial system: the physical building. Like any other model, the mental building model is "something that exists only in our mind" (Laszlo, 1996: 16). However, the building model differs from a 'system of logic' or a 'philosophical system', in the sense that it represents a 'yet-to-be-constructed' physical object. It is not a mere representation, but it is a creation – imposition of order.

On the other hand, the creation cannot be detached from its physical reference. Though it exists as a mental construct, the building model needs to be firmly linked with physical reality. Each operation on the model requires its contextualization and a reference to the physical object. Eventually, any building model, though composed of abstract symbols, has to be 'translated' into the language of physical objects. The spatial relationships of a building, its visual form, distribution of lights and other building's attributes need to be visualized "not as symbols but as *realities*" (Brawne, 1992: 13).

So, the building model has a dual character –it is a 'free' creation of human mind but at the same time it is a creation that is constrained by the context of the design (physical reality, site, functional objectives). It cannot be developed at the same level of 'freedom' as a piece of music or an abstract drawing[19]. This dualism means that the design process involves both creative acts (shaping the situation) and functional constraints (reading situation's 'feedback').

---

[19] This claim needs to be nuanced a bit. There are highly constrained forms of poetry, like a sonnet or a villanelle in which the poet has little 'freedom' in creating them. The constraints exist also in classical music (e.g. traditional number of movements, cadences, etc.) and in historical visual art (canons). The point here is, that many requirements in architecture are inevitable (like presence of entrances to rooms, windows, floors, bearing structure etc.), while in the poetry, music and visual arts, the constraints are a matter of convention not necessity.

Making models of natural phenomena is different from making building models. Science creates models of observed entities by discovering their common properties. Its method is empirical and often inductive. A model constructed on a basis of empirical observations is supposed to be valid for the whole class of the phenomena, e.g., a model of the life cycle of a star applies to all stars with similar characteristics. In contrast, a building model is a model of a unique entity. Thus, the development of each individual building model has to be accompanied by simulations and visualisations (tests of its performance in the real world). The method applied to building models is *hypothetical and inductive* (Popper, 1963).

A mental building model can be realised in many physical forms. Examples of physical building models comprise drawings (plans, sections, perspectives) or physical, 3D scale models. There are many reasons for making physical building models:

- By realising building models the architect displays, explains, communicates and discusses ideas with the actors involved in the design process. Especially, during the building phase, the set of plans, sections and technical specifications explain the architect's intentions in details.
- Human cognitive abilities are limited and cannot process all building's specificities or predict all the consequences (spatial, functional, social, economic, ecological) of design decisions. By giving a building model a physical form (by visualising it), the architect augments his or her cognitive capacity. As Schön (1982: 157) put it: "the graphic world of the sketchpad is the medium of reflection-in-action (…) the drawing reveals qualities and relations unimagined beforehand." Many properties of the building become apparent through its physical realisation: the allocation of shadows, relation of volumes, proportions and the like. This might guide the revision and further development of the model.
- Finally, by giving a building model a physical form, the architect records his ideas about the building. These ideas may be forgotten if they were not recorded or given a physical form.

A DIGITAL BUILDING MODEL

In this thesis, the term building model is used in a more specific sense, as a digital representation of a building, stored in computer memory – a *digital* building model. Depending on how advanced the applied CAAD system is, a building model can be a mere repository of unrelated geometric figures or it can be an *organized* entity, whose elements are interrelated.

The integrated digital building model meets all the three reasons for making physical building models. Firstly, the digital model could be displayed as a 3D object, it could be easily explored and examined, communicating the architect's ideas to the actors involved. Secondly, easy access to different properties and performances of the represented building could significantly inform the design process. Thirdly, each modification of the digital building model could be automatically recorded.

There have been numerous attempts at creating a possibly complete and coherent digital representation of a building. A concrete realisation of such a model was considered at least as early as in the 1990s. For example, Woodward wrote that digital building models "will be composed of not just representations of physical objects, but groups of data that know how to behave in particular circumstances" (Woodward, 1997: 28-29).

Building Information Model technology is an example of a realisation of a digital building model, in which geometric figures correspond to actual architectural objects. The predetermined 'semantic' content of the BIM components allows an unambiguous connection between the drawn element and its real counterpart. For example, a group of cuboids purposefully 'combined' into one object, can represent a window. In such a group, each cuboid can stand for a particular sub-object, one for a pane of glass, another for a window frame, etc. All the other building components, such as walls, roofs, floors, columns, doors, staircases, etc., can be created similarly. This allows a more direct and concrete connection between a geometric representation and its physical counterpart. The system informed by "what the components are" is able to automate drawing of these components, and to place them in a correct spatial relation to each other:

> (…) because the system 'knows' that windows go into walls, and that when there is a window there is no wall, it can automatically make an opening in a wall when a window is inserted. The constructional details around the window are also added to show the closure of the cavity and so on. (Day, 1997: 54)

There are other benefits of a coherent building model. Because the building elements are defined as 3-D bodies, the building model could be automatically controlled for its overall spatial coherence, or whether the building components overlap each other. Particular 2-D visualisation, such as plans or sections, can be derived from the building model automatically and thus, they do not contradict with each other.

Finally, such a digital building model not only can be easily investigated in terms of its coherence but it could be effortlessly evaluated from various perspectives. It can be combined with algorithms that for example support cost estimation, site management, structural analysis, facility management and others.

The major disadvantage of BIM systems when compared to the mental building models seems to be the lack of flexibility. As BIM is a digital realisation of a mental building model, in many ways it is less flexible than the latter. The mental building model is limited only by the imagination of a designer, while BIM technology is restricted to what is programmed[20].

Specifically, BIM offers a user a predefined (though extensive) database of entities representing architectural objects. Application of non-standard elements is often complicated. The non-standard elements have to be implemented to the system by a user and that usually follows reduced automation of these elements. Likewise, application of standard elements in non-standard ways is limited. For example, even though a connection between two types of walls is automated in a BIM system, the program still might have problems with an automatic and proper connection between two composite walls (a wall made out of prefabricated elements and a typically insulated wall with cladding). Which layers of the two walls should join? An architect must design the detail of the connection individually. Only typical connections between walls are automated. Day puts it as follows: “These systems [analogues to BIM] can be very effective in situations where the constructional technology is fairly standard but they cope less well where innovation is being practiced or where special conditions are encountered” (Day, 1997: 54).

The lack of flexibility in BIM reveals a more fundamental drawback of *every* digital system that is aimed at representation in architecture. The definition of building elements (the most basic entities of representation) and the definition of possible relations between them is to some extent already a *part* of a design process.

Summing up, the generative design system should harmonize two opposing options. On the one hand, it is desirable to *define* available set of building elements by attaching information to them and thus relating them to tangible architectural objects. Consequently, this option would restrict the spectrum of available elements. So, on the other hand, it is important to implement a *flexible* building model. Such a building model would need a broad spectrum of available building elements and their relations.

---

[20] On the other hand, complex calculations can generate sophisticated forms, which go beyond capacity of human imagination.

2. THE BUILDING MODEL AS A SYSTEM

In the following paragraphs I evoke the definitions and general properties of a system introduced in section III, in order to apply them to a building model. I postulate that a digital building model for the generative design system must itself be a *system*.

BUILDING MODEL AS A WHOLE

Laszlo speaks of systems as of the 'wholes' that are different than 'heaps'. The heap consists of elements that are weakly interrelated or not related at all. On the other hand, wholes are assemblies in which the relations between elements are strong and important. In Bertalanffy's terms characteristic of wholes are constitutive, while characteristics of heaps are summative.

The building model is a whole. The characteristics of the building model cannot be derived only from the properties of its individual elements. The amount of daylight in a living room, for example, cannot be determined from the characteristics of individual windows only. Rather, the spatial relations between the windows, roofs, floors, as well as other building elements need to be taken into account. Each part of the building is a constitutive element of it, and usually removing or changing one element affects the whole building. Contrary to a heap, where adding or removing elements has no effect on its essential properties, displacing a column in a building might make it collapse.

The renovation of a concrete silo that was originally used as a grain store can be an example. The objective is to transform the silo into a student hostel. This means, amongst others, installing windows to let the daylight in. But installing these windows is a delicate operation, because it can negatively affect the bearing capacity of concrete walls. So there is a strong relation between the bearing capacity of the wall and the size of the windows. The size of the windows needs to be adjusted accordingly.

The design process consists of a series of decisions that constrain each other until the completion of the building model, which is a whole. An architect, by a series of logically consequent design decisions, forms a coherent building model. It does not mean that the integration of the building elements is uniform and absolute. There are parts of a building where the relations between elements are stronger and where the relations are less numerous and weaker.

Furthermore, characteristics of the building model are constitutive. For example, the internal walls of a house, made of plasterboards and metal profiles, divide the house spaces, affecting the way of living in that house. Plasterboards fixed to a metal bearing profiles make up a wall of a living room, separating it from the kitchen, blocking the sound, the light and the smells. On the contrary, plasterboard in a building store has none of these functions – it might potentially be a division wall as well as a bookshelf. The assembly of plasterboards in the building store has summative characteristics – two plasterboards will double the weight. The assembly of the plasterboards in the building has constitutive characteristics – removing one plasterboard from the division wall would certainly destroy the function of the wall.

BUILDING MODEL AS AN ORGANISED COMPLEXITY

Weinberg's definition of a system is more selective than that of Laszlo and Bertalanffy. Weinberg takes into account the amount of the system's elements and the strength of their relations. Weinberg considers that a system is characterised by a neither too small nor too large number of elements, and that the relations between these elements are neither too strong nor too weak. Unlike mechanisms (which consist of a small number of strongly related elements), and unlike aggregates (which consist of a large number of weakly related elements), systems consist in relatively large number of elements in relations (organised complexity) (Weinberg, 2001: 19).

Following Weinberg's definition, one can speak of an organised complexity when a number and organization of constitutive elements of investigated object make it difficult to investigate with the analytical procedures. But conventional CAAD systems successfully apply the analytical procedures. For example building information models use the analytical methods to determine different properties of the model. The parametric design software too, uses the analytical methods for automatic distribution of building elements over a defined surface (a facade cladding over the facade). Considering this, one can conclude that the majority of CAAD systems – i.e., those that use only the analytical procedures – can be assigned to the first of Weinberg's regions, the mechanisms.

The generative design system should be able to go beyond that. In contrast to the 'static' building models that are applied in the conventional CAAD systems, the generative design system is supposed to manage a dynamic, adapting building model, being able to consider it as a whole. Thus, it requires a more subtle approach – application of the mechanisms from the region of organised complexities.

According to Bertalanffy, the application of analytical procedures in natural science requires that either a natural phenomenon has weak connections between its elements (and these relations can therefore be omitted) or that the relations between the elements are linear. In natural science, complex objects are partitioned and each part is examined separately.

In the case of the adapting building model, the connections between the elements are strong and their number is large. The simplest topological analysis of alternative areas and connections between spaces of different functional type is very difficult to perform. Even if the topological problems were simplified and if it were possible to formulate the solution in terms of linear equations, the computation needed for solving these problems would take a very long time. The following example illustrates the problem:

> Consider, for a simple example, a direct graph of N points. Between each pair an arrow may exist or may not exist (two possibilities). There are therefore $2^{N(N-1)}$ different ways to connect N points. If N is only 5, there are over a million ways to connect the points. With N=20, the number of ways exceeds the estimated number of atoms in the universe. (Bertalanffy, 1968: 25)

Again, the conclusion is that the generative design system cannot be based solely on analytical mechanisms, but it must apply the complex systems' mechanisms.

BUILDING MODEL AS A COMPLEX SYSTEM

Following Mitchel (2009: 12), there are three features of complex systems. The first one is complex collective behaviour, i.e., a high number of components give rise to complex patterns of behaviour. In the case of building models, patterns of behaviour can be likened to building characteristics. Here, usually the mechanism works the opposite way: the intended behaviour (a set of intended building characteristics) determines the configuration of components (position of the building elements). There is no analogy between a building model and a complex system in this respect.

The next feature is signalling and information processing, i.e., the system produces and uses signals from internal and external environments. In the case of a building model, elements are interdependent – a change in one element implies a change in another element. This is similar to the internal signalling of a complex system. External signalling, on the other hand, is analogical to the information exchange between a developing building model and a designer[21]. A designer makes decisions as how to modify the building model based on feedback information on effects of the decisions already made.

The third feature of a complex system is its ability for *adaptation* and perhaps this one is the most important for the generative design system. The adaptation consists of a change in the behaviour of a system that improves its chances of survival. An analogy to the adaptation would be the gradual transformation of a building model, in order to satisfy the design objectives in the best possible way.

3. A HIERARCHIC STRUCTURE OF THE BUILDING MODEL

In the subsequent paragraphs I argue for a hierarchic structure of a building model. I refer to selected, general approached to hierarchic structure of a system and apply them to the building model specifically. The point for structuring a building model in a hierarchic way is to make the generation process more efficient.

An important property of complex systems related to its hierarchic structure is *near-decomposability*, a notion proposed by Simon (1962). It posits that the number of relations is higher *within* subsystems than *between* subsystems. Near-decomposability is a property that facilitate a species' development, because the evolution process does not 'need' to decompose and reorganise entire structures of individuals, but it keeps the integrity of groups (cells, tissues or organs for example). Following Ashby (1960: 192), the hierarchic composition of systems significantly increased the efficiency of their adaptation: "No complex adaptive system will succeed in adapting in a reasonable amount of time unless the adaptation can proceed subsystem by subsystem, each subsystem relatively independent of the others."

---

[21] More generally speaking external signaling would be the information exchange between a developing building model and a building environment, where a designer functions as a 'mediator'.

Thus, forming groups of elements and operating on them rather than on basic building elements would make the generative process much easier. Such approach corresponds to the actual design practice, where a design process involves more often reconfigurations of groups of elements, such as staircases, rooms, certain fragments of walls and facades, than individual building elements. In this manner, instead of decomposing the model into its smallest elements and working on them individually every time there is a need for redesign, an architect can keep the successfully designed parts of a model only reorganizing them.

A reasonable arrangement of a building model might enhance efficiency of the generative design system. First step in such an arrangement would be recognition of distributions of connections between building elements. In the second step, strongly connected building elements should be grouped. For example, adjacent stair units together with handrails, balustrades and the like, should become a group. In this manner, similar types of elements (a certain type of wall unit) or cooperating types of units (e.g. frame of a window, glass units) that are close to each other should make groups. Examples of such groups might include:

1. a continuous wall type A, B, C… (a series of units of wall of the same type);
2. a window type A, B, C (units of windows of the same type);
3. a floor type A, B, C (floor units of the same type);
4. a ceiling type A, B, C (ceiling units of the same type);
5. a glazed facade, type A, B, C… (windows of the same or other type combined);
6. an entrance (doors and windows combined);
7. a bearing system component (columns and beams combined);
8. a ramp system (ramps and floors combined), and so on;

McShea extends the idea of near-decomposability by the notion of *nestedness*: a higher-level entity consists of 'lower' entities. McShea illustrates nestedness with the following biological example:

> Level 1: Prokaryotic cells (the simplest cells, such as bacteria);
> Level 2: Aggregates of level 1 organisms, such as eukaryotic cells (more complex cells whose evolutionary ancestors originated from the fusion of prokaryotic cells);
> Level 3: Aggregates of level 2 organisms, namely all multicellular organisms;

> Level 4: Aggregates of level 3 organisms, such as insect colonies and 'colonial organisms' such as the Portuguese man o'war (Mitchell, 2009: 110).

When it comes to the building model, its organization does not have to be 'two-dimensional' (building elements and groups of elements). It seems beneficial, that the building model is hierarchic on more than two levels, in a manner of nestedness. Small groups of elements can make up bigger groups.

Example of a structure that has more than two levels of hierarchy could be as follows. Level 1 of organisation would contain very basic components that represent elements manufactured and quality-controlled in a factory, like screws, beams, cladding plates, bricks, insulation fragments, window panes etc. Level 2 would contain assemblies of these elements: e.g. units of certain type of a wall, balustrades, doors, windows etc. Finally, level 3 would contain assemblies of elements of level 2, that refer to functional parts of a building, such as: staircases, entrance areas, communication areas, or structurally coherent parts of a building, like for example a glazed façade.

In the automatic adaptation process, only certain level of the building model would be decomposed. The depth of such decomposition should be a parameter of adaptation that is adjusted by the generative design system dynamically. Usually, large alterations of a building model occur at the outset of a design process and then the generative design system could operate on very basic building elements. Later phases of the adaptation could involve redistribution of only groups of elements. This approach can be applied very concretely in the context of Evolutionary Computing, using strategies such as a parameter adjustment or a parallel-terraced scan, where the extensiveness of building model's decomposition can be an adapting parameter (ref. section VIII.3.)

Finally, Christopher Alexander (1970) applies the concept of near-decomposability to architectural design. Novelty of Alexander's approach consists in the fact that he focuses on connections among the *properties* of a building model instead of connections among the building elements. He starts with the concept of misfit. A misfit occurs when a certain property of design does not fit the context of its use (for example a kettle that is too small, or hard to pick up when it is hot, or hard to store in the kitchen etc.) The role of the architect is to 'solve' these misfits, i.e., to adjust the properties of design to the design context.

Here comes the reference to the near-decomposability. Because the number of design properties is large, and the cognitive capacity of a designer is limited, a standard design practice is to 'decompose' the design problems into sub-problems. This decomposition is usually based on verbal categories, such as aesthetic, functional, economic, acoustic, structural etc. It is easier to work and solve design problems within each category separately, because this limits the number of interdependent qualities that the designer needs to take into account. Eventually, the designers try to put together the separate solutions and apply them to the design.

However, according to Alexander, such a conventional approach is problematic, because the distinction into verbal categories does not reflect the *actual* distribution of connections among the design aspects and the design properties from the different verbal categories might strongly depend on each other. For example, a functional quality (the shape of a home theatre room) is connected to the acoustic quality (quality of music) and thus, the solutions worked out separately for the functional aspects (overall layout and shape of rooms) could not match the acoustic qualities. Therefore, the conventional division of a design problem into sub-problems based on verbal categories is deficient.

Alexander proposes an alternative division, based on the actual dependencies amongst the design qualities. For example, the shape of a home theatre room would be in the same group as the acoustic quality of this room, because one property affects another. In this method it is essential to identify all the relevant design qualities and the links between them. Because the distribution of the links is not uniform (the design structure is not homogenous), it is possible to mark out groups of qualities which are more strongly interconnected (figure 4).

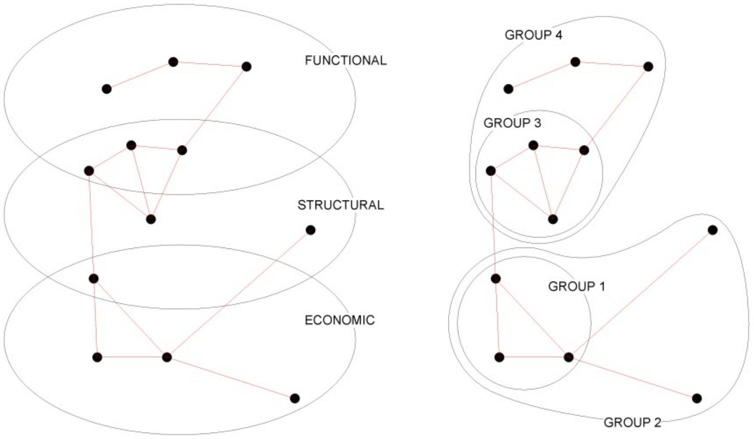


Figure 4. Two ways of dividing design properties. The first way (left) is the conventional one, where qualities are divided according to the verbal categories (functional, structural, economic etc.) The second way (right) is proposed by Alexander - here the qualities are grouped in terms of their actual dependencies.

Alexander's approach can be applied to the generative design system in a very interesting way, using EC techniques for multi-objective problems. Here I only shortly refer to the discussion in section VIII. Design involves multi-objective problems, because it considers many different and interconnected properties of a building model; and these properties have to meet respective design objectives. One approach to multi-objective problems is evolution of individuals in 'environmental niches'. Grouping qualities of design that are interconnected can be compared to grouping individuals in separate environmental niches.

PRINCIPLES FOR THE GENERATIVE DESIGN PROCESS

Three following principles, that address the building model, conclude this section:

1. A generic structure for building models should be flexible. The generative design system can represent only those building models that are anticipated in a predefined generic structure (it cannot represent all conceivable building models).

2. A building model should consist of elements which resemble real architectural components. That is, building elements should consist of geometric objects and auxiliary information about which architectural component is represented by the object. The information can include physical properties of a certain element, its function or how it relates to other objects.
3. A building model should have a hierarchic structure. Certain groups of building elements should be strongly interconnected making groups of elements. This would be beneficial in the adaptation process, because each redesigning of a building model would not entail a total decomposition of the model, but the groups of strongly interconnected elements would be kept.

SECTION VI – BUILDING CHARACTERISTICS

In this section I discuss *building characteristics* – properties of a building model. I propose that the generative design system should operate on building characteristics rather than building elements. Specifically, the adaptation of a building model would involve a systematic check of the similarity between characteristics of a generated building and design objectives. Theoretical aspects of this idea, especially the relations between behaviour (building characteristics) and a structure (configuration of building elements) of a system are discussed in section X.2.

This section consists of two parts. In the first one, I examine the 'emergent' character of building characteristics – the fact that usually they are not apparent, but need to be 'derived' from a configuration of building elements. Specification of building elements cannot be based on a mere configuration of building elements. One needs the additional information that, through a mathematical operation, could derive a particular building characteristic. I identify three ways of specifying the building characteristics: a visualisation, a digital simulation and an interpretation.

In the second part of this section, I discuss examples of building characteristics that can be implemented into the generative design system. I focus on the characteristic related to building aesthetics. I give an example of how a selected visual quality of a building can be determined mathematically and I suggest how this approach can be applied specifically to the generative design system (developing Salingaros' model). Finally, I suggest to distinguish the basic building characteristics from the complex ones. I postulate that the implementation of complex building characteristics – those that involve simulation and thus, need more computation – should be gradual. A generative design system should adjust the level of implementation considering complexity of a particular design task.

1. SPECIFICATION OF BUILDING CHARACTERISTICS

Architects tend to approach and consider buildings in terms of their qualities and characteristics rather than individual elements. Just like architects, laypeople focus on the overall effect of a building, for example its visual quality. Obviously, the building characteristics depend on the configuration of the building elements[22]. The characteristics (or qualities) of a system can be determined through the configuration of its elements (states). But as a quality of a system is something more than a state of the system, a building characteristic is more than a configuration of the building elements.

MICROSTATE VERSUS MACROSTATE AND THE CONCEPT OF EMERGENCE

The following paragraphs approach the *relation* between a state of a system and its characteristics. Its purpose is to outline a theoretical background for more specific discussion on how the building characteristics can be derived from a building model.

Dynamical system theory (or dynamics) is a discipline aimed at describing and predicting the behaviour of complex systems. On a microscopic level one can speak of a state of elements (configuration and interaction), and on a macroscopic level one can speak of the behaviour of a system. So, to some extent, the building characteristics correspond to behaviour of a building model (an energy performance could be an example).

This distinction also finds its application in statistical mechanics, which is a branch of physics dealing with large populations of particles – systems of particles. Since the position and velocity of each particle and the interactions of the systems of particles cannot be determined exactly, a system's description is statistical. In statistical mechanics, a microstate refers to the actual spatial configuration of particles and their velocity at a given time; a macrostate refers to the property of the system on a human-scale. A macrostate is an effect of a certain configuration of particles: "A type of

---

[22] Usually, to determine a building characteristic one needs selected building elements. And as "a quality is a way of grouping the states of a system" (Weinberg, 2001: 52), a building characteristic is a way of grouping the states of a building model. One building characteristic can be common to many different configurations of building elements. For example, a total floor area of 240m^2 is a common property of a number of different building models. Considering this, a building model can be represented as a set of building characteristics. A change in one of these characteristics would cause a change in many building elements. So, each building characteristic could be understood as an 'agent' that organizes a certain group of building elements.

microstate, for example (…) ‘molecules clump together –we can’t breathe’ versus ‘molecules uniformly spread out – we can breathe’, is called a macrostate of the system” (Mitchel, 2009: 50). It is evident that one macrostate (we can breathe) can be attributed to a number of different microstates (molecules can be uniformly spread out in many possible ways). Following the microstate/ macrostate distinction, the macrostate can be compared to building characteristic and the microstate can be likened to a configuration of building elements.[23] Furthermore one realises that each building characteristic (a macrostate) could be an effect of many different configurations of building elements (many microstates). For example, planned energy consumption per m^2 can be achieved in many different ways and using different configurations of building elements.

Often, on a microscopic level, there are a limited, small number of types of elements and a small number of relations between them. In physics, the variety of forms and behaviours (macrostates) in a ‘human-scale’ world arises from a few particle types and four types of forces between them. So the complex behaviour (or variety of forms) on a macroscopic level is an effect of interaction of a large number of simple forms subjected to simple rules on a microscopic level.

A similar principle works in mathematics. A logistic map is a simple equation, demonstrating how a complex behaviour (or trajectory) can emerge from a simple rule. The equation represents a recurrence relation $x_{t+1} = R \cdot x_t \cdot (1 - x_t)$. For R values greater than 3,57, the trajectory of the succeeding values of x_{t+1} looks surprisingly complex. Regardless the initial value of x, this trajectory is chaotic – the values of x_{t+1} produced by the equation are random and they form a complex, irregular graph on a diagram.

An important consequence of the fact that multiplicity of simple rules or relations can produce complex behaviours and patterns is a phenomenon called *emergence*. One cannot predict the behaviour of the logistic map unless one actually has carried out computations and visualised the results of the computation. The properties of the equation emerge only when the system performs (when the computations are executed). An example used in statistical mechanics illustrated a similar phenomenon, where a complex

---

[23] Salingaros (1995) proposes an extensive application of analogies from physics to architecture. He postulates three ‘laws’ for architecture, based on physical laws. Firstly, Salingaros finds an analogy on a small-scale. The contrasting pairs of elementary components (like electron-positron, neutron-proton, electron-nuclei) should be reflected in architectural detail in terms of contrasting shape, direction and colour. Secondly, he finds analogy on a large-scale. The interaction of particles in a micro scale effects in certain organization of matter on a macro scale, for example local structural symmetries emerge in a crystallization process. Salingaros postulates, that the same geometrical principles should be applied in architecture in order to achieve a harmonious form. Thirdly, the small-scale and the large-scale should be related to each other through intermediate scales, in a rule called ‘hierarchy of scales’.

behaviour of a system of particles on a macroscopic level was not an obvious derivative from the simple rules of interaction between particles.

In case of the generative design system, the architect is not that much interested in the precise position of building elements (a microstate), but rather in the effect of these positions – the building characteristics (macrostate). On the other hand, a digital building model in its basic form allows only the configuration of building elements (microstate). As a digital record, the building model is represented as a string of numbers, on which the building characteristics need to be based. Metaphorically, building characteristic should be derived from a string of digits in a way that is similar to how complex behaviour emerges from a configuration of particles.

The same situation takes place in the context of a usual design process. The elements of a composition have to be perceived from a perspective of the whole design, and only then the design qualities become apparent.

I assume that there are three methods for deriving the building characteristics from the configuration of building elements: visualization, simulation and interpretation. The following paragraphs will discuss these three methods.

BUILDING CHARACTERISTICS – VISUALISATION

In CAAD systems, a building model is recorded as a numerical description of its constitutive elements. This could be represented by three variables indicating positions of building elements in a Cartesian coordinate system plus a variable corresponding to the kind of each element. However, even if the description were detailed and complete, it would be impossible to draw any tangible characteristic directly from the ‘encoded’ building model. For example, in order to obtain a perspective view of a building, one would need a method to decode the numbers first and then to transform them into 3-D objects and then ‘project’ them onto the 2-D computer screen[24]. Examples of other transformations can include geometrical ‘projections’, such as plans, sections, facades, etc., as in a conventional design scheme.

The *visualisation* is essentially a change in representation. It does not require any other information than a set of geometric transformation rules. The visualization is a conventional way of representing a building model,

---

[24] In perspective views, change in representation is often combined with simulation. The digital perspectives include simulation of a colour of building material its reflectivity, softness or translucency and these qualities are rendered with high fidelity. Architects recognize a number of building characteristics that were difficult to identify from other representations.

because it is easy to 'read' building characteristics from the geometrical representation. The analogy to representation in mathematics is illustrative:

> That representation makes a difference is evident (…) All mathematics exhibits in its conclusions only what is already implicit in its premises (...) Hence all mathematical derivation can be viewed simply as change in representation, making evident what was previously true but obscure. (Simon, 1981: 153)

BUILDING CHARACTERISTICS – DIGITAL SIMULATION

Unlike the change in representation, simulation requires 'additional' information in the form of physical laws or social patterns – a specialized expertise that "draws its resources from many diverse disciplines, including physics, mathematics, material science and human behaviour" (Kolarevic and Malkavi, 2005: 86). But even though the fundamental physical laws and patterns of human behaviour are known and can be applied to building components, "the difficulty of the design problem often resides in predicting how an *assemblage* of such components will behave" (Simon, 1981: 19).

Thus, algorithmic approach is very applicable to simulations. In architectural practice, a simulation is often made without any formal modelling, especially when it concerns qualitative or social related characteristics (e.g., circulation of people in the building). To test their design hypotheses, architects make mental 'models'; they use analogies from the real world and rely on their experience, which in the best case scenario can roughly simulate the real behaviour. It is so, because "typical conceptual models (…) are too large and complex to simulate mentally" and "the complexity of our mental models vastly exceeds our capacity to understand their implications" (Sterman, 2000: 37). The intuitions about how the combinations of feedbacks would work are often misleading, resulting in incorrect conclusions, because "the feedback is very slow and often rendered ineffective by dynamic complexity, time delays, inadequate and ambiguous feedback, poor reasoning skills, defensive reactions, and the costs of experimentation" (Sterman, 2000: 37). Usually, human intuition fails to determine the behaviour of a model.

Finally, simulations (especially quantitative ones) usually involve extensive calculations. For unaided designer, such calculations are practically unfeasible. Even if one had sufficient knowledge about the laws and rules to apply (to calculate energy consumption for example), one would need a computer “to work out the implications of the interactions of vast number of variables starting from complicated initial conditions” (Simon, 1981: 19).

Considering this, the mental simulation of complex behaviour can be unreliable and a computer aided simulation offers a number of advantages. Not only it can deal with a complex interrelations of feedbacks and process more data, but it is also unbiased.

BUILDING CHARACTERISTICS – INTERPRETATION

When using computing in a generative way, it regularly happens that a building model can be interpreted in several ways. If the generated forms are not interpreted by a human, they may potentially represent anything: “the material resulting from the computerized emergent process has no meaning since it doesn’t necessarily represent anything.” (Sevaldson, 2005: 49)

Weinberg speaks in a similar manner, when he says that the properties of a system largely depend on human interpretation. He shows that the ‘emergence’ of the system’s properties is not something ‘intrinsic’ to the system, but that it is derived from the system by the observer’s creative act of ‘reading’ the configuration of the system elements:

> [System writers] speak in absolute terms, as if the ‘emergence’ were ‘stuff’ in the system, rather than a relationship between system and observer. Properties ‘emerge’ for a particular observer when he could not or did not predict their appearance. We can always find cases in which a property will be ‘emergent’ to one observer and ‘predictable’ to another. (Weinberg, 2001: 60)

Accordingly, interpretation is a key element in determining the building characteristics. Every representation of a building model (plans, sections, facades and perspectives) or every result of a simulation (a set of figures describing the building behaviour) has to be interpreted by a user. It is only then that the ‘raw’ building characteristics become meaningful. So, although a building characteristic to some extent is already present in the configuration of building elements, two steps are usually required to make it explicit: firstly a visualization or a simulation and secondly, the interpretation.

In addition, the ‘open’ and flexible character of building characteristics is evident. A significance of a particular building characteristic is contingent and depends on what is in a centre of the designer’s intention or interest.

Selection of relevant building characteristics in a design process is pragmatic and subordinated to the design purpose. For some clients the composition of the windows might not matter at all, as long as the function and energy efficiency are guaranteed. To others, a building cost might be of no importance, form being the most interesting aspect. Many clients would not be interested in structure details or technical details.

SUMMARY

The building characteristics are determined through simulation and visualisation combined with interpretation. Although these aspects are intertwined, one can formulate their typical features.

The visualisation requires configuration of building elements (a state of a system) and a set of transformations (decoding) rules. It differs from simulation in this, that it does not require application of natural laws or patterns from social science. Visualisation is a change in representation, which makes selected building characteristics visible and readable.

The simulation requires both: a configuration of building elements (a state of a system) and a set of rules based on principles of physics (then the simulation can determine such characteristics as energy performance, structural reliability of a building or distribution of daylight and shadows) or on patterns provided by social science (then the simulation can show for example circulation of people in a building or how the building functions in case of fire emergency).

The interpretation requires a designer (a user of a computer system), who eventually defines building characteristics, putting them in a broader design context. One representation of a building model can be interpreted in many ways, depending on a focus or an experience of a designer.

Application of these three methods would be beneficial to the generative design system. The visualization and the simulation can be automated, while interpretation suggests a need for user’s involvement. The user should contribute to the specification of some building characteristics. A concrete algorithmic technique that allows the user’s involvement is discussed in the context of Evolutionary Computing in section VIII.3.

2. EXAMPLES OF BUILDING CHARACTERISTICS

An example of a building characteristic can be ‘energy consumption per m^2’. To determine this, one needs to take into account the following figures: amount of energy produced by sunlight coming through glazed surfaces, losses of heat through external barriers of the building, the location and orientation of the building, the thermal performance of heavy parts of a building (thermal inertia) and other related phenomena. To make such a simulation “we do not have to know, or guess at all the internal structure of the system but only that part of it that is crucial to the abstraction.” (Simon, 1981: 20) In other words, we do not have to consider all the building elements, but we need to select the relevant ones. Then we process them with the set of equations derived from physics. The result would be a desired building characteristic – a parameter representing annual energy consumption per m^2.

Another example of a building characteristic could be a spatial adjacency, i.e., a list of connections between all the rooms in a building. A correct spatial adjacency is often a condition of proper building performance. Looking at the problem from the perspective of the generative design system, one can consider a scenario where the unconstrained system generates configurations of building elements without assigning functions to the rooms or even without defining the functional boundaries of the rooms. In this case, the role of the designer would be to assign a function to each room ‘manually’ – either in an arbitrary way or based on its size, relative position in the building, orientation, or identified number of connections (or centrality of the room). Unlike the previous example, where the energy consumption was determined with a set of equations (by simulation), here the building characteristics require an action of an observer – an attribution of a room type to otherwise unassigned room (the role of interpretation is apparent). It confirms Weinberg’s statement that at least some of the building characteristics are open and that they cannot be simply derived from the state of a system, but that they are an effect of the relation between the system and the observer.

The creative character of interpretation is even more evident when it comes to building characteristics related to aesthetics. It seems that to determine if a building looks ‘good’, harmonious or elegant, or rather unpleasant, awkward, clumsy, etc. is beyond the capacities of equations and it largely depends on the interpretation of an individual observer. The next paragraphs will look at this problem from another perspective.[25]

BUILDING CHARACTERISTICS RELATED TO AESTHETICS

Although the visual building characteristics are generally regarded as subjective, there are approaches attempting to measure the quality of a building's form in inter-subjective terms. For example, Salingaros attempts to formulate the aesthetic qualities of a building in mathematical terms. His key concept of building's 'life' is defined as a multiplication of the building's 'temperature' and its 'harmony':

$L = T \times H$,

The 'temperature' (T) is a measure of 'intensity' of the design and the degree of its 'visual stimulation' (in terms of differentiation of colour, details and curves) and it expresses the structure's informational richness. More precisely, T consists of five components: intensity of perceivable details, density of differentiations, curvature of lines and forms, intensity of colour hue and contrast (amongst other colour hues). On the other hand, 'harmony' (H) is a measure of visual organisation – symmetry and visual coherence of forms. Harmony consists of the following five components: reflectional symmetries on all scales, translational and rotational symmetries on all scales, degree to which distinct forms have similar shapes, degree to which forms are connected geometrically one to another, and degree to which the colours harmonize. Harmony is something opposite to randomness. While 'H' corresponds to a conventional meaning of architectural harmony, 'T' is a new concept. All the components of 'temperature' and 'harmony' are specified in detail in Salingaros' model, and the specification can be a basis for a further mathematical definition.

In broad terms, the degree of a building's 'life' expresses one's feelings about a building - the higher the 'life', the more comfortable and relaxed one feels. The opposite notion to the building's 'life' is the building's 'complexity', defined as follows:

$C = T \times (10-H)$

---

[25] The topic of intersubjectivity in art is large and addresses, amongst other issues, the role of art and specifically the role of architecture. For example, in phenomenology there is a key concept of 'lifeworld', the horizon of understanding of each individual which is based on his or her experience. The structure of lifeworld is both subjective and universal, allowing intersubjective communication. Thus – according to phenomenology – art should be not be only individual (like for example Kant postulated), but it should be understandable and participate in the socio-cultural reality (Sirowy, 2010). This thesis does not discuss this broad theme, assuming that the generative design system should be possibly a neutral (or 'transparent') tool, subordinated to the user's approach.

When one stands in front of a building with high ‘C’, one feels interest and excitement but also anxiety (Salingaros, 2006: 106). Combinations of different levels of ‘temperature’ and ‘harmony’ affect human emotions. A human being is frustrated either when structural information is missing (low ‘H’) or when it is overwhelming (high ‘H’):

> Since pattern recognition is a low-level brain activity, we may be intrigued intellectually by a low H form, but our visceral reaction is negative. A low H form can create interesting and powerful images that play with our emotions. (Salingaros, 2006: 113)

All the four variables of Salingaros’ model (‘life’, ‘randomness’, ‘temperature’ and ‘harmony’) can be applied to the generative design system as the building characteristics. If defined more carefully mathematically, they can be derived from the configuration of building elements and constitute a criterion for building model evaluation.

Salingaros’ model was inspired by a concept proposed by Alexander, presented in his comprehensive book *The Nature of Order* (2004). Alexander’s approach is very wide-ranging and detailed, but here I focus only on one dimension, which I think can contribute to a definition of some of the qualitative building characteristics. Alexander admits that in his study of a great number of architectural objects, he observed that some of them possessed certain, desirable structural features and therefore they had more ‘life’[26] (Alexander, 2004: 144). Eventually, Alexander distinguishes fifteen properties, which he considers essential for a form with a high degree of ‘life’: levels of scale, strong centres, boundaries, alternating repetition, positive space, good shape, local symmetries, deep interlock and ambiguity, contrast, gradients, roughness, echoes, the void, simplicity and inner calm, and not-separateness[27] (Alexander, 2004: 144-295).

For example, objects having the property of ‘level of scales’ “tend to have beautiful range of sizes, and these sizes exist at a series of well-marked levels, with definite jumps between them” (Alexander, 2004: 145). Alexander gives two contrasting examples of artworks – Josef Albers’ abstract painting which lacks noticeable levels, and thus depth, and the Matisse’s drawing of a woman with a range of scales. Alexander describes the level of scales of the

---

[26] Please note that Alexander’s definition of life differs from the one of Salingaros (which is defined as $L = T \times H$).

[27] Alexander’s fifteen properties find their analogies in three laws of architecture formulated by Salingaros in a context of physics (Salingaros, 1995). The two works are complementary.

drawing as follows: “There is the young woman’s body as a centre; the centres formed by a large open area on her back; the intermediate centres like her head, the hat, the brim; smaller centres like the flowers; and the very small centres like the petals in the flowers and the details of lace and buttons” (Alexander, 2004: 146). This ‘level of scales’ is elaborated mathematically by Salingaros. He specifies a scaling factor, setting its value to $e \approx 2{,}7183$, and shows a principle for its application on two-dimensional views of a design. According to Salingaros, one needs first “decide whether to measure areas, or linear dimensions” and then one should group the different substructures of roughly the same size into distinct sets. If one specified the size of the smallest detail as x_{min} and the scale of the whole object as x_{max}, then (presuming that a scale between two levels is ‘e’) the number of scales can be derived from the following equation: $n = 1 + \ln x_{max} - \ln x_{min}$, where n is the nearest integer value. For a three storey building, when the smallest detail is about 2,5 cm, the n value (the number of levels of scale) would be 7.

The property of level of scales is cited here because of its clear formulation and its broad and successful application in the architectural practice. It seems that it is not impossible to implement it algorithmically. The number of ‘levels of scales’ in a building model could be obtained by a change in representation, i.e., by a set of mathematical transformation applied to a building model. The transformations should first identify the groups of objects of similar sizes, an then put the sizes in an incremental order. Finally, it would examine the ratio between the subsequent sizes, checking whether and how many times it fits ‘e’.

The essential here is not a detailed description of all the properties and their practical implementation, but Alexander’s and Salingaros’ attempts to formulate universal properties or laws for a ‘good’ architectural form (forms to which human beings can feel connection). Both Alexander’s list of the fifteen properties and Salingaros’ model of ‘life’ can be a starting point for more detailed mathematical formulation of qualitative building characteristics (especially the visual qualities of form).

In addition to a mathematical implementation of the visual characteristics, again, it is necessary to involve a user to control a degree of their application and the overall effect of their combination. In this sense, the role of these characteristics would be informative rather than normative, i.e., the user would decide on application of an individual characteristic carefully considering a design context.[28]

---

[28] Strictly speaking, in the generative design system, the building characteristics would be functions. Outputs of the functions would be defined as results of respective simulations. An advantage of functional

COMPLEX BUILDING CHARACTERISTICS

I suggest distinguishing between basic and complex building characteristics. The criterion for the distinction is extensiveness of computation required for determining a given characteristic. The *basic characteristics* require simple computing and involve visualisation, while *complex characteristics* require extensive computing and they involve simulation. Examples of basic characteristics might include:

1. Areas: of different types of spaces, windows, facades, roofs, materials, etc.
2. Volumes: of different types of spaces and materials.
3. Numbers: of floors, spaces of different type, staircases, elevators, ramps, columns, beams, windows, doors.
4. Relations: of areas, volumes, materials, windows, but also combined relations (e.g. windows to floor area and the like).
5. Connections between types of spaces.
6. Form related quantities: number of external corners, different types of materials, different areas of flat surfaces, etc.

Examples of complex characteristics might include:

1. Energy performance of the building
2. Illumination of certain spaces or of the building.
3. Ecological related analysis: materials life cycle analysis, materials eco-profile etc.
4. Form analysis: complexity, regularity, symmetry or scale differentiation of the form etc. (terms developed by Salingaros and Alexander are relevant here: 'life', 'randomness', 'temperature', 'harmony').
5. Fire protection and safety system analysis.

A specification of some complex building characteristics should be implemented with an increasing level of accuracy, adjusted to the computational capacity of the machine. The precision of specification should be adjusted to overall complexity of a building model.

---

notation is that the parametric outputs can be easily combined linearly. In addition, such notation would permit to express and make use of partial knowledge about a system (a building model), i.e., it would permit to work on partly defined systems (building models) (Weinberg, 2001:106). The formula might be: $F_{building\_char} = f_{building\_char}(a, x)$, where 'a' is the variable known about a given building characteristic and 'x' is a variable unknown, but still affecting the building characteristic. A temporary definition of building characteristic (these with unknown variables) enables to work on their incomplete definitions, and consequently on incomplete definitions of building models.

SECTION VII – THE BUILDING ENVIRONMENT

In this section, I develop the central element of a generative design system – the *building environment*. The building environment defines a direction for a building model's development. It consists of a set of *building characteristics* that describe in a possible comprehensive way the intended building model.

The term building environment is closely related to the term *design objectives* (discussed in the context of a design process, Section IV). The difference is that the building environment is a specific element of the generative design system, proposed in this thesis. Due to the nature of computing it cannot include all types of qualitative design objectives.

The preliminary discussion about environment for natural and artificial systems in section III indicates that while natural systems adapt in physical space to fit physical environment, artificial systems adapt in 'mental space' to fit man-made objectives. Specifically in the generative design system, a digital building model should adapt in a 'digital space' (a computer memory) to fit the building environment (design objectives).

This section consists of three parts. In the first part, I formulate the principle, which says that a building environment should be *inclusive* and *balanced*. In the opposite case, i.e. if the building environment were fragmentary, the generative process would produce 'extreme' instances of building models, which for example would not have a feasible function, construction or form. The fragmentary definition of a building environment is a *common drawback* of all the generative design systems that I investigate in section IX. In order to make a building environment inclusive and balanced, I propose to base its structure on the guidelines of *sustainable architecture*.

Second important aspect of a building environment is that it not only should offer a list of building characteristics, but it should include their *predefined* parameters. So, the system's user would not have to define all the building characteristics that the building environment includes.

In the second part of this section I provide an outline for the building environment's structure. I construct a list of building characteristics that can be included in the building environment. I also categorize included building characteristics in order to secure, that the list is inclusive and balanced.

Firstly, I refer to a professional practice and classify building characteristics in terms of design *constraints* they address:

- design requirements – elements that are given and constant during a design process, such as: site constraints, building codes and master plan regulations;
- design intentions – elements that can be changed, such as: client intentions and architectural qualities;

Secondly, I refer to a 'triple bottom line' of sustainable architecture and classify building characteristics in terms of design *domains* they address: ecology, economy and society.

The third part of this section gives an overlook of how the *evaluation* of a building model should be realized in the generative design system. It should consist in the check of a match between the generated digital building model and the building environment defined by a user. The level of similarity between the *characteristics* of a building model and the *characteristics* included in the building environment would define the model's fitness. The evaluations would function as feedback information for the adapting building model. The adaptation process would involve a number of such evaluations.

1. INCLUSIVENESS OF THE BUILDING ENVIRONMENT

The main principle of the thesis is that the building environment should be inclusive and balanced. The generative design system should be able to consider a number of very different building characteristics. Such a capacity is typical for a designer, who more or less directly, takes into consideration a broad array of building characteristics, even if he or she does not address all of them explicitly. And even though a designer puts emphasis on selected building characteristics, he or she does not ignore the other aspects – they are still considered, sometimes not even intentionally. For example, even if the emphasis were on aesthetic qualities of a house, a designer could not entirely ignore its function, its energy consumption, its cost or its impact on the neighbourhood.

Such an inclusive approach is especially important for the generative design system. It is because, unlike a designer, a machine does not have any tacit knowledge about even most basic requirements for a building. Thus, if the building environment were defined in a very fragmentary way, the generative process could produce 'extreme' instances of building models – random objects that are not buildings[29]. The undefined design aspects could deviate very much from the admissible and reasonable expectations.

Moreover, considering the current applicability and accuracy of computer modelling and simulation, there is a reasonable tendency to approach design in the integrated way. For example Kolarevic and Malkavi (2005: 205) advocate the 'performative architecture', which: "spans multiple realms, from financial, spatial, social and cultural to purely technical (structural, thermal, acoustical, etc.)" They claim that very different aspects of building performance (building characteristics) should not be overbalanced by the formal expression (or the utilitarian approach) but they should be considered in parallel.

Concluding, the generic structure of the building environment should contain several building characteristics and they should address very different design aspects.

Furthermore, in architectural practice the clients' intentions usually need to be informed and constrained by architects. For example, the client might "not fully know what is concretely desired in the beginning [of a design process]", being only vaguely aware "that something is pressing for expression". In such situation "the designer must help bring to the surface a clearer articulation of a client's desiderata" (Nelson and Stolterman, 2003: 49). Thus, if the generative design system is supposed to automate the design process, it should be able to supplement the client's intentions automatically. To do this, the system should be equipped with a predefined set of 'normative' parameters for building characteristics included in the building environment.

What should be the source of these parameters? Certainly, the default building environment should feature a universal formulation of an 'optimal' set of building characteristics. A potential critique might point at the fact, that there is no universal measure for peoples' preferences and expectations and that they are subjected to the local cultural and economic circumstances. But, the purpose of the default building environment is not to replace peoples' preferences or impose alternative choices, but rather to supplement them in a moderate way. The intention is that the default building environment should 'secure' the development of feasible, complete building models. The

---

[29] Generation of extreme building models might be desirable in special cases.

generative design system would act similarly to an architect who, accepting the client's intentions, supplements them with features preventing the building from not functioning.

Now, the problem consists in:

- Specifying a set of building characteristics that is inclusive and balanced.
- Assigning default, normative parameters for these building characteristics.

I believe that the principles of *sustainable architecture* address both points. Though it is common to associate sustainable architecture with ecology, one can also emphasise its balanced and comprehensive character. Here, sustainable architecture is defined as a harmonized and inclusive architecture:

> (…) thinking about sustainable development is often represented in terms of three conceptual subsystems – environmental, economic and socio-cultural, the triple bottom line. Sometimes this representation is tied to an image that a sustainable development is like a three legged stool; fail in one subsystem, take one leg away, and the quest for a sustainable development falls over. (Williamson et al., 2003: 84)

Sustainable architecture addresses the triple bottom line of social, economic and environmental issues: social in the sense of community engagement and inclusiveness; economic in the sense of long-term growth and prosperity; environmental in the sense of local and global impact. Ideally, these three areas should counter-balance each other. Because of its completeness, sustainable architecture is a holistic approach. Within the framework of sustainable architecture, a building is supposed to be perceived as a whole, which divided into pieces, loses its logic. For example, focusing only on developing the environmental profile of a building (trying to fit it with the best possible environment-friendly devices) one can easily go out of the budget, devastating 'one leg of the stool' – building economy. Moreover, maintaining a complex system of 'ecological' equipment might be uncomfortable for the users. In order to bring back the balance, the extensiveness of the environmental part should be cut down (the number of environment-oriented devices should be reduced). As a result, social comfort would increase and maintaining cost would decrease resulting in a more balanced building.

Still, the term sustainable architecture might seem controversial because it applies to very different types of buildings:

> (…) from a woven grass and thatch bure on a Pacific island to a high-tech office in the United States. The former is reckoned to be a sustainable design because it is constructed entirely of biodegradable material and appropriates only a tiny amount of the world's resources for its construction, compared with a typical 'western' building. The office building may be considered an example of sustainable design if it requires significantly less energy for heating, cooling and lighting than is typical for its class. (Williamson et al., 2003: 6)

Both examples are expressions of the values that are associated with sustainability. And there is no contradiction here. These examples display universality of sustainable architecture and they are arguments for its application to the generative design system. A number of different architectural styles and building types can be regarded as sustainable.

Summing up, sustainable architecture aims at integrating environmental, economic and socio-cultural concerns in one building. It is opposed to those architectural approaches in which a *single* concept for a 'good building' is dominant. Sustainable architecture defined as an attempt to integrate and counterbalance diverse design concepts may provide a universal set of principles for architecture: "If we take out the adjective 'sustainable' in our checklist, we find a list that applies to all architecture" (Williamson et al., 2003: 136).

2. DEFINITION OF THE BUILDING ENVIRONMENT

In the following paragraphs I attempt to formulate the generic structure of the building environment. This structure should include a set of building characteristics considered by the generative design system. Below, I suggest how the selection of relevant building characteristics can be approached.

My starting point is, that the selection should be a top-down process. It should start with definition of general groups or areas of characteristics and proceed downward to specific characteristics. Such procedure should secure inclusiveness and balance of the building environment. I propose two general classifications for groups of building characteristics: according to the design constraints and according to the design domains.

CLASSIFICATION OF BUILDING CHARACTERISTICS IN TERMS OF DESIGN CONSTRAINTS THEY ADDRESS

> Imagine a client organization that asks its architect to prepare a highly detailed brief for a building, with all of the aims and requirements for the project set out, so that all will be clear when decision come to be made. When it is delivered, the client finds that only about a fifth refers to its own appreciation of its needs. Why all this other stuff? He asks. That's the brief for the other stakeholders' objectives, is the reply: the regulators, the neighbours, the community, us as architects, the other design professionals, and the planet. (Williamson et al., 2003: 65)

By the design *constraints* I understood groups of design aspects that affect (constrain) a building model. The basic design constraint is related to a building site. Other constraints refer to the stakeholders involved in a design process and they include: building codes, master plan regulations, client's intentions and architectural qualities.

Usually in architectural design practice, architects get a graphic description of the building site that includes the boundaries of a property, the shape of terrain, the layout of surrounding buildings, the plan of infrastructure and the like. From these data the architect derives certain properties of the site, like for example the distribution of shadows, the level of noise or the interesting eyesight. These properties in turn constrain position of the planned building and arrangement of its functional parts. The *site constraints* are important set of factors affecting development of a building model and they have to be included in the building environment defined within the generative design system.

Another constraint are *building codes*. Each building must be subordinated to building codes. In the case of housing architecture, the building codes can define for example the minimal size of bedrooms and bathrooms (e.g., according to the demands of universal design) or the number and layout of staircases (e.g., for security issues and fire prevention). The building codes can be implemented to the generative design system as a set of predefined building characteristics that have to be satisfied by a building model (ref. penalty functions in section VIII.3).

In addition to the *site constraints* and the *building codes*, the building model should be convergent with the spatial development plan of the local community. These plans are yet another element constraining a building model. For example, they can define the maximal height of the building, the type of a roof, the general aesthetic expression of the building, etc. The set of these regulations is here referred to as *master plan regulations*.

A bit different group of building characteristics that constrain design are *client intentions*. They are different, because they might not be constant during the design process. Usually the client has a more or less clear vision of his or her future house: number of floors, area, number of bedrooms and bathrooms, type of roof, etc. Sometimes, the client's intentions are not formulated precisely. They can be limited to a very vague sentence such as: "A house of about 200 square meters, for a family of four, with a big living room facing west." But whatever the initial client's intentions are, it hardly ever happens that they are not refined or even redefined entirely in the course of the design process. A realisation of a building model reveals unanticipated challenges and opportunities that might inspire redefinition of the initial design intentions.

Finally, the architect supplements, modifies and improves the client's intentions. Especially, if the client's intentions are deficient, including for example only a general shape and cost of the house, the architect should complete the intentions providing certain aesthetic and functional qualities. The role of the architect is to fit the house with all the basic functions and characteristics of a functioning house by applying his knowledge and experience. Moreover, the design should include 'higher' architectural qualities: aesthetic, ecological, symbolic and others, which make the house a valuable architectural object. I refer to the set of building features which result from the architect's work as *architectural qualities*.

In summary, the building environment should consist of building characteristics linked to the *site constraints*, the *building codes,* the *master plan regulations*, the *client's intentions*, and the *architectural qualities*. These constraints can be further divided in two groups. Firstly, the constraints, which are constant throughout the design process, such as the site constraints and the building codes. Secondly, the constraints which are being redefined and adjusted during the design process, such as the architectural qualities and the client intentions. The first group is here referred to as *design requirements*; the second group is here referred to as *design intentions* (see Figure 5)[30].

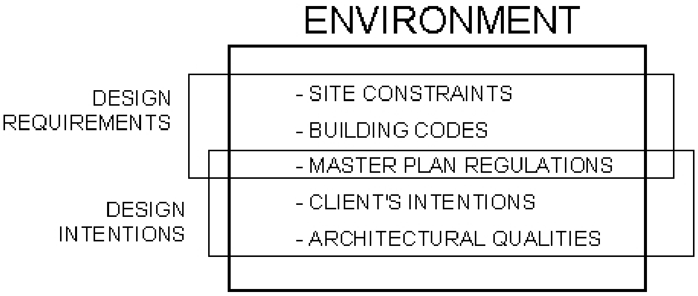


Figure 5. Constituents of the building environment.

---

[30] The building environment (understood as a set of factors affecting design) can be conceptualized in many ways. Nelson and Stolterman (2003), for example, give the alternative conceptualization. They divide the "triggers of design change" into three components: the real, the true and the ideal. "The real" is a *cause* of change and it comes from human intention; "the true" is associated with *artistic*, *religious* and *scientific thinking* (and it addresses chance and necessity) and "the ideal" refers to *norms* and *values*, coming from both the spiritual, higher order ("the Word of God") and legal regulations. Comparing this approach to the building environment proposed in this thesis, "the real" would correspond to the design intentions (both the client's intentions and the architectural qualities), "the true" would correspond to the site constrains and "the ideal" would correspond to the building codes and the master plan regulations. In addition, 'the real' would include the three methods for specification of the building characteristics discussed in section VI: the visualization, the digital simulation and the interpretation, because these methods 'verify' the design, revealing which of the design intentions are achievable in the real world. Moreover, "the ideal" would include the default building characteristics residing in the database and associated with the sustainable architecture.

CLASSIFICATION OF BUILDING CHARACTERISTICS IN TERMS OF DESIGN DOMAINS THEY ADDRESS

The second classification of building characteristics for the building environment of the generative design system is based on the triple bottom line and includes: ecology, economy and society.

A good example for grouping building characteristics around the triple bottom line is a tool developed by Chris Butters[31]. This 'value map' is meant to be a benchmark tool for sustainable architecture. It has a circular form, reflecting the inclusive and balanced approach to the building environment. Here, ecological aspects such as energy consumption, land use or material cycles are counter-balanced by social and economic aspects, such as socio-diversity, aesthetics, functionality, cost and others.

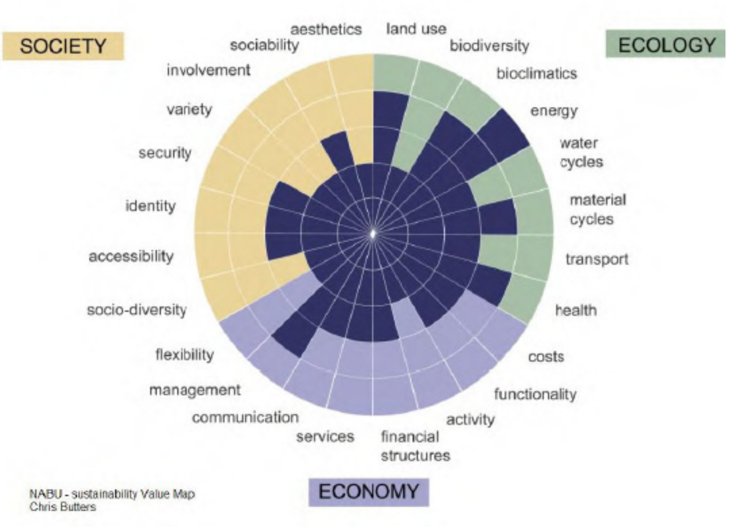


Figure 6. The sustainability 'value map' based on a triple bottom line (author: Butters, 2004).

---

[31] There are elaborated, point-based evaluation methods for sustainable architecture that are worth examining in this context. The most prominent methods are Leadership in Energy and Environmental Design (LEED) in USA and the Building Research Establishment Environmental Assessment Method (BRE EAM) in Europe. The purpose of these methods is (1) to examine and evaluate a design (or a constructed building); (2) to check its selected characteristics; and (3) to assign points if the characteristics correspond to the normative figures from the checklist. One reason for which these methods seem worth examining is that they contain very broad spectrum of building characteristics and that they address different design domains. Another reason is that evaluation criteria for the characteristics are well-specified. Because the evaluation system of these methods is quantifiable (based on points), therefore it could be applied easily to the generative design system.

The map does not emphasize any particular architectural style or does not prioritize any aspect (again, the circular shape implies their equality). It aims rather at a holistic description of a 'good' architecture.

In the following paragraphs I will try to examine selected building characteristics of the 'value map' in order to extract those that are relevant to the building environment.

Firstly, the generative design system assumes that the building site is given at the outset of the process and *constant*. The design system cannot change the site of a building, but it can only transform the building model. This means that such building characteristics as: accessibility to schools, to work, to commerce and to culture, cannot be included in the building environment.

Secondly, aspects that are unquantifiable or that cannot be specified using algorithmic methods should be excluded. They are for example identity, sense of place, belonging, history, culture and low crime.

Last group of aspects that the building environment cannot include are those that address building process, management and the like. They go beyond the scope of the generative design system. These aspects include:

- supplying employment,
- security of tenure,
- credit systems,
- process governance,
- participation, connection, shared responsibility,
- collective, energy-efficient, non-polluting transport.

In addition, there is a group of building characteristics that are partly quantitative and partly qualitative, and that to some extent can be addressed algorithmically. They include the following:

- life cycle analysis,
- functionality,
- communication transparency,
- adaptability to change over time,
- social diversity,
- accessibility for: children, elderly, disabled,
- experiential and sensory richness and variety,
- spatial hierarchy, private/public,
- ecological landscaping, green profile, land productivity,
- design localization, orientation, climatic adaptation,
- outdoor and indoor environment, noise, healthy ventilation,
- artistic, psychological and spiritual stimulation and pleasure.

THE CHARACTERISTICS DIAGRAM

The modified value map can be applied to the building environment to show a scope of its implementation in the generative design system. I will refer to such modified value map as to the *characteristics diagram*, to mark the difference. Specifically, the difference is that the characteristics diagram applies to the structures of building environments, and it indicates the *accuracy of simulation* of the included building characteristics. The value map, on the other hand, applies to buildings and it shows the extent to which a given characteristic is *realized* in a building.

To explain the difference I will use energy consumption as an example. The value map indicates how energy-efficient the evaluated building is: the larger grey area on the diagram, the more energy-efficient the building. The characteristic diagrams, on the other hand, shows how precisely the generative design system can specify energy consumption of the building models it generates. Some building characteristics can be absent on the characteristics diagram. It would mean that a given building environment does not include this specific characteristic.

The proposed characteristics diagram has two specific functions. The first one is 'constructive'– the diagram can be used as a 'guideline' for the inclusive and balanced implementation of the building environment. Secondly, the diagram can be a tool that measures the level of implementation of the building environment. I will use this diagram as an assessment tool in section IX, when I make the analysis of prototypes of generative design systems.

Furthermore, the characteristics diagram can be described by two features: the *resolution* and the *depth*. The resolution reflects a number of building characteristic that are included in the building environment. The depth reflects accuracy (or extensiveness) of the implementation of a particular characteristic (assumption is that 1 is a fully implemented characteristic). Graphically, the resolution corresponds to the number of grey 'rays' on the diagram, while the depth corresponds to their length (figure 7).

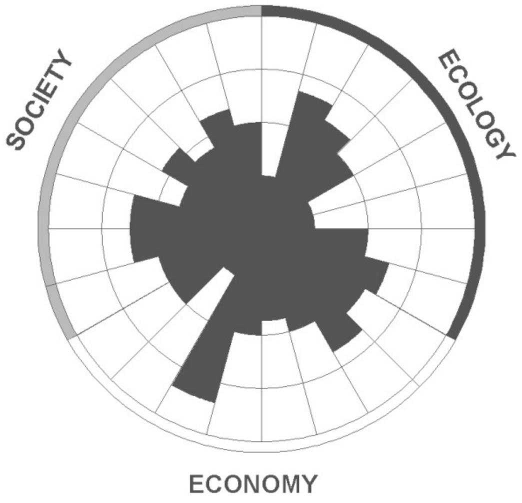


Figure 7. The characteristics diagram that shows example of a structure of the building environment. The resolution of this building environment is 24 and the average depth is 0,5.

THE GRADUAL IMPLEMENTATION OF THE BUILDING ENVIRONMENT

The complex building characteristics, such as for example: energy consumption, acoustic, thermal or air-flow performance and the like, require detailed simulation and thus extensive computing. Because the computational capacity of machines is limited, one has to implement only a few selected building characteristics. This is the option chosen by the authors of the prototypes of generative design systems that I analyse in section IX. Concluding from this analysis, this is not a good option. It seems that instead of implementing a few aspects of the building environment to a high degree (*low resolution* and *high depth* of a building environment), it is more beneficial to implement more aspects in less details (high *resolution* and low *depth*). Alternatively, some complex building characteristics could be excluded from the system in the first stage. The aim of such an approach is to secure the generation of feasible building models.

I suggest the incremental development of the building environment that grows ‘evenly’, following the growing computational capacity of machines. Figure 8 shows three examples of phases of a building environment implementation into the generative design system. The first diagram features 12 ‘rays’ stretching from its centre and having about 1/ 4 of its radius. Each ray represents a building characteristic that the system includes, and the area of each ray represents extensiveness of its implementation. Thus, the depth of the first building environment is about 0,25 (the average depth of all included building characteristics) and its resolution is 12. It is apparent, that even though there is a number of characteristics addressing different design domains, they are only ‘broadly’ implemented. The building environment is inclusive and balanced.

The characteristics included in the second diagram continue to ‘grow’ towards circle’s perimeter, increasing the average depth of the building environment to about 0,38.

Finally, the building environment illustrated on the third diagram is not only deeper (depth about 0,5), but also it resolution is higher (resolution is 24). Obviously, more comprehensive implementation of a building environment increases the quality of generated building models, but at the same time it requires more computational capacity.

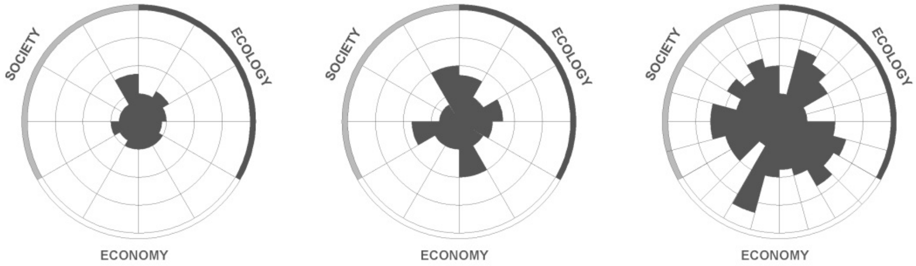


Figure 8. Desirable implementations of building environments. The ‘shallow’ but relatively ‘high-resolution’ building environments should be a starting point for a development of the system (diagram at the left hand side).

3. THE EVALUATION OF THE BUILDING MODEL

In a design process, a mental building model undergoes a series of transformations, in order to meet design objectives: “every design problem begins with an effort to achieve fitness between two entities: the form in question and its context” (Alexander, 1970: 15). Analogously, in a generative design system, the evaluation of a building model would consist in determining how much a given model fits with the building environment. Specifically, this could be achieved by comparing each building characteristic of the building model with that adequate building characteristic of the building environment (figure 9).

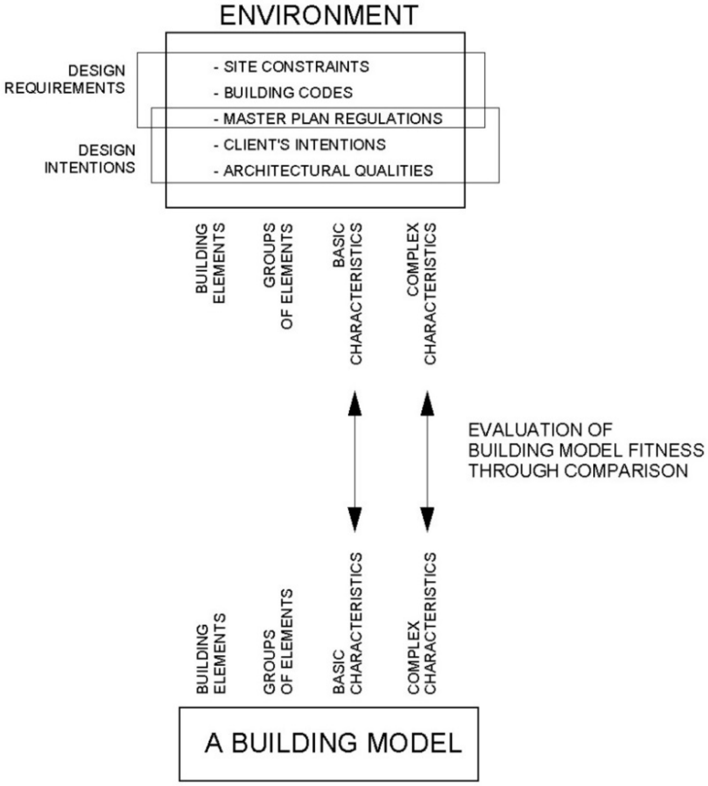
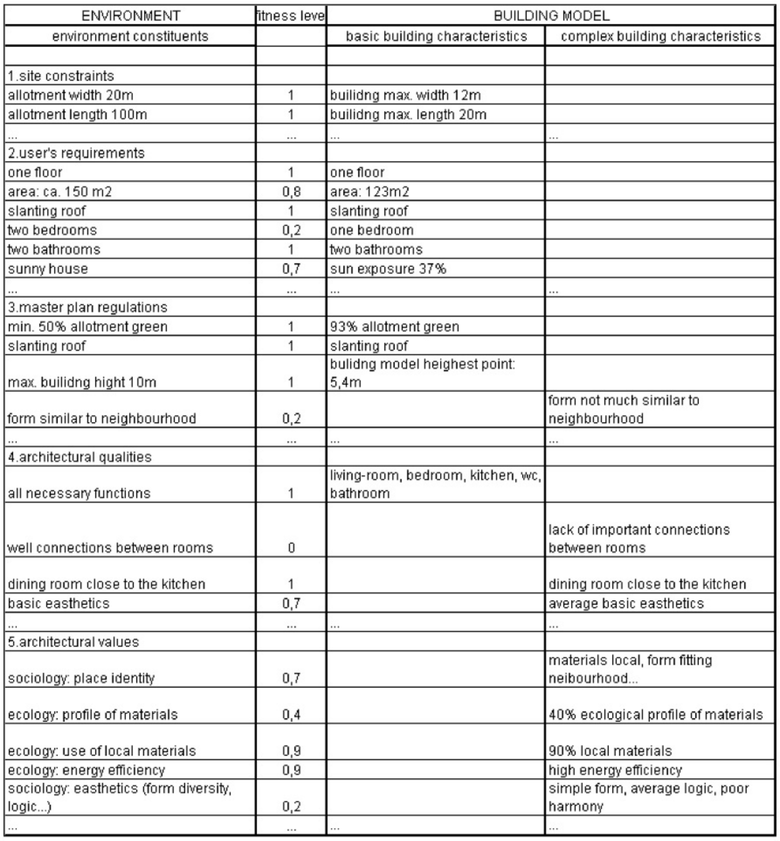


Figure 9. The evaluation of a building model’s fitness is a comparison of its building characteristics with that of the building environment.

The sum of all the similarities would define the fitness of a building model. If all the building characteristics fitted their counterparts, then the building model would fit the environment perfectly. Figure 10 provides examples of specific building characteristics and their parameters both in building model and in the building environment.



ENVIRONMENT	fitness leve	BUILDING MODEL	
environment constituents		basic building characteristics	complex building characteristics
1.site constraints			
allotment width 20m	1	builidng max. width 12m	
allotment length 100m	1	builidng max. length 20m	
...	...	...	...
2.user's requirements			
one floor	1	one floor	
area: ca. 150 m2	0,8	area: 123m2	
slanting roof	1	slanting roof	
two bedrooms	0,2	one bedroom	
two bathrooms	1	two bathrooms	
sunny house	0,7	sun exposure 37%	
...	...	...	...
3.master plan regulations			
min. 50% allotment green	1	93% allotment green	
slanting roof	1	slanting roof	
max. builidng hight 10m	1	bulidng model heighest point: 5,4m	
form similar to neighbourhood	0,2		form not much similar to neighbourhood
...	...	...	...
4.architectural qualities			
all necessary functions	1	living-room, bedroom, kitchen, wc, bathroom	
well connections between rooms	0		lack of important connections between rooms
dining room close to the kitchen	1		dining room close to the kitchen
basic easthetics	0,7		average basic easthetics
...	...	...	...
5.architectural values			
sociology: place identity	0,7		materials local, form fitting neibourhood...
ecology: profile of materials	0,4		40% ecological profile of materials
ecology: use of local materials	0,9		90% local materials
ecology: energy efficiency	0,9		high energy efficiency
sociology: easthetics (form diversity, logic...)	0,2		simple form, average logic, poor harmony
...	...	...	...

Figure 10. Example of an evaluation of the building model by comparison of its characteristics with the environment constituents.

PRINCIPLES FOR THE GENERATIVE DESIGN PROCESS

Perhaps the central principle for the generative design system relates to the building environment. I postulate that the building environment should be inclusive and balanced. The organization of building characteristics included in the building environment should be a top-bottom process. The design *constraints* and the design *domains* should be two main classification criteria. The design constraints include: site constraints, building codes, master plan regulations, client's intentions and architectural qualities. The design domains are drawn from the triple bottom line of sustainable architecture and they include: society, economy and ecology.

Secondly, assuming a limited computing capacity of a machine, it is more desirable that the building environment includes many building characteristics and simulates them less accurately instead of including one or a few building characteristics simulated very accurately. Thus, specification of some complex building characteristics (such as for example energy consumption, ventilation or acoustic performance) should be implemented with an increasing level of accuracy, adjusted to the growing computational capacity of machines. Alternatively, selected complex building characteristics could be omitted in the first implementations of the system and included later.

Lastly, the building environment should offer a default parameter for each building characteristic, so a user would not have to define a large number of building characteristics. The default set of parameters should be based on the framework of sustainable architecture.

SECTION VIII – EVOLUTIONARY COMPUTING

This section discusses principles of the generative design system which concern Evolutionary Computing (EC). EC is a research area within computer science, inspired by the mechanisms of natural evolution. I suggest that functioning of the generative design system should be based on evolutionary algorithms. Particularly, the adaptation of a digital building model can be simulated by these algorithms. In EC context, the adaptation of a building model would be realized as its *evolution* towards better fitness with the building environment.

The section consists of four parts. The first two parts include introduction to EC. First I explain essential terms and mechanism of EC and show how EC is applied to general problem solving. Then I show how EC can be applied to the generative design system. I establish an analogy between the general problem solving mechanism of EC and the adaptation of a building model that takes place in the generative design system. This analogy makes possible application of a number of well-established EC techniques to the building model generation.

In the third part, I make use of the established analogy. I develop the generative design system, applying such techniques as: mutation rate control, parameter adjustment, penalty functions, tournament selection, user's evaluation, multimodal and multi-objective problems and Pareto front.

Finally, in the fourth part, I support the claim, that EC is a relevant approach for the generative design system. EC approach is based on the 'explore, evaluate and refine' processes that reflect the non-linear nature of architectural design.

1. THE EVOLUTIONARY COMPUTING IN THE DESIGN CONTEXT

In the following paragraphs, I give an overview of what Evolutionary Computing is. I indicate the areas of its application and explain its basic mechanisms. Moreover, I show how this programming technique can be applied to the generative design system.

A number of biological mechanisms, such as growth, differentiation, wholeness, hierarchical order dominance, control and competition appear to be applicable in a variety of disciplines and in this sense may be said 'universal' (Bertalanffy, 1968). Identifying, defining and examining these concepts and their applications in different disciplines were the aim of systems theory. These mechanisms, generalized, gave rise to new methods for solving problems. Evolutionary Computing is one of them. It is a research area within computer science, which is inspired by the mechanisms of natural evolution. EC is successfully applied to many types of problems in fields as diverse as art, biology, chemistry, economics, engineering, genetics, operations research, physics, robotics and social sciences. When it comes to generative design systems, EC finds its application in research rather than in commercial software. Selected EC applications to generative design systems are discussed in section IX.

Although architecture generated by these systems can be called organic, it does not necessary *imitates* shapes of biological organisms. Rather, these systems apply the processes and mechanisms that are inspired by biology. Generally, there is a change in recent design thinking on organic design: "'nature' as a source of shapes to be copied" has been replaced by "'nature' as a series of interrelated dynamic processes that can be simulated and adapted for the design and production of architecture" (Hensel et al. 2010: 27).

In the natural evolution model, a fundamental concept is the environment filled with populations consisting of *individuals*. The time of individuals' lives is limited and thus they reproduce in order to preserve the population. The reproduction, if not stopped, would result in a population size that grows exponentially. But the *environment* can host only a limited number of individuals, because of the limited amount of resources. The individuals who are most likely to survive are those who utilize the resources most effectively. Each individual in a population varies from the others. The set of behavioural and physical features of an individual is called phenotypic traits. The phenotypic traits directly respond to the environment, determining the *fitness* of the individual. The chance of survival and reproduction of each individual is thus determined by its fitness. The better-fitted individuals proliferate, copying the desired features to their offspring. The offspring are not exact copies of their parents, but they are randomly modified, and it happens that some of the children are more fitted than their parents. These children in turn remain in the selection process, giving birth to next generation. In this manner, the population changes over time such as the average fitness of an individual increases (Eiben and Smith, 2007).

GENERAL PROBLEM SOLVING IN EC

Following the evolutionary mechanism, one can construct a universal algorithm for improving the quality of a solution for a number of problems. A starting point for such an approach would be the generation of a number of random solutions for a given problem. For example, if the problem were to find a best sequence of the next three moves in a chess game, the algorithm could generate a hundred random sequences of possible next three moves (anticipating the opponent's responses). These random sequences of three moves could be referred to as *candidate solutions*. The level of how well a particular candidate solution (a particular sequence) solves the problem would define the candidate's quality. The algorithm would then select a certain number of solutions (the most beneficial sequences of moves), reproduce them in a number of copies, and modify the copies. The modified copies of the best solutions would become a new collection of the candidate solutions, presumably with a higher quality on average. The cycle would be repeated many times, until a satisfactory solution is found.

The fitness of each individual is determined by the environment that the individual lives in – the same individual placed in a modified environment would have a different fitness value. Analogously, the same solution would have a different quality if the problem were formulated differently. The analogy described above is conventionally called the main Evolutionary Computing metaphor (Figure 11).

EVOLUTION	PROBLEM SOLVING
environment	problem
individual	candidate solution
fitness	quality of solution

Figure 11. The Evolutionary Computing metaphor (after Eiben and Smith, 2007).

EC APPLICATION TO THE GENERATIVE DESIGN SYSTEM

In the following paragraphs I will establish a link between the problem solving mechanism and the elements of the generative design system developed so far, such as: a building model, building characteristics and a building environment. Firstly, a *problem definition* corresponds to the definition of a *building environment*. The design requirements (site constraints, building codes, master plan regulations) and the design intentions (client's intentions and architectural qualities) form the design problem. A *candidate solution* for such defined design problem would be a *building model*. A set of initial solutions for the problem would be a set of randomly generated building models. The solution finding procedure would be analogous to the natural selection process: the building models that best fit the building environment would be reproduced and modified, while the less fitted would gradually disappear. As a result, a new 'generation' of building models would appear. This generation would undergo the same procedure, until a satisfactory building model had been found.

Furthermore, determining the fitness of an individual building model would consist in evaluating of how similar the building characteristics of the individual building model are to the building characteristics defined in the building environment. The fitness of a building model cannot be measured in absolute terms, but is relative to the building environment, just as the fitness of an individual organism depends on the environment the organism lives in. Because the building environment consists of selected design objectives, the fitness of a building model would be measured by the design objectives. Figure 12 summarises the basic evolutionary computing metaphor and applies it to the domain of architectural design.

EVOLUTION	PROBLEM SOLVING	ARCHITECTURAL DESIGN
environment	problem	building environment (design objectives)
individual	candidate solution	building model
fitness	quality of solution	fitness of building model

Figure 12. Extension of the main Evolutionary Computing metaphor to architectural design.

2. BASIC NOTIONS OF EC IN THE CONTEXT OF A GENERATIVE DESIGN SYSTEM

GENOTYPE AND PHENOTYPE

The individual in a population is called a *phenotype* while information necessary to build the individual is called a *genotype*. The genotype encodes the phenotype. In the process of natural selection, the phenotypic traits are considered (such as individual's strength, speed, hunting skills, ability to conceal etc.) defining the individual fitness. But it is the genotype that undergoes modifications during reproduction.

In terminology of a design process, the phenotype finds its analogue in the building model, and the phenotypic features in the building characteristics. The genotype has no direct analogue. In order to apply EC into design generation, one needs to construct a genotype, i.e., to find a way of encoding the building model.

GROWTH FUNCTION

The function that builds the phenotype using the genotype as an input parameter is called the *growth function*. Constructing the phenotypes and the genotypes in terms of mathematical objects is a significant challenge. Some guidelines of how the phenotype (the building model) should be constructed are given in section V.3.

In the generative design system, the growth function should allow generation of a possibly broad set of building models, or in EC terms, it should secure a possibly large search space. It should not block the development of some of the building models, because of the limitations in its definition.

To ensure the near-decomposability of a building model, it is desirable to create a growth function that would produce hierarchically structured phenotypes. The transition from a genotype to a phenotype should yield building elements that are grouped. Each modification of a genotype should result in a reconfiguration of groups of elements rather than building elements.

Finally, the growth function should be implemented in a way that discards spatially deficient building models, i.e., models that include spatial errors or inconsistencies such as overlapping spaces (overlapping windows, doors, rooms)[32]. It should also ensure that all the necessary building elements are

---

[32] Alternatively, the spatially inconsistent building models can be discarded by a technique called penalty function. The penalty function is discussed later.

present in a building model. For example, each room must be connected to at least one another room; and there should be at least one entrance to the building.

MODIFICATION AND SELECTION

In the evolutionary process, individuals strive for survival, and those best fitted reproduce and transfer their characteristics to their offspring. To ensure optimal development, these characteristics are not only copied, but also modified by a mutation or a recombination. The mutation is a change of randomly selected elements of the genotype. In the recombination, fragments of genotypes of two different individuals are combined to form a new genotype. It is important that the modification process happens at the genotypic level, while selection is based on phenotypes.

Accordingly, two mechanisms take place in the evolutionary processes: modification and selection. These mechanisms improve the fitness of individuals in successive populations. The modification creates the necessary diversity within the population (providing a novelty) while the selection increases the average quality of solutions (Eiben and Smith, 2007: 16).

ADAPTIVE LANDSCAPE – LOCAL OPTIMUM AND GLOBAL OPTIMUM

The relationship between the fitness of a building model and its characteristics can be represented graphically. In a simplified scenario, where there is only one design objective, for example a building footprint $100m^2$, the graph would be a 2-D curve. In figure 13, the vertical axis represents the building model fitness and the horizontal axis represents the different configurations of building elements, resulting in a different footprint area (a building characteristic).

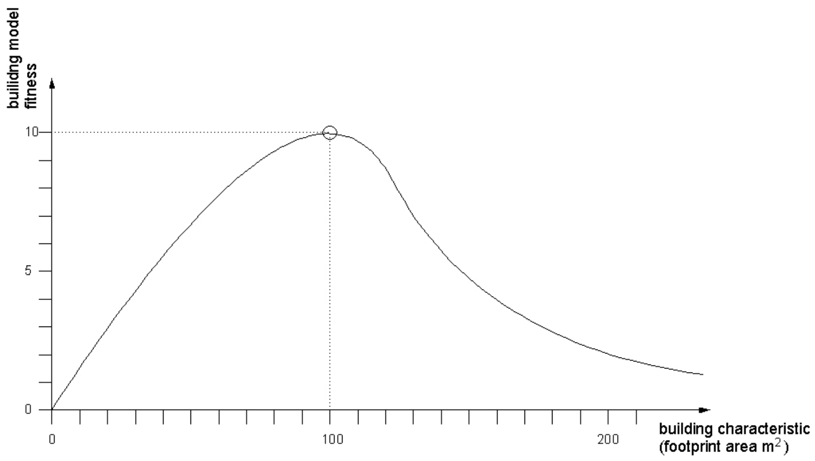


Figure 13. An adaptive landscape of a building model with one characteristic (footprint of a building).

When there is more than one building characteristic (which is usually the case), the line becomes a surface (figure 14) or generally a multidimensional space. The line, the surface or the space is called an *adaptive landscape*.

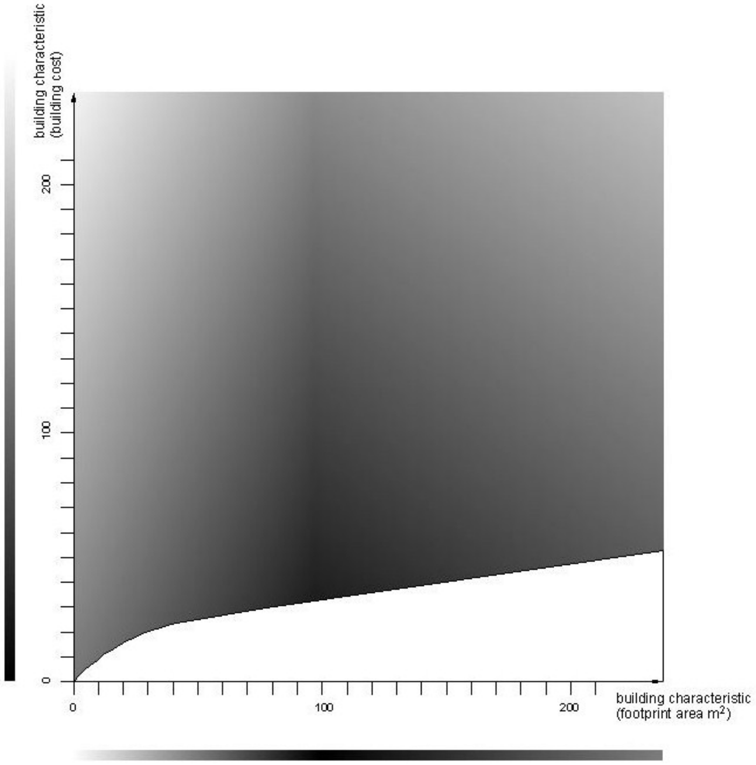


Figure 14. An adaptive landscape of a building model with two characteristics (footprint of a building and building cost). The design intention is a building footprint of about $100m^2$ and possibly low building cost. In this figure, the degree of fitness of a building model is indicated by more or less darker shades of grey. The cost of a building cannot drop beyond a certain level, which is indicated by the white shape at the bottom of the figure.

In complex combinations of building characteristics, finding the most fitted building model can be difficult. There could be many building models of comparably high fitness, positioned in very different places on the adaptive landscape. Each of these models would satisfy different set of design objectives. For example, one building model can suit the functional requirements – space adjacencies, room distribution and room area – but at the same time be energy inefficient. Another building model can meet aesthetic expectations, but it can be expensive. Both of them might have similar fitness, but their position on the landscape would be different. And, both would be represented in a form of ‘peaks’ on the adaptive landscapes.

So, the fact that a building model is placed ‘higher’ on the adaptive landscape than its neighbours does not mean that there is no other, better fitted building model. The best-fitted models in the fragment of adaptive landscape are referred to as *local optima*, while the best-fitted model in the whole adaptive landscape is referred to as a *global optimum*.

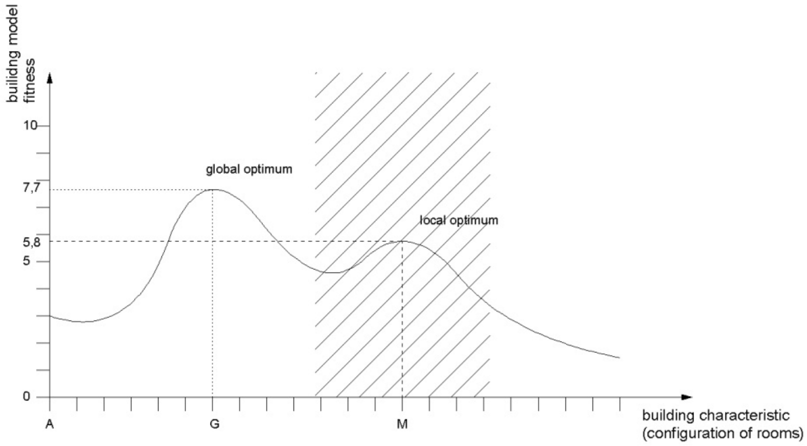


Figure 15. A local optimum and a global optimum.

Figure 15 illustrates the concept of a global and a local optimum in the context of generating a building model. The configuration of rooms is the only design objective. On the diagram, the horizontal axis represents building models that have different configurations of rooms. The vertical axis represents fitness of a building model with the design intention. The hatched area represents the scope of building models that have been generated and examined in the hypothetic generative design process. Even a relatively good configuration of rooms obtained in the process (local optimum M on the diagram) does not guarantee that there is no better solution (global optimum G on the diagram).

The building models that lie close to model M on the diagram are its small modifications. It is apparent, that in order to generate model G, the system needs to intensely modify model M.

Alternatively, in the context of an ordinary design process, one can assume, that model M is developed by a designer, based on his or her intuition and skill. The hatched area would represent a designer's scope of anticipation of alternative building models. The designer can anticipate only these alternatives that are relatively small modifications of building model M, and which thus lie in the neighbourhood of M. According to this diagram, the designer is unable to assess the quality of the models, which are large modifications of M (which lie outside the neighbourhood of M). In order to develop a better building model, the designer needs to introduce a large modification to model M, which would correspond to a considerable repositioning on the adaptive landscape.

GENETIC DRIFT

In connection with the adaptive landscapes that contain a number of local optima, it is worthwhile to mention a phenomenon called a genetic drift. This phenomenon is undesirable, because it might reduce diversity in a population of building models.

Often, two (or more) building models are *almost* equally fitted, but they owe their fitness to different set of building elements. Building models J and M on figure 16 lie on different positions on the adaptive landscape. However, because these building models are never equally well fitted (here M is slightly better), the probability that the offspring of the fitter will be more numerous is greater. This effect will be even stronger in the next cycle, and will gradually eliminate the offspring of the less fitted building model (model J). Consequently, this would deprive the population of its variety, neglecting potentially desirable building models. The effect of the genetic drift would 'push' the population development in undesired direction. The initial population containing the two building models would eventually 'stick' to the local optimum, as the offspring of the model J would disappear from the population. Even though the building model M is best fitted at the outset of generation process, the development of J is more beneficial, as it is closer to the global optimum.

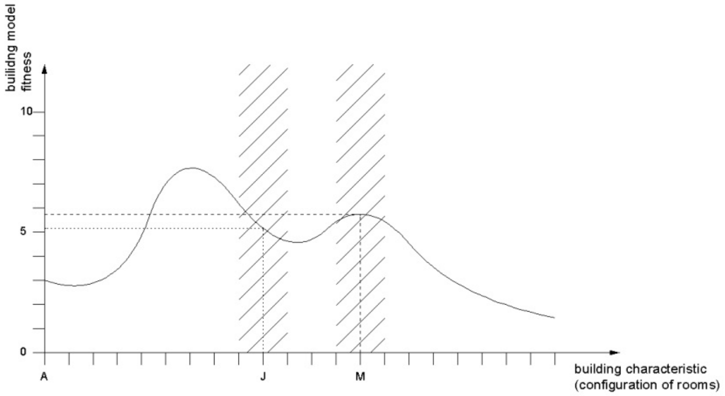


Figure 16. Genetic drift – a population might lose highly fitted individuals (J) that are close to the global optimum, following the proliferation of other highly fitted individuals (M).

In the context of the ordinary design practice, the design process, which starts with two or more competing versions of a building model, can end up too quickly in only one version of the building model. Focusing on one solution might rule out other, potentially even better solutions, too soon.

PREMATURE CONVERGENCE

Looking at the selection and the modification from another perspective, the selection can be described in terms of *exploitation*, as it narrows down and focuses the search, while modification might be described in terms of *exploration*, as it shifts the focus of the search. In this context, the evolutionary search can be seen as a trade-off between exploration and exploitation. Too much exploration leads to inefficient search, and too much exploitation narrows down the scope of the search too quickly (Eiben and Smith, 2007: 29). Extensive exploitation might cause a phenomenon termed a *premature convergence*. This happens when the population loses variety too rapidly and reaches only the nearest local optimum.

The analogue to premature convergence might take place in an ordinary design process as well. If during the early, conceptual design phase, no alternative building models were tested, then there would be a great chance that the project would not develop its full potential. In figure 17, instead of exploring many diverse alternatives, the search resources are focused on six relatively similar building models (K, L, M, N, O, P), which encompass a narrow area of the adaptive landscape. Here, the design exploitation dominates over the design exploration. Even though a fairly good solution is found very quickly (building model O) it is only a local optimum. Unfortunately, the potential of this design situation is only partially realised – the global optimum is not found.

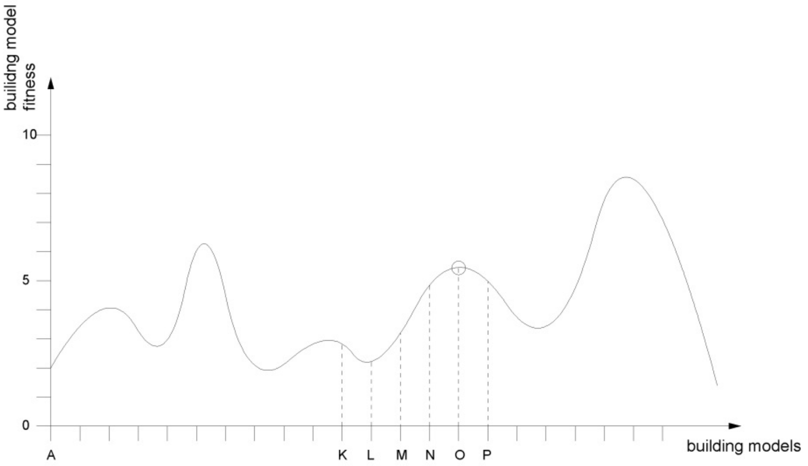


Figure 17. Exploitation. The search resources are focused on relatively small area of the adaptive landscape.

Figure 18 illustrates a different situation. As in the previous scenario, the resources are the same and they include six building models (C, G, J, M, P and S). However, now these models are allocated over a much broader area of the adaptive landscape. Three best-fitted building models – G, P and S – are selected for further development. The diversity of the initial set of building models (the fact that they are dispersed across a large area of the adaptive landscape) made it possible to discover the global optimum (which is in the neighbourhood of the model S).

Comparing this example of exploration to the previous example of exploitation, here the initially best-fitted model P is weaker than the initially best-fitted model O in exploitation. Nevertheless, after a few modification steps in the exploration scenario, the model S ‘climbs up’ the hill, quickly overrating the model O, which is ‘trapped’ in the local optimum.

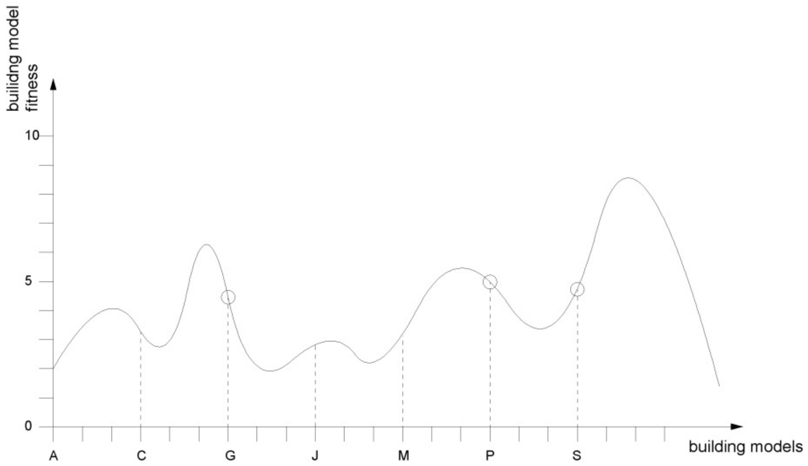


Figure 18. Exploration. The search resources are allocated over large area of the adaptive landscape.

3. APPLICATION OF SELECTED TECHNIQUES OF EC TO DESIGN GENERATION

The following section discusses selected mechanisms of EC that can be applied to the generative design system. These mechanisms are listed at the end of this section in the form of principles, on which the function of the design system can be based.

The first two mechanisms described here are mutation rate control and parameter adjustment. These mechanisms make the parameters of the evolutionary process a subject of the process itself. Based on the information about the search efficiency, the parameters are dynamically refined.

The second technique is called a parallel-terraced-scan. Assuming that the search is a trade-off between exploration and exploitation, this technique aims at appropriate distribution of the search resources between these two options.

Finally, there are four mechanisms that improve the management of the building characteristics. They offer strategies for working with building characteristics that are:

1. Undesirable or inappropriate, such as a lack of spatial coherence (the strategy is a penalty function).
2. Difficult to quantify in absolute terms (the strategy is a tournament selection).
3. Difficult to evaluate with algorithmic methods, because they are qualitative (the strategy is a user's intervention).
4. Too numerous, conflicting and belonging to different categories (the set of strategies dedicated to multi-modal and multi-objective problems).

MUTATION RATE CONTROL

The mutation rate control is a method for evading the premature convergence – the disposition of the evolutionary algorithm to focus and refine the first, good solution found. The premature convergence is an undesired effect. Because of the premature convergence, the search algorithm might overlook a number of better solutions only because they are placed at a certain distance (on the adaptive landscape) from the solutions that were initially found.

The premature convergence can be also explained in terms of exploitation and exploration. This phenomenon takes place, when the exploitation dominates the exploration. There are two decisive factors that determine whether exploitation or exploration prevails: the variation operators (consisting in mutation and recombination) and the selection mechanism.

In a search process, the relation between exploration and exploitation does not need to be equal throughout the whole process. In order to avoid the premature convergence, it is better to allocate the resources to exploration rather than exploitation at the beginning of the design process. Later in the process, it is desirable to focus on refining the best solutions that have been found so far. Thus, it is beneficial to be able to control the rate of mutation and make it larger at the beginning of the evolutionary process and decrease it over time. Again, a significant part of the search space has to be examined at the beginning of the search process, in order to identify promising solutions. Later, "as the search proceeds and optimal values are approached, only fine tuning of the given individuals is needed; thus smaller mutations are required.” (Eiben and Smith, 2007: 83)

The mechanism mentioned above fits the design experience. At the beginning, the architect operates on sketches of building models, which show only the main features of the intended building. This is because a sketch can be quickly modified, and thus it is an excellent tool for exploring of the search space. Making sketches and thus representing very different building models is similar to generating solutions with a large mutation step size (large transformations of a solution). Usually, a few most ‘promising’ versions of the building model are selected and developed further. Then, the mutation step size decreases, and the smaller and more detailed modifications are applied to the building models (exploitation).

PARAMETER ADJUSTMENT

Another important aspect of evolutionary search is the possibility to adjust and customise the parameters (such as the population size, the probability of mutation and crossover) to a particular problem.

There are at least two approaches to parameter adjustment. The first one is to run the algorithm with a set of intuitively defined parameters, and observe how the algorithm works. The parameters could then be adjusted on the basis of the algorithm’s performance. This solution has a disadvantage however: a large number of possible configurations of the parameters needs to be tested. Even if a large number of runs were performed with the different parameters, it would still be very difficult to conclude that the algorithm would not work better with yet another combination of the parameters. This is because the combination of parameters makes the reaction of the algorithm non-linear and practically unpredictable.

The second possibility is to incorporate the parameters into the evolutionary process, i.e. make the parameters themselves evolve. The change of the mutation step size, for example, does not have to be defined in advance. It can be adjusted throughout the evolutionary process, letting the algorithm itself determine the best mutation step size.

Furthermore, the parameter adjustment can be applied for example to the depth of decomposition of the hierarchic building model. At the beginning of the search process, the decomposition can be deeper, in order to amplify variations of generated building models. In time, the groups of elements should be more stable.

The analogy with the design process is apparent again. According to the discussion in section IV, a design has a self-reflective character. For example, the extent to which a designer modifies a building model is carefully adjusted to the actual phase of the design, in order to respond to constant re-evaluations of the developing model. When it comes to population size – i.e., the considered number of versions of building models – it depends on many factors, such as: a current design phase, the available time resources, the success of the current building model or the type of the building. Importantly, these figures are adjusted to changing design situations, in a manner similar to parameter adjustment in EC.

PARALLEL TERRACED SCAN

A technique called a *parallel-terraced scan* is a kind of integration of the mutation rate control and parameter adjustment. It optimally allocates the available searching resources in order to find the best balance between exploration and exploitation (Hofstadter, 2000). The focus of the search changes dynamically, based on information provided by the search. The parallel-terraced scan consists in a parallel exploration of many areas of the search space, with different concentration. The parallel character of the search consists in the fact, that it explores many different candidate solutions simultaneously. It is 'terraced' because not all solutions are explored with the same intensity and depth. The feedback information about the fitness of the examined candidate solutions informs the algorithm and adjusts focus of the search[33].

This mechanism can be applied to the search mechanism of the generative design system. The number of alternative building models should be large at the initial phase of the generation process. Successively, a few most successful alternatives should be selected and refined. Depending on the feedback information from testing the individual building models, the search resources should be either intensified on a development of the most promising building models (exploitation), or distributed over a broad area of the adaptive landscape (exploration). A random search for still new alternatives should be continued until the end of the whole process, though with only little resources allocated.

---

[33] The parallel-terraced scan is well illustrated by the way the human immune system works, producing white blood cells, lymphocytes. The way the lymphocytes are produced depends on the information about foreign bodies such as bacteria and viruses. The immune system focuses on developing the 'successful' types of lymphocytes, i.e., those that recognize and neutralise foreign bodies. Nonetheless, the production of random types of lymphocytes proceeds simultaneously, in case that an utterly different type of lymphocyte was even more effective in fighting the bacteria. The process has a tendency to change over time. At the beginning, the bacteria type is not recognized, and it is reasonable to allocate all resources on the production of random lymphocytes, even if they turned to be inefficient to fight the bacteria. In other words, at the beginning it is reasonable to produce as many lymphocyte types as possible (exploration). When the enemy is recognized, gradually more resources can be distributed for refining the successful lymphocytes (exploitation) (Mitchell, 2009: 182).

Figure 19 shows a hypothetical scenario. The building model N is the best-fitted at the outset of the generation process. Most of the search resources are allocated around his neighbourhood (hatched area). At the same time, the random search across the whole adaptive landscape continues (the dotted lines).

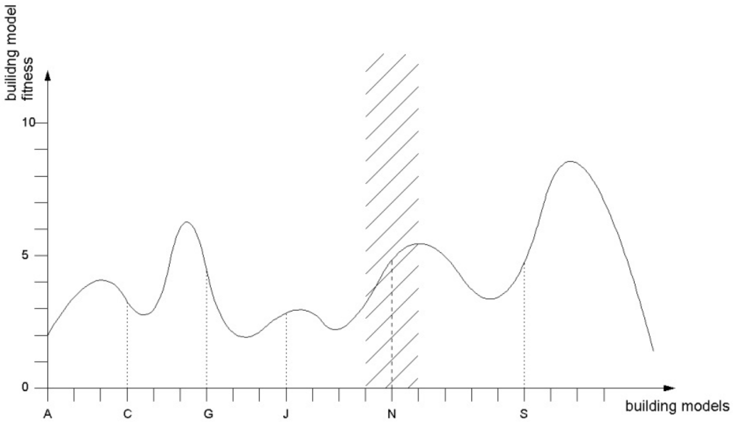


Figure 19. The parallel-terraced scan. It consists in a parallel exploration of many areas of the search space (here models C, G, J, N, S) with different concentration (here focus on area around model N).

PENALTY FUNCTIONS

The evolutionary algorithms might produce building models which are either: impermissible (for example they violate the building codes) or unfeasible (for example they are spatially incoherent), which practically eliminate the model from consideration.

A deliberate design of the growth function may be a way to avoid this situation. While it seems difficult in the case of building codes, it seems possible in the case of spatial coherence. It is desirable to have a growth function that develops a spatially coherent building model directly from the genotype.

A *penalty function* is easy to use as it requires only a straightforward modification of the fitness function. Through the penalty function, the user obtains extra control over the model development. The function can be used to substantially decrease the fitness value of deficient building models, and thus remove them from the population. The penalty function can be complementary to the growth function.

Finally, because the penalty function is another parameter of the evolutionary algorithm, it can be adjusted dynamically during the evolution process.

TOURNAMENT SELECTION

One of the crucial challenges in applying EC techniques to the generation of building models is the definition of fitness function (ref. appendix – an interdisciplinary discussion of this project). It seems especially difficult to assess the fitness of a building model in absolute terms (in numbers). To achieve this, one would need to assign the same units of measurement to very different types of building characteristics. It would be much easier to compare two building characteristics, and decide which one is better. For example, it is easier to compare two alternative room layouts and decide which one fits the design objectives better, than try to specify their absolute fitness values. Especially aesthetics is a domain in which assessments are relative: many can tell which of two buildings is more beautiful or harmonious, but few could justify their judgments in quantitative terms. Obviously, the evaluation of visual qualities will always be to some extent subjective.

The EC strategy that compares the qualities of two individuals is called the *tournament selection*. Usually it is applied to cases where no universal fitness can be defined, like game playing strategies. Here, the target of the evolutionary algorithm is to find best play strategy for a game. Although it might not be possible to quantify the strength of a given evolved strategy, it is possible to compare the two by simulating a game played by these strategies as opponents (Eiben and Smith, 2007: 63).

Additional advantage of the tournament selection is that it does not require knowledge of the whole population – it selects a representative sample. Normally, EC evaluates every individual in a population, while the tournament selection compares only the most characteristic individuals.

USER INTERVENTION

Typically, the evolutionary algorithms work without user intervention in the search process. The user's role is to specify the fitness function only at the outset of the process. However, there are cases in which the fitness function cannot be fully defined, because some features of the evolving individual are difficult to assess algorithmically. Therefore, the quality of the solution cannot be assessed fully automatically; a user's judgement is required. In the case of building models' evaluation, some of the building characteristics are qualitative and therefore a fitness function must be 'supported' by the user during the evolution. The involvement of the users in the evolutionary process is not unusual. For example, Herdy reports a case in which "a group of students is used to act as a subjective evaluation function for an Evolution Strategy" (Voigt et al., 1996).

A potential difficulty is that a user's judgment (even in the case of simply selecting between two competing solutions, such as in a tournament selection), significantly slows down the evolutionary process. The extra time needed for one evolutionary search would be proportional to the number of cycles multiplied by the average time of one judgment. It would therefore be beneficial to reduce the number of human interventions to, say, every tenth cycle or so.

MULTIMODAL PROBLEMS

Problems for which there is only one optimal solution (there are no local optima) are called *unimodal* problems. On the contrary, problems "in which there are a number of points that are better than all their neighbouring solutions, but do not have as good a fitness as the globally optimal solution" are called *multimodal* problems (Eiben and Smith, 2007: 154). Design problems are a good example of multimodal problems, because the design objectives (the problem) can be met by many different building models (there are many good solutions).

The generative design system can benefit from the approaches to the multimodal problems that EC offers. For example, the building environment (design objectives), initially defined by a user, usually change in the course of a generation process. Based on the information that a user derives from the generated building model, the originally inputted building environment might simply turn out to be wrong or unfeasible and there might be a need for its reformulation. In consequence, the generated building model might no longer be well-fitted to the newly formulated environment and there would be a need to repeat the generative process, which is usually very time-consuming.

In such situation, EC advices to focus on solutions from areas of broader peaks on the adaptive landscape rather than on solutions from areas of sharp peaks. The reason behind this is that "the latter may be over-fitted (that is overly specialized) to the current fitness function and may not be as good once the fitness function is refined" (Eiben and Smith, 2007: 155). In terms of the generative design system, the broad peaks represent building models whose characteristics are easier to modify, though these characteristics do not fit very well with the design objectives. The sharp peaks represent building models whose characteristics are difficult to modify, though they fit very well with the design objectives. In more explorative approach to design, where one assumes that the design objectives would change, it is reasonable to focus on the broad peaks.

Another interesting technique assumes that, instead of measuring how the individual fits with the whole environment, the algorithm can consider selected regions of the environment. Specifically in the generative design system, the building environment can be divided into niches; each containing selected building characteristics. Building models would then develop in parallel with these niches, and would be evaluated in terms of selected characteristics only. For instance, there could be four niches or 'sub-environments': a first one containing aesthetic of a building (evaluated by a user), a second one containing energy performance, a third one containing building function, and a fourth one being a combination of the three. The whole population of building models could be distributed over these niches in the following way: 10% in the 'aesthetic niche', 25% in the 'energy performance niche', 15% in the 'function niche', and the remaining 50% in the combination of these. The niches could also be defined as two or three correlated building characteristics, which would either reinforce or weaken each other. An example of such characteristics may be floor area and a functional plan solution. In this case, the small area reduces building cost, but it might be insufficient for the functional requirements. Another example can be energy efficiency versus a complicated building form. A very simple building form might be not interesting but it would be energy efficient. Finally, building models from different niches can mix occasionally, in order to increase a diversity of the population.

The 'niches strategy' can be developed, by simulating 'species'. An algorithmic equivalent of a biologic species would contain building models, which are similar in terms of form. The algorithm would maintain the diversity of species by ensuring that they evolve in parallel, 'communicating' with each other only after a fixed number of generations (EC term for this number is 'epoch'). The 'communication' would consist in exchanging a number of building models from a population of two selected neighbouring species (analogue to the phenomenon of migration). In this manner diversity in a population would be maintained, providing highly fitted building models from the different niches.

Maintaining diversity of building models by applying the niches strategy or simulating 'species' would improve the evolutionary process, making the generative design system more efficient.

MULTIOBJECTIVE PROBLEMS

The *multi-objective problems* are a sub-area of the *multimodal problems*. The technique of multi-objective optimization (or multi-objective programming) is the process of simultaneous optimisation of two or more conflicting objectives. In the multi-objective problems "the quality of a solution is defined by its performance in relation to several, possibly conflicting, objectives" (Eiben and Smith, 2007: 165). This technique is highly relevant to architectural design problems, where the design objectives are many and they are correlated and constrained. Here the problem is not only that there are many objectives and that they affect each other, but also that they are to some extend incomparable, because they represent different categories. Are some building characteristics more important than others? Is the efficient room layout more important than a good distribution of daylight? Such questions reappear over the whole design process. How to combine the building characteristics algebraically in one fitness function, to get one quantity of overall 'fitness' of a building model?

A typical approach would be to define arithmetic 'weights' that would label the importance of each building characteristic. Such a set of weights would be inputted to the system by a user at the outset of a generation process. This approach practically converts a multiple-criteria problem, into a single-criterion problem, which is formulated as a weighted sum of functions for each building characteristic. The equation for a fitness function would look as follows:

$$\text{fitness function} = \Sigma\ W_i \text{ x } b\_char_i(BM)$$

Where W_i is a weight, a parameter defining an importance of a building characteristic, b_char_i is the respective building characteristic, and BM is the building model that includes all the characteristics. Furthermore, the building characteristics that are irrelevant would have a weight of zero, and the building characteristics which are undesirable would have negative weights.

An alternative approach would be to “present the user with a diverse set of possible solutions, representing a range of different trade-offs between objectives” (Eiben and Smith, 2007: 166). It would be especially interesting to apply this approach to the generative design system. The system would generate not just one best building model, but rather a spectrum of them, offering the various ‘trade-offs’ between the conflicting characteristics. Such approached, the generative design system would function as a ‘provider’ of choices that are possible in a given design situation (e.g., what can be realised at what cost). The following paragraphs discuss this alternative closer.

PARETO FRONT

The previous paragraphs discussed those multi-objective problems, in which maximizing one objective leads to the weakening of another objective (the objectives oppose each other). Now, one can imagine design situations in which there are many objectives, but they are not in opposition, and in which all the desired building characteristics are to a high degree achievable in one building model. In this case, the only difficulty is to find the right model. The design process is then similar to a search for a solution to a well-defined problem. The fact that all the desired building characteristics are achievable in one building model implies that this model is better than other models in respect to any building characteristic (the ‘best’ building model). In EC terminology, such a model *dominates* other possible models.

The illustration of this concept can be a hypothetic design situation, where there is no budget limit to the design and the only objective is a low energy consumption of the building and a functional room layout. Assuming that low energy consumption and a functional room layout are not in conflict, the dilemma would be to maximise them. Theoretically, it would be possible to find one building model that is better than all the others in respect to these two objectives.

One simplification of this situation consists in the fact, that a designer’s only concern is to satisfy client’s intention, regardless environmental, social or other design aspects. In a design practice, such situations hardly ever happen. Rather, a design process always consists of choices between more or less important objectives. A designer typically has to deal with alternative building models, each of which meets only a few of a number of conflicting design objectives. In contrast to the dominated building models, these building models are called *nondominated*. The fitness of any nondominated building model cannot be increased with respect to any of the building characteristics without negatively affecting one of the other characteristics. Furthermore, the best nondominated building models usually lie on the edge of feasible regions of the search space. The set of all nondominated solutions is called a *Pareto set* or a *Pareto front*.

A practical application of the Pareto front for design might consist for example in classifying all the building characteristics as either benefits or costs (Williamson et al., 2003). The proposed distinction seems reasonable, as the building costs are special building characteristic in the sense that they may oppose a large number of other building characteristics. In this case, there would be two nondominated building models: the cheapest one and the most 'functional' one. In figure 20, a thick black line represents a Pareto front – the collection of best possible trade-offs of benefits and costs. All the solutions laying at the Pareto front are good solutions – nondominated ones. The grey area is the set of all feasible solutions.

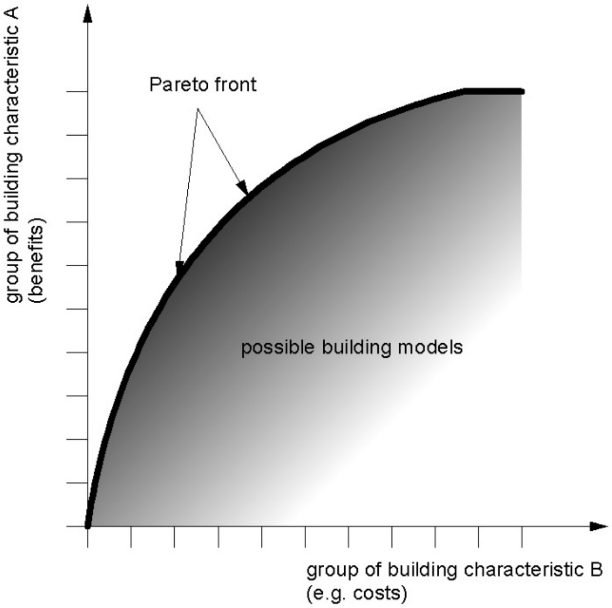


Figure 20. Pareto front for benefits and costs of a building.

There is no method for choosing the 'absolute' best solution from the Pareto front. The selection should be done subjectively, by a user – an architect or a client, either by maximizing one of the opposing groups of building characteristics (either benefits or low cost) or by finding tolerable compromise between them (maximum benefits within an acceptable cost).

The adaptive landscape graphs presented above illustrated problems that were two-dimensional and continuous, but the combinations of 'real' building characteristics would make the adaptive landscapes both highly dimensional and discontinuous. Again, evolutionary algorithms seem to be an adequate tool as they "have a proven ability to identify high-quality solutions in high-dimensional search spaces containing difficult features such as discontinuities and multiple constraints." (Eiben and Smith, 2007: 167)

There are voices sceptical about the Pareto front approach, arguing that it is better to focus on dominating solutions and to set the evolutionary algorithm so as to keep the diversity of solutions throughout the evolution process. In particular, Goldberg (1989) advocated the environmental niches concept and the idea of simulating species.

Concluding, design problems are a class of multimodal problems, because they involve several design objectives (in form of building characteristics of intended building) at the same time. This kind of problems can be approached in two different ways:

- A multi-objective problem can be 'reduced' to a single-objective problem by a system of 'weights'. Each design objective (building characteristic) would be associated with and multiplied by a parameter (a weight), describing a rank of the respective objective. The new single objective would be a sum of such products.
- A diversification of building models population can be applied. Here, the strategy would consist in simulating environmental niches, in which the building models would evolve in parallel, adjusting to carefully selected subsets of design objectives. Especially beneficial would be a distribution of design objectives, in which each niche consisted of design objectives that do not oppose each other. In this manner, the system could generate a dominated building model in each niche – a model that maximizes all design objectives that a given niche includes. The generative design system would then offer building models, which are best trade-offs between the subsets of design objectives in form of a Pareto front. So, in contrast to the former approach involving weights, where there would be only *one* best solution, this approach would generate a *set* of solutions, which would be best compromises between conflicting groups of objectives.

4. APPROPRIATENESS OF EC TO DESIGN GENERATION

In a paper presenting a prototype of a generative design system, O'Reilly and Ramachandran argue that application of EC to architecture may be fruitful: “(…) computational evolutionary process (…) is ideally suited to the process of architectural design. An evolutionary process exemplifies the ‘explore, evaluate and refine’ sub processes of architectural design and its overall non-linear nature” (O'Reilly and Ramachandran, 1999:1) A general argument for the relevance of EC for design generation is that EC was purposefully developed to address the type of problems that are present in complex adaptive systems. The EC mechanism is based on the mechanism that a complex system uses to adapt to the environment. In architectural design process, the following problems can appear:

1. The enumerative method of searching for the best building model (by checking all the possible models one by one) is practically impossible because the number of possible building models is extremely large.
2. The ‘best’ building model is difficult to achieve because:
    a. The building model consists of building elements that are in complex relations and it is not easy to find the configuration of elements that would form a model of expected qualities.
    b. The quality of a building model can be defined from a number of perspectives and there are many different building models that all have their advantages and disadvantages
    c. The quality of a building model depends on conditions that may change over time.
    d. A building model developed as a solution for a set of design objectives may turn out to have undesired properties. This might necessitate redefinition of the initial design objectives.

Similar problems are present in the context of the adapting systems:

1. The enumerative method (as a searching method for best performance) cannot be used because the set of possible transformations of a system structure is large - there are too many alternatives to test.
2. The good performance is difficult to achieve, because:

a. The structure of the system is composed of a great number of elements in complex relations, and so it is difficult to determine which substructures or components are responsible for the good overall performance.
b. The measure of performance (a fitness function) is complicated and involves many independent variables (it has many dimensions, is nonlinear, exhibits local optima, discontinuities, etc.)
c. The fitness of a system varies over time and space and therefore a particular adaptation may not be useful everywhere and in every case.
d. The environment consists of a flux of information, which must be filtered and sorted for relevance (Holland, 1975: 5).

In the following paragraphs I will address some of these points in the context of EC.

LARGE SEARCH SPACE

The complexity of spatial relations amongst the elements of a building model is very apparent and well known to designers. Even if the amount of building elements and their types were very limited, the number of possible configurations would still be vast. It is hard to imagine any algorithm that could explore all the possible combinations in an enumerative way, assessing them one by one. Specifically, in more complete and detailed building models the number of potential solutions grows exponentially with the number of variables considered. Because the space of all possible building models is so large, one would never know whether there is not a better building model.

Exploring a high number of possibilities with limited resources is a common problem in nature (Holland, 1975). The location of relevant information in the environment is unknown to an adaptive system and therefore the space is explored by random probing, which is targeted at finding valuable information as soon as possible. But randomness is replaced by ‘deterministic’ rules, once relevant information is found. The behaviour becomes adjusted – the system allocates a part of its searching resources to the area where the relevant information is located. Moreover, the system has a ‘memory’ – a record of preceding searches that ensures that the solutions already explored are not considered again.

Both an individual organism and a whole species can be understood in terms of adapting systems. In the latter case, the memory of a species is recorded in the structure of the organisms representing the species. A series of successful structural transformations, which made the organism well fitted to the environment, is a 'record' of the survival information.

The mechanism described is in principle similar to the design process. At the outset of the design process, the design actions are random and arbitrary to a large extent, becoming more adjusted and deliberate over time. The designer obtains information from simulations, visualisations or 'mental experiments' on how a building model would work in the physical reality. These simulations and visualisations can be compared to the organism's probing of the environment in searching for relevant information. After encountering the relevant information the organism focuses on this information, analogously to a designer who refines the promising design solutions. Furthermore, a current generation of building models is a record of the 'successful' realisations of the required building characteristics.

EC is inspired by natural adapting systems, providing mechanisms that are relevant for exploring vast 'space' of potential design solutions. Based on the behaviour of adapting systems, EC combines a random search with a 'deterministic' method (utilizing problem-specific knowledge), which makes the exploration of vast solution spaces possible. As adapting systems, EC simulates a memory – a progress in evolutionary search is recorded in the structure of the individuals.

'WICKED' PROBLEMS

The problem is not only the large search space, but also that it cannot be fully defined. The search space for a building model, or any object of design, cannot be formulated in the same precise sense as a mathematical problem (for example the 'traveling salesman' problem). The search space is not 'given' but should rather be 'constructed', and also in this sense it is difficult to speak of an optimal solution.

Moreover, because the architectural problems are 'wicked', it is not only difficult to find a building model that would meet the expected qualities, but there is no consensus as to what exactly the 'expected' qualities are. Without precise criteria of the optimal design, computational techniques should be programmed to search for 'satisfactory' solution rather.

Holland speaks in a similar manner, in the context of adapting systems. He points out that the fitness of a system is relative to the environment and given adaptations are advantageous only at certain places and times (Holland, 1975: 5). An organism removed from its environmental niche might stop functioning. Similarly, a design solution working well at one place and time might not work at a different place and time. So, putting too much effort in fine-tuning of building models might not be reasonable, because a change in their surroundings (e.g., erecting a massive structure in the close neighbourhood) would make the building no longer perfectly fitted (a shadow from the massive structure would change the energy performance of the building or would obstruct the view from windows). All depends on a specific situation and the building model must be regarded in a broad design context. EC approaches for the multi-modal problems find their application here. For example one should avoid over-fitness of a building model when there is a high probability that the building environment (site constraints, client's intentions etc.) might change in a short time.

'MULTIDIMENSIONALITY' OF A FITNESS FUNCTION

In his list of problems for adapting systems, Holland shows that the formulation of a performance measure is complicated because it consists of many interrelated variables. It is difficult to find the best combination of desirable features because they affect each other, or even exclude each other (a heavy shield of a tortoise is a good protection, but makes it a bad runner, a large animal might be strong, but dependent on food shortages etc.) Although most adaptive properties are useful and desirable it is physically impossible to fit one organism with all of them.

Likewise, in a context of a building model, desirable building characteristics often work against each other. In the case of a house, for instance, bigger windows provide more sunlight but they affect negatively energy efficiency; and a complex, aesthetically rich form usually increases the cost of the building.

EC approaches these problems using a technique known as multi-objective problems. There are two main approaches to multi-objective problems. In the first approach, a multi-objective problem can be 'reduced' to a single-objective problem by a system of 'weights'. The generative design system would produce a single, best-fitted building model. In the second approach, a simulation of environmental niches can be applied, in which the building models evolve in parallel, adjusting to carefully selected subsets of design objectives. The generative design system would then offer a number of building models, which are best trade-offs between the subsets of design objectives.

UNANTICIPATED SOLUTIONS

The exploratory function of EC is very attractive if one considers its capacity to find unexpected and unconventional solutions. This capacity is an effect of how the algorithm approaches the solution. Unlike a human being, the algorithm does not have tacit design knowledge, and it may therefore generate random rather than knowledge-based solutions. As Eiben and Smith point out, the evolution:

> (…) is not limited by conventions, aesthetic considerations, or ungrounded preferences for symmetry. On the contrary, it is purely driven by quality, and thereby it can come to solutions that lie outside of the scope of human thinking, with its implicit and unconscious limitations. (Eiben and Smith, 2007: 10)

On the other hand, such an unlimited approach has to be monitored to some degree, especially in the design fields, where a high number of possible structures may be architecturally meaningless. If a building model were not developed in a specific direction, it may not be useful at all. One should distinguish the unwanted from the unexpected, the latter being actually desired in a creative process.

The following example illustrates how EC can generate unexpected solution. In this example, Mitchell (2009: 140) compares two solutions for a problem: M – her solution, which was consciously planned, and G – a solution that was found by an evolutionary algorithm. What strikes the author is the fact that although the two solutions were very similar in many ways, the G solution had a few ‘nuances’ that made it slightly better than her solution. Mitchell concluded that even though she knew that her solution was not perfect, these nuances developed by evolutionary algorithm never occurred to her. Because the evolutionary algorithm develops solution in a different way than human brain, it often comes up with solutions that humans do not even consider. The evolutionary algorithm developed a strategy which seemed uncommon and ineffective at first glance, but which actually performed very well. This false impression of ‘strangeness’ makes the solution G so difficult to invent for a human.

> (…) in real-world applications, the GA will often evolve a solution that works, but it’s hard to see why it works. That is often because GAs find good solutions that are

> quite different from the ones humans would come up with. (Mitchell, 2009: 142)

Lock makes a similar observation:

> Evolutionary algorithms are great tool for exploring the dark corners of design space. You show (your designs) to people with 25 years' experience in the industry and they say 'Wow, does that really work? (Jason Lock, in: Mitchell, 2009: 143)

This property of EC could without doubt be beneficial in design. The unanticipated and clever solutions are always well perceived in architecture.

SUMMARY

EC seems relevant to design because of the following reasons.

- Firstly, it is appropriate for exploring the large search spaces of architectural design.
- Secondly, it can handle the wicked problems – it does not search for an optimal solution in a mathematical sense, but for the best solution available in a particular context. Consequently, it can quickly identify an interesting area of the search space and investigate it further.
- Thirdly, EC can deal with 'multidimensional' problems in architectural design, where many different types of building characteristics have to be considered and evaluated simultaneously.
- Fourthly, because the algorithm is not biased, it can identify solutions which a human mind may overlook because they seem 'strange'. These unexpected solutions are very valuable in a design process.

PRINCIPLES FOR THE GENERATIVE DESIGN PROCESS

I propose the following list of principles for the generative design system, related to Evolutionary Computing:

1. The evolutionary algorithm should apply *mutation rate control.* It is more desirable to have a large mutation rate at the beginning of the generation process and then to focus on refining the solutions that have been found so far. Due to large mutation rates, a large part of the search space could be examined at the beginning of the search process, increasing probability of finding good solutions.

2. The algorithm should use *parameter adjustment*. The building model's development could be improved by application of a dynamic adjustment of such parameters as mutation rate control or probability of mutation and crossover. Practically, these parameters would be subject of the evolutionary process. For example the level of decomposition of a building model can be a parameter that is adjusted dynamically. The decomposition can be deeper at the beginning of the adaptation and decrease gradually towards the end of the process.
3. The evolutionary algorithm should apply a *parallel-terraced scan*. This technique optimally allocates the available searching resources in order to find the best balance between exploration and exploitation of a search space. Thus, at the initial phase of the search (generation) process, building models should be very different from each other, exploring possibly large space of design versions. Successively, the algorithm should focus on refining a few best-fitted building models. At the same time, the algorithm should allocate a fragment of the search resources on random search for very different alternatives.
4. The algorithm should employ *penalty functions* – a method for eliminating deficient (for example spatially inconsistent) building models from evolving population.
5. The algorithm should use a *tournament selection*. It is a method of *relative* evaluation of building models. Instead of evaluating a building in absolute terms (for example by points) the algorithm should indicate which model is 'better' in a direct comparison ('tournament') concerning selected building characteristic. For example a real cost of a building does not have to be determined; instead, the algorithm can indicate which of a few selected buildings would be more costly. Alternatively, the algorithm might use representational units for rating building models.
6. A *user intervention* in the process of building models evaluation should be allowed. A user should be able to monitor the adaptation process and intervene in defined periods to support the automatic evaluation of building models. This seems desirable because of the qualitative nature of some building characteristics (such as a building shape) which make an automatic evaluation problematic.
7. *Multimodal problems*, *multi-objective problems* and the *Pareto front* are tactics from which a generative design system could probably benefit most. These tactics involve evolution that considers many different objectives (often opposing each other) at the same time. Design problems are a class of multimodal problems, because they

involve several design objectives (the building environment is inclusive). There are two main approaches to multi-objective problems. In the first approach, a multi-objective problem can be ‘reduced’ to a single-objective problem by a system of ‘weights’. The generative design system would then produce a single, best-fitted building model. In the second approach, a simulation of environmental niches can be applied, in which the building models evolve in parallel, adjusting to carefully selected subsets of design objectives. The generative design system would then offer a number of building models, which are best trade-offs between the subsets of design objectives (a Pareto front).

SECTION IX – ANALYSIS OF PROTOTYPES OF GENERATIVE DESIGN SYSTEMS

In this section I make an analysis of four realized generative design systems. These four prototypes are discussed in four papers published as proceedings of selected conferences on the generative design. My method of selection of the articles is described in section I. The purpose of the analysis is to answer the question why these design systems did not find a broader application in an actual design process. In the analysis, I employ the theoretical model for a generative design system that I developed in the previous sections. Specifically, I compare the principles of the analysed generative design systems with the principles developed by me so far, and grouped around such themes as:

- the architectural design process,
- the building model,
- the building characteristics,
- the building environment,
- the evolutionary algorithm.

Particularly, the analysis focuses on:

- how a building model is constructed (specification of building model elements);
- how a building model environment is specified (which building characteristics a given system contains and how a fitness function is implemented).

Additionally, the following questions are posed:

- Does a given system examine housing design?
- Does it examine plan solutions?
- What problems do the authors identify?
- What group of users does the system aim at?

This section is concluded with a summary of findings about these generative design systems. The main drawback of all these prototypes is *low resolution* of their building environments, i.e., the fact that these systems take into account only a few building characteristics.

1. “ARCHITECTURAL CONSTRAINTS IN A GENERATIVE DESIGN SYSTEM: INTERPRETING ENERGY CONSUMPTION LEVELS”

The paper was presented during the Seventh International IBPSA (the International Building Performance Simulation Association) Conference in Rio de Janeiro, Brazil, in August 2001. The authors are Luisa Caldas and Leslie Norfold, from the Massachusetts Institute of Technology. The theme of the paper is the generative design system understood as a support for architects during the design process; it achieves this by “identifying potentially problematic areas and suggesting ways to approach them” (Caldas and Norfold, 2001: 1397). The generative design system consists of:

- a genetic algorithm, which is used for searching and optimising solutions;
- a building simulation program, which is used as an evaluation tool.

The building simulation software - a program called DOE-2 - applied in the experiment assesses the use of natural lighting, thermal performance and energy consumption. The generative algorithm starts by generating a number of solutions, and then calculates their fitness and applies the basic genetic operators of reproducibility, crossover and mutation. This procedure generates a new population with a higher average fitness, which in turn undergoes the same procedure of evaluation and then mutation and reproduction. The procedure is repeated over a number of cycles.

Specifically, the system was applied to the study of alternative facade solutions. The object of investigation was The School of Architecture at Porto, Portugal designed by Alvaro Siza. The building was chosen because of its complex facade composition and because of interesting relations between the different spaces and their light sources. The interactions amongst these elements (possibilities of setting them in different configurations) provided a good framework to work on a genetic algorithm. Particularly, the objectives were to:

- Keep Alvaro Siza’s design language constraints.
- Analyse the interaction of light and thermal performance of the building.

A particularity of this study is that Caldas and Norfold used a micro generative algorithm instead of a conventional genetic algorithm. The difference lied in the population size, which in this study was limited to five individuals, while a typical range varies from 30 to 200 individuals.

Furthermore, the generative system worked on a three-dimensional description of the building (building model), including its:

- geometry,
- orientation,
- spatial organisation,
- construction,
- materials.

Unfortunately, in this study the search space does not include different forms of building models but is limited to facade design solutions. The generative system takes into consideration the following building characteristics:

- use of the daylight in the space (the distribution of natural light in the building is measured and controlled in selected points of the building, which is important in a school building);
- energy consumption to heat and cool the building (annual energy consumption).

There are correlations between these building characteristics. For example, increasing the use of daylight (a desirable effect) requires larger windows, which increases heat losses (an undesirable effect). Finding a compromise between these two building characteristics is the goal of the generative algorithm.

The composition rules for the facade were investigated in order to encode the design intentions of Alvaro Siza. These design intentions included:

1. compositional axes of the facades;
2. general proportion of the openings;
3. other rules (for example long horizontal window always used in the studios);
4. the fact that the south elevation presents a strong symmetry axis for the openings and introduces other elements, as overhangs and the loggia.

The interpretation of the rules defines the search space for a generative algorithm (in particular, the maximum and minimum dimensions of the openings, the compositional axis, and the step size for window and for exterior shaders)[34]. The figure below shows the facade solutions generated by the system (the red line shows the compositional constraints).

---

[34] What is interesting in this example is the fact that the authors considered the aesthetic qualities of the facade, being aware that these qualities have to be somehow preserved in the generated solutions. Unfortunately, the artistic qualities – instead of being a part of the fitness function - are implemented into the system as a set of constraints limiting the search space. Such approach is very case-specific because it limits

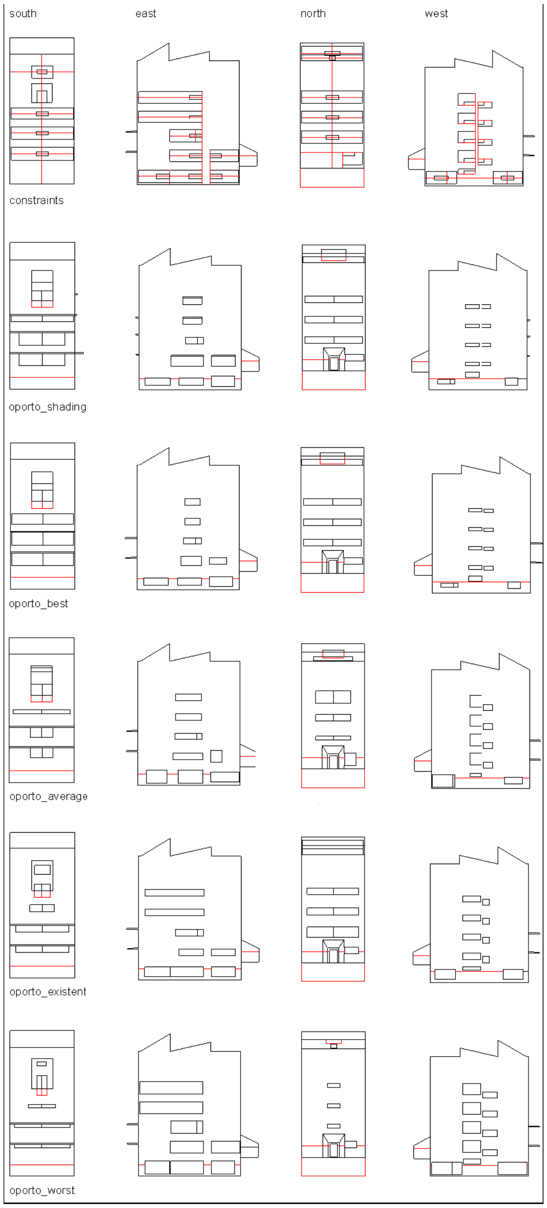


Figure 21. Facades generated by the system for the School of Architecture in Porto (authors: Caldas and Norfold, 2001).

---

the definition of the building model (and consequently the search space), making the system applicable only to the School of Architecture in Porto.

Caldas and Norfold illustrated the results using a CAD system and manually building and rendering a 3D model from the output data. The figure below shows the 3D building models.

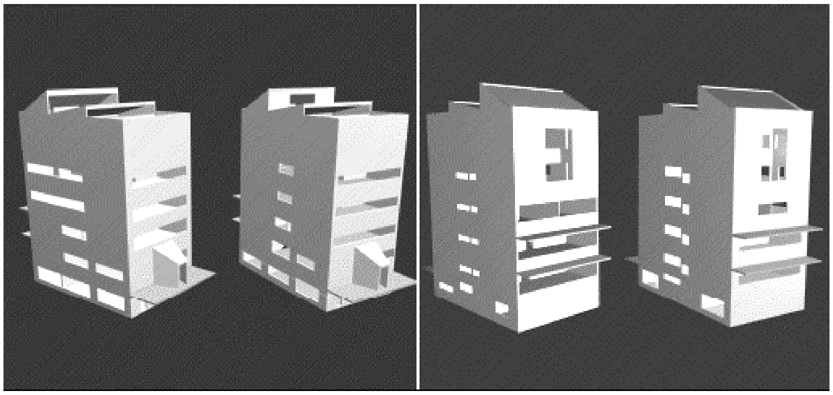


Figure 22. Visualisation of the *building model*s generated by the system (authors: Caldas and Norfold, 2001).

In terms of energy efficiency, the facade solutions provided by the algorithm were on average 10% better than Alvaro Siza's original solution. Furthermore, the new design solution kept the illumination levels inside the selected spaces at the same, required level.

Caldas and Norfold concluded that the generative design system might find its application in the early, conceptual phases of the design process. Moreover, the authors emphasised the exploratory character of the generative process:

> Solutions must not be interpreted as definite or optimal answers, but as diagnoses of potential problems and as suggestions for further architectural explorations, thereby building an innovative and promising interaction between architecture and computation. (Caldas and Norfold, 2001: 1404)

Finally, the authors used the system to investigate the building's performance in other locations (at different sunlight conditions). They argued that the findings of such theoretical experiments must be interpreted in the broader context of the application:

> A note should be added here that the exercise of placing the building in a different geographical location is purely academic, since Álvaro Siza would never use the same approach for designing a building in such different geographical and cultural environments. While the experiments for Oporto illustrate a process that could have taken place in Siza's design, the other examples are a test of the generative system but do not represent an appropriate architectural design process. (Caldas and Norfold, 2001: 1399)

The authors point at the interesting fact that GA keeps a holistic structure of a building model, 'embracing' the building in its totality:

> Another interesting dimension of the GS is its capability to account for interactions between different elements of the building and make the design for each specific element dependent on its integrated role on the architectural whole. The relations between the solutions for the loggia and the roof monitors, or between south and east facing windows in some of the studios, are a demonstration of that capability. (Caldas and Norfold, 2001: 1404)

Caldas and Norfold concluded that the design system can be improved. For future research, they intend to incorporate the building geometry and spatial organisation into the system. It sounds very promising as the two aspects could let the system exceed the present limitation of too rigidly constraint building model.

SUMMARY

In the generative design system presented above, one can discern the following building elements: walls, floors and openings. The spatial configuration of these elements, which was generated by the system, is a parameter of a fitness function.

Caldas and Norfold defined the system's environment using two building characteristics: daylight accessibility in key places and energy consumption level.

The energy consumption level can be classified as a complex building characteristic, because its specification requires complex calculations (using a special program DOE-2). One building characteristic embedded in the system is the aesthetic value of the facade. Unfortunately, aesthetic considerations are not included in the fitness function, and consequently aesthetic quality is not computed in this prototype. It is rather defined by a user's interpretation of the original facade design intentions. It is implemented in the system by limiting the sizes of possible facade elements (thus constraining the search space). Because of that, the system's application needs a user to 'encode' the design intentions for each new building processed by the system. This fact reduces the generative dimension of the system, making it rather an optimisation tool.

The diagram below (based on the triple bottom line and proposed in section VII) is a graphical illustration of the building characteristics that were included in Caldas and Norfold's system. It shows the fragmentariness of the building environment implemented in the system, in comparison with a fully implemented generative design system.

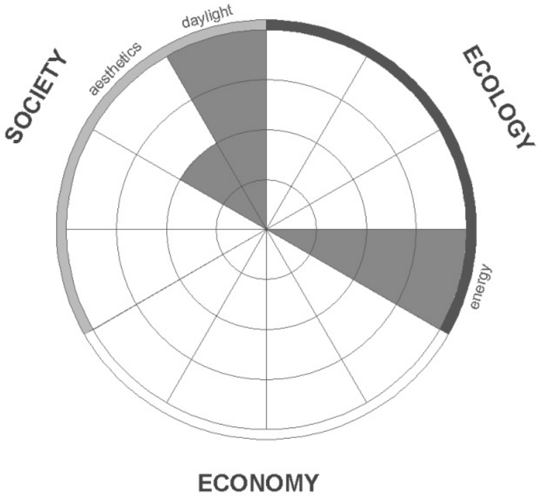


Figure 23. The building characteristics included in the system (grey areas) in relation to the full implementation of a generative design system (whole circle).

Caldas and Norfold's prototypic system investigated a school building with a focus on the interplay of daylight accessibility and energy consumption. Although it was not developed for to housing architecture, it may be applied to it with minor adjustments. On the other hand, the system does not examine plan solutions, which is an essential factor in housing design. Caldas and Norfold propose to address these shortcomings in future research, by including building geometry and spatial organisation in the system.

2. "DUTCH DWELLINGS"

The second design system prototype was presented in the paper "Dutch Dwellings", at the 7th Generative Art Conference in 2004. It was co-authored by Bauke de Vries, Aant van der Zee and John Carp from the Eindhoven University of Technology. The authors made an attempt to apply generative design techniques to support housing design. Their interest focuses on generating plans for a detached house.

De Vries et al. used the Dutch building codes and regulations as a starting point. They applied the codes with respect to lighting, energy consumption and minimal requirements for furnishing. An input for the generative design system was:

1. Minimal and maximal areas of each type of room, taken from a typical Dutch house:
   a. circulation room 2m2 – 10m2
   b. entrée 2m2 – 10m2
   c. kitchen 5m2 – 12m2
   d. living room 20m2 – 37m2
   e. bedroom nr.1 20m2 – 25m2
   f. bedroom nr.2 12m2 – 15m2
   g. bedroom nr.3 12m2 – 15m2
   h. bath 5m2 – 10m2
2) Spaces adjacency, where a 'circulation room' is connected to all spaces, and in addition:
   i. The kitchen is connected to the living room and to the entrée.
   j. The bedroom nr.1 is connected to the bathroom.

De Vries et al. also limited the width between the bearing walls to 6.00 meter, as the most frequently used length in new housing projects in the Netherlands. They further limited the extensions in the front and in the back of the building to 2 meters (see figure 24).

De Vries et al. defined the building environment with the following building characteristics:

- local building code and regulations with respect to lighting, energy consumption and minimal requirements for furnishing;
- minimal and maximal areas for different types of room;
- connections between spaces;
- typical distance between the bearing-walls used in the Dutch housing industry.

The output of the generative algorithm was visualised in the form of coloured, rectangular shapes, which represented the house's plans (three plans for each floor). Additionally, the body of the house was visualized in 3D (figure 25).

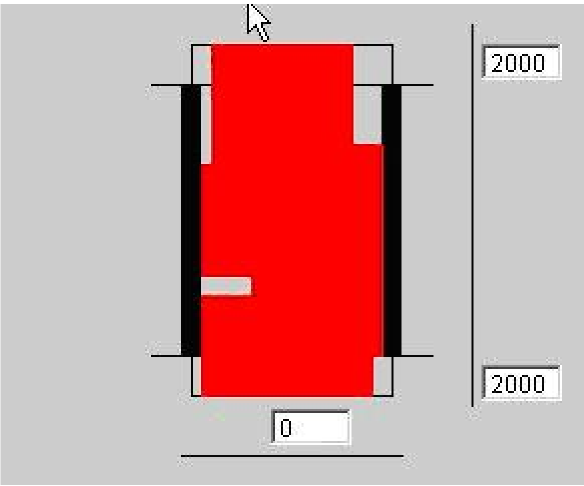


Figure 24. Visualisation of a house plan by the system's interface. The black stripes show the bearing walls (authors: De Vries et al., 2004).

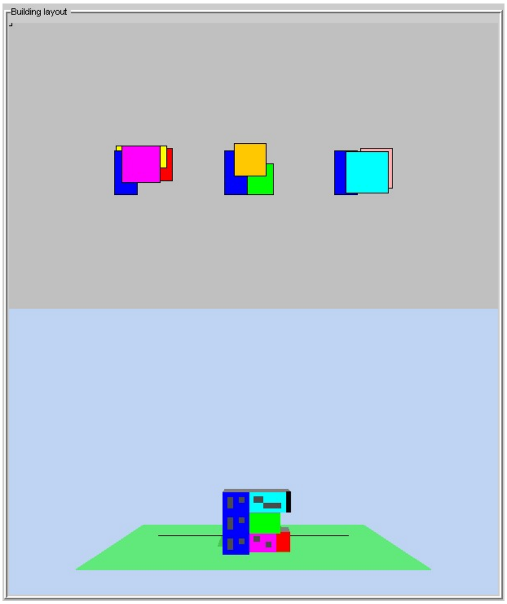


Figure 25. Visualisation of a building model (2D view and 3D view) (authors: De Vries et al., 2004).

The authors interpreted the results by comparing the output of their system to the existing types of Dutch houses. They focused on a Dutch house identity, posing the question: do the properties of the generated house resemble the properties of a traditional Dutch house? The conclusion was that the generative system was capable to reproduce some of the traditional dwelling types. However, the generation of other house types was limited. The authors were unable to explain this shortcoming. They claim that the most probable explanation is that the system was stuck in a local optimum of the fitness function and did not 'discover' other optimums.

SUMMARY

In de Vries et al.'s system, building model elements are not only representatives of physical building elements such as walls, floors and openings, but they also represent the functional spaces of a detached house, such as a living-room, a bedroom, etc. Furthermore, the building environment consists of the following building characteristics:

- local building code and regulations (including lighting, energy consumption and minimal requirements for furnishing);
- areas for different types of room;
- connections between spaces.

Here, unlike in the previous prototype, the light and energy consumption are not determined by a simulation but they are an effect of an implementation of the building codes.

The figure below positions the building characteristics of this system within the characteristics diagram. Again, this approach is limited and includes only a small number of building characteristics. However, in comparison with the system discussed in the previous section, this system contains more building characteristics and is therefore more comprehensive.

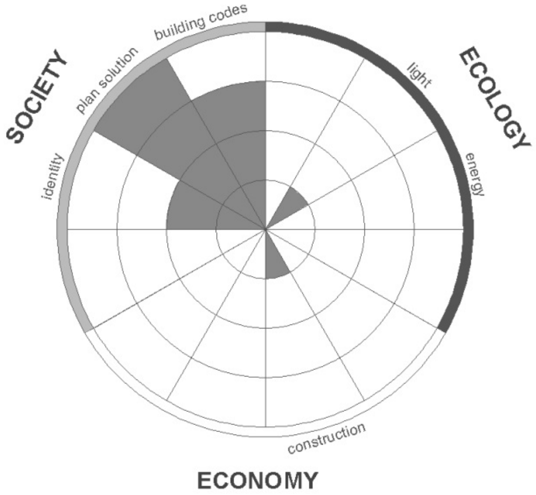


Figure 26. The building characteristics included by the system (grey areas) in relation to the full implementation of a generative design system (whole circle).

Furthermore, the system focuses on housing architecture, especially on the plan solutions, which fits the interest of this thesis. A serious limitation of this approach is the lack of aesthetic considerations, however. Another limitation is that although this system considers more characteristics than the one presented in the previous section, these characteristics are examined in a relatively superficial way, because the system operates only on basic characteristics (no energy consumption or other kinds of simulation). On the other hand, the inclusion of the basic characteristics only speeds up the evolutionary algorithm. This in turn makes possible the inclusion of many different types of characteristics, and thus develops building environment in a more balanced and complete way. Summing up, it seems more desirable to develop the characteristics diagram uniformly, starting from its centre and moving toward its perimeter (if technically possible), than to include in the system a few, well elaborated building characteristics.

3. “EVOLUTION AS A DESIGN STRATEGY FOR NONLINEAR ARCHITECTURE: GENERATIVE MODELLING OF 3-D SURFACES.”

This prototype was developed by Una-May O’Reilly and Girish Ramachandran, both based at the Massachusetts Institute of Technology. O’Reilly and Ramachandran identify the shortcomings within the Computer Aided Design tools. They claim that contemporary CAD systems are not really design tools, but merely automate the drawing process. In order to improve the capabilities of CADs, the authors utilise Evolutionary Computing. The following quotation expresses the authors’ concern:

> A CAD tool is supposed to support and foster the design process. Instead, despite having transformed the studio by replacing drawing boards and drafting equipment, CAD systems are usually passive and do not contribute beyond acting as an electronic pen. (O'Reilly and Ramachandran, 1999: 3)

As a response to the limitation of design capacity of the existing commercial CAAD systems, O’Reilly and Ramachandran propose a prototypic generative software based on evolutionary algorithms – the Generative Genetic Explorer (GGE). In the GGE, “genetic operations interact with AutoCAD to generate novel forms for the architect” (O'Reilly and Ramachandran, 1999: 1). It is apparent that the authors focus on architectural form, especially on its complexity and novelty. Similarly to the systems discussed above, GGE also finds its application in the initial stage of the design process:

> The ultimate goal of GGE is to actively generate and suggest new designs to the architect. GGE is intended to be used during the initial stage of the design process when form is being explored through the use of visual models. (O'Reilly and Ramachandran, 1999: 1)

The system is supposed to function as a source of visual inspiration for designers. The output of the system is visualised with a popular design application - AutoCAD. The system is still a prototype and it is more an ‘investigative platform’ for assessing a-life CAD tool issues than a design aid in an ‘authentic’ design process. It means that its primary purpose is to explore possibilities of evolutionary programming techniques in architectural design.

O'Reilly and Ramachandran discuss how artificial intelligence could be applied to architecture. Their approach consists in recognising the mechanisms observed in natural systems and their algorithmic simulation. The authors use the approach of John Frazer, who argue that an artefact (a building model in particular) should "evolve [in interaction] with natural forces including these of society", and that computer based systems should follow an 'inner logic' of design rather than an 'external form' (Frazer, 1995). Frazer's approach to "the natural forces including these of society" is intuitive and similar to the concept of a building environment; it is also similar to the holistic principle of the triple bottom line. Unfortunately, it seems that the authors did not address many design aspects, but focused on the generation of unexpected and complex forms.

In this prototype, a genotype is defined as a sequence of transformations applied to an initial form. In other words, the form is not encoded in the genotype but exists independently of the genotype. The genotype is 'an instrument' for transformation of the initial form. The mutation here consists in changing the sequence of transformations that are defined in the genotype.

The building model is defined in a 3D space; it distinguishes the vertices, profiles and planes that constitute the building elements. The possible transformations defining the genotype are:

- insertion (inserts randomly one or more profiles to the initial form);
- deletion (opposite of insert);
- folding (rotates a plane);
- lifting (increments or decrements a subset of vertices);
- poking-hole (removes skin from the form's skeleton leaving a hole).

Figure 27 shows a series of building models generated by the system:

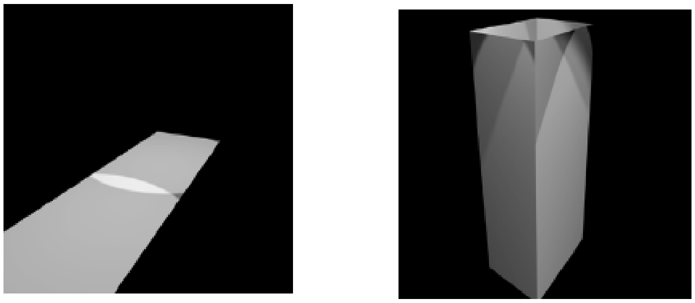


Figure 1: Initial Form: plane-0 Figure 4: Initial Form: cube-0

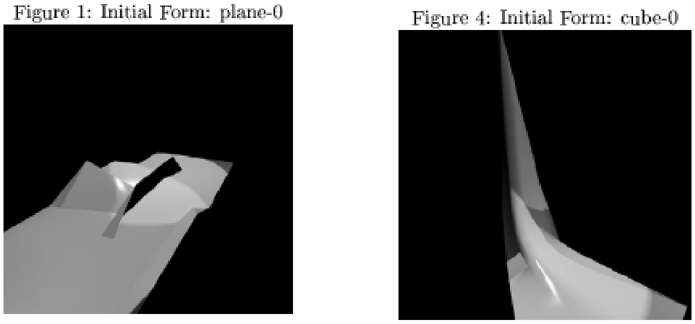


Figure 2: plane-1, transformation of plane-0 Figure 5: Cube-1, transformation of cube-0

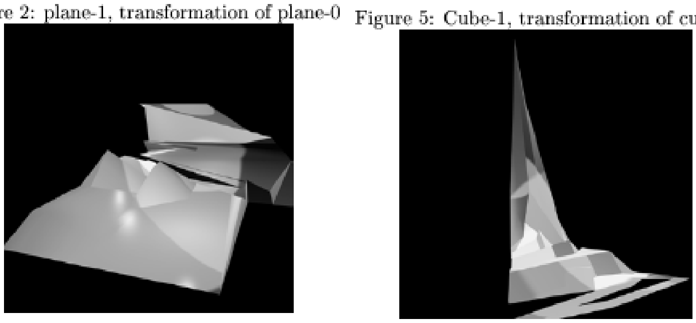


Figure 3: plane-2, transformation of plane-1 Figure 6: Cube-2, transformation of cube-1

Figure 27. Examples of building models generated by the Generative Genetic Explorer (authors: O'Reilly and Ramachandran, 1999).

One weakness of this system is the lack of automatic evaluation of the generated building models (the lack of a fitness function). Instead, the user acts as the evaluator, by assessing the generated models (displayed on a screen) and manually selecting the best one on the basis of his or her own taste. In comparison with the other generative design systems presented in this chapter, the lack of implemented fitness function is a disadvantage of this approach.

SUMMARY

O'Reilly and Ramachandran admit that their system is far from being a complete architectural design aid. Nevertheless, this system showed that:

1. Evolutionary algorithms are well suited for CAAD systems, though they need to be constrained in terms of random variation - the evolution needs to be controlled.
2. The architect must interact with the generative design system during a generation process - a system cannot operate independently of the architect. The authors did not suggest an alternative way of controlling the evolution of building models.
3. A genotype representation applies the architect's vocabulary (transformations such as insert, delete, fold, lift, etc.) and is therefore mediated by human spatial intuitions that facilitate the understanding of the tool.

Building elements are vertices, profiles and planes represented in a 3D space. O'Reilly and Ramachandran emphasise the importance of the complexity and the novelty of the buildings' forms, and they state that the first function of the generative system is to inspire the user. Because the system does not define the building environment, it largely depends on user's intervention and on the random process of genome modification.

Figure 28 positions the building characteristics of this system - a complexity and novelty of a form - in the characteristics map. Typically, the aesthetic characteristics are regarded as subjective, vague and difficult to express. Because they are indescribable in terms of algorithmic definition, the authors left the evaluation of the aesthetic value to the user. The user's intervention during the evolution process seems unavoidable and beneficial. It seems beneficial to involve the user in order to avoid the automatic evaluations of the building characteristics, which are likely to be subjective. Because the system focuses only on the complexity and novelty of the building form, it is limited and relative to the whole spectrum of possible building characteristics.

This system has no special interest in housing design and it can be applied to any type of building.

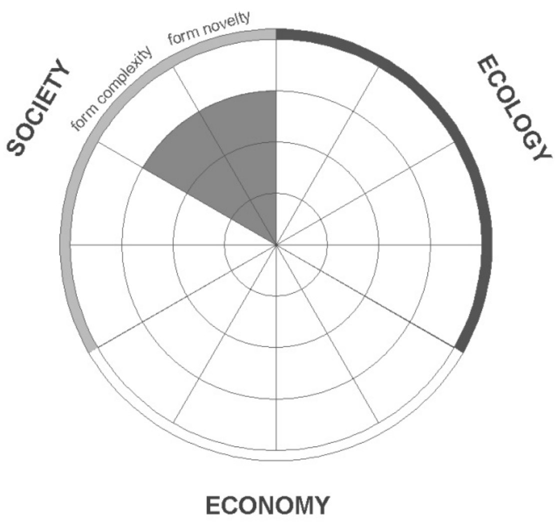


Figure 28. The building characteristics included in the system (grey areas) in relation to the full implementation of a generative design system (whole circle).

4. “PERFORMANCE-BASED DESIGN EVOLUTION: THE USE OF GENETIC ALGORITHMS AND CFD.”

The fourth prototype of a generative design system was presented at the Eighth International Building Performance Simulation Association (IBPSA) Conference in Eindhoven, Netherlands, in August 2003. It was developed by Ali M. Malkawi, Ravi S. Srinivasan, Yun Kyu Yi and Ruchi Choudhary.

Malkavi et al. focus on the thermal and ventilation performance of a building model. The authors underline the complexity of a projected design system. The complexity arises from the fact that the system operates on two complex building characteristics. They classify the problem as an optimisation problem and they propose to use evolutionary computation to approach it. Their argument is based on the fact that the search space is difficult to explore systematically because the complexity of the model makes a trade-off between different design changes obscure. Here again, the authors advocate the evolutionary approach to design, stating that it is "a generate-and-test approach which corresponds well to the procedures for design synthesis and evaluation in the design process" (Malkavi et al., 2003: 793).

Furthermore, the authors emphasise the exploratory character of the evolutionary approach in architectural design. The generative system can act as a source of inspiration for a designer or as an aid in discovering unanticipated solutions:

> (…) design evolutions can be used as an aid in stimulating the designer creativity (…) The advantage of such an evolutionary approach is the creation of diverse sections of the state space that increases the possibility of discovering a variety of potential solutions." (Malkavi et al., 2003: 793)

Malkavi et al. use the term 'performance-based design', which denotes a design supported by computer simulations of the building performance. Based on the output of the simulations (which specify the complex building characteristics), the building model is further redesigned or refined. The core of the system computes the performance of the building model (in terms of energy conservation and ventilation) and the role of the user is to control the performance parameters and the functionality of the space.

The authors use a 'four-layer' approach:

- design evolution (creating/ modifying a building model);
- performance evaluation (automating the evaluation of a building model with well-defined fitness function);
- morph visualisation (3D visualisation of a set of generated building models);
- design evaluation (evaluation of a building model by a user).

The fitness function is defined by simulation and includes the thermal and ventilation efficiency of a building model. The building models are processed by the genetic algorithm automatically until the best design is found. However, a mechanism called a 'morphing module' enables the user to intervene in the middle of the evolution process. Each such intervention consists firstly in visualising the building models generated so far, and secondly in selecting the most functional or the most interesting model. The selected model can be processed further, for example, its thermal and ventilation efficiency can be optimised. Thus the system combines in an interesting way an automated evaluation of the quantitative building characteristics (which is difficult to achieve by the user) with the user's evaluation of the qualitative building characteristics (which is difficult to achieve by the algorithm).

SUMMARY

In this fourth prototype, the building environment is a combination of a fitness function that includes the thermal and ventilation qualities of a building model, and the user's assessment of its overall quality. The user's involvement in the operation of the system is desirable, because only the user can control the quality of the design and its functionality. The user can intervene in the evolutionary process in the selected moment, completing the evaluation process and 'shifting' the evolution into the right direction. If the evaluation module were proceeding independently, it would regard only the thermal and ventilation qualities of a building model, which in turn could result in generating functionally deficient (or aesthetically unappealing) models.

When it comes to the building elements, in this prototype they are limited to those which define a cubical room, i.e., the length and the width of the room, the position and the size of a window and a door, the area of a supply duct, etc.. The authors focus on the optimisation of the room's performance rather than on the creative potential of the system.

Because the variety and the number of building elements in this system are very limited it is difficult to define many building characteristics. The building characteristics processed automatically by the system are the thermal and ventilation performance. They both are complex building characteristics. The figure below positions the building characteristics of this system in a building characteristics map.

The system does not operate within housing architecture, nevertheless it can be used as an optimisation tool for some types of room in the housing architecture.

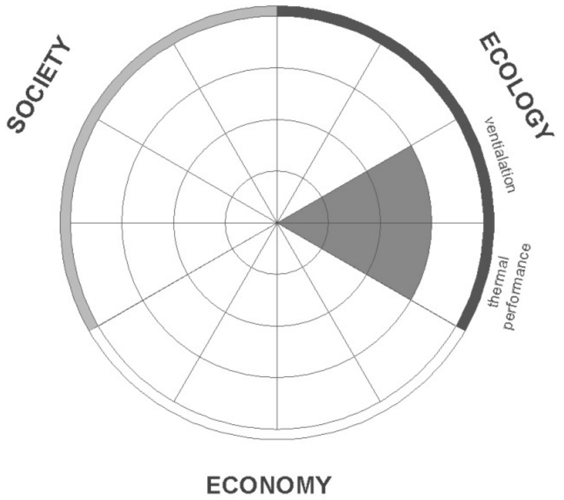


Figure 29. The building characteristics included in the system (grey areas) in relation to the full implementation of a generative design system (whole circle).

5. FINDINGS ABOUT THE GENERATIVE DESIGN PROTOTYPES

This section contains conclusions about the four prototypes presented above. The prototypes' principles and the presumed principles of the generative design system developed throughout sections IV-VIII are compared and discussed. I focus on:

1. the potential pitfalls or limitations of the prototypes and how a given prototype could benefit from the presumed principles;
2. the advantages and benefits of the prototypes' principles and how these principles could be verified and improved.

FINDINGS RELATED TO A BUILDING MODEL

To denote a building model, the authors of the prototypes discussed use terms such as: an individual, a (design) solution, a phenotype, a (generated) design, a (generated) model, a (generated) building. The building models of all the prototypes are composed of clearly defined sets of building elements. The 'operational space' is three-dimensional and in most of the cases the requirement of spatial coherence (e.g., that the building elements do not overlap and that the walls/ windows have no random breaks, etc.) is fulfilled. An exception is prototype 3, in which a building model is represented in a quite an abstract way; this makes it difficult to relate its elements to their physical representatives such as walls and windows (in this prototype, building elements are vertices, faces and surfaces). Prototype 3 approaches a building model generation from an aesthetic point of view. The focus is on shaping a complex, unexpected, 'sculptural' form, which can be interpreted in many ways, and which is thus meant to be an inspiration for an architect. Prototype 3 lets the user operate directly on the level of the building elements (vertices) with the following possibilities of modification: insert, delete, fold, lift, poke-hole.

Unlike prototype 3, prototype 2 relates a building model and its structure to the physical building elements (walls). Moreover, it combines the walls into functional entities, like rooms, which can be interpreted as groups of elements. It also makes calculations determining areas of the rooms, and it examines the spatial adjacencies of spaces. Prototype 2 further includes a basic hierarchical structure of the building model. The building model of prototype 2 is a relatively balanced representation of a detached, 2-storey house.

The construction of the building model of prototype 1 is based on facades, and it consists of walls, windows and roofs, which are distributed over the facades on a predefined, constant volume. The definitions of the complex building characteristics such as energy performance and daylight illumination are realistic, and they require complex simulation and computing.

The situation is similar in prototype 4. Here the general shape of the building model is predefined – it is a single room. Building elements are windows, doors and an air supply pipes that are distributed over the walls of the room. As in prototype 1, the complex characteristics – thermal and ventilation performances - are specified through simulation.

Unfortunately, none of these prototypes attempts to define a comprehensive set of building elements and building characteristics, making their building models comparable to those currently used in BIM systems. A more comprehensive definition of building models would make the solutions more unanticipated and interesting, because the size of the search space would significantly increase. Moreover, a more comprehensive building model implementation would provide more elaborated building environments.

FINDINGS RELATED TO A BUILDING ENVIRONMENT

In all the prototypes analysed, there was no clear distinction between the design requirements and the design intentions.

When it comes to site constraints, only prototype 1 takes into account the geographical location of a building. The intensity of the sun in a given location plays a crucial role for the thermal performance of the building. Other site parameters, such as the shape of an allotment or the form of a terrain are not included in the system. There is no need, however, to consider other site parameters in prototype 1, because it does not operate on the form of the building model, but only on the facade solutions. Consequently, a building model definition that is limited makes a broader definition of site constraints unnecessary. The three remaining prototypes do not take into consideration the site constrains and their potential effect on building models.

Only prototype 2 encompasses building codes. The local building codes are applied in terms of lighting, energy consumption and minimal space for furnishing.

None of the prototypes directly refers to the master plan regulations. The authors probably assumed that the master plan regulations should be inputted as a part of the design intentions.

The prototypes discussed make no distinction between the client's intentions and the architectural qualities. In prototype 1 and 4, the genetic algorithms serve more as a design optimiser (in both cases energy efficiency was optimised) than a fully operative design generator. Prototype 1 is not capable of altering a building model's form while prototype 4 reduces the form to a box of various sizes. On the other hand, prototypes 2 and 3 do operate on form. Especially, prototype 3 is focused on form and it can generate very complex and refined forms; however, it neglects all the remaining building characteristics. The building forms generated by prototype 2 are relatively simple as they consist of boxes distributed over two floors. In contrast, they are well balanced by building characteristics. Thus, it seems that prototype 2 is the closest to the approach advocated in this thesis, as it reflects the structure of the environment in a most complete way. It implements local building codes, takes into account construction recommendations, and it considers design intentions such as space adjacencies and rooms' areas (the last two correspond to the client's intentions). Prototype 2 is also interesting because it operates within the housing architecture.

Furthermore, prototype 3 and 4 involve the user in the evaluation process. Though it seems inevitable in the case of architectural design, the strength of Evolutionary Computing is that a large number of evaluation and modification cycles are performed in a relatively short time. The algorithm takes advantage of the computing speed of a machine only if the process is not interrupted by users. Assuming that each evaluation has to be confirmed by a user, the algorithm's ability to generate, evaluate and transform generations of individuals within seconds is compromised. Thus, it is desirable to reduce the user's involvement as much as possible. One solution may be to limit the number of the user's interventions (e.g., to every hundredth cycle). But then, the system's evaluation capacity would be reduced, which could result in undesirable solutions. Specifically, in the case of prototype 4, a building model might become very energy efficient, but it would lack basic functionality or it would be unattractive.

Summing up, the common drawback of all these prototypes and probably the reason why they did not find application in the actual design practice is that they include only a few building characteristics (though their implementation is deep). In other terms, their building environments are deep but they have low resolution. It is worthwhile to recall the principle I propose in section VII, saying, that a building environment development should start with a *high* resolution and low depth. Progressively, the depth could increase, in consideration with the results and performance of the system.

SECTION X – CONCLUSIONS

This section brings together the finding of the thesis. In addition, it shows the contribution of the thesis to the existing body of knowledge and suggests directions for further research. This section consists of four parts.

In the first part I explain the overall functioning of the generative design system, using a graphical illustration (figure 30). Based on that scheme, I list and describe specific elements and principles of the system. Next I describe the benefits of the system for a designer, a client and the nature. Finally, I examine possible challenges and shortcomings of the generative design system, such as for instance management of qualitative building characteristics.

The second part is a supplementary discussion on the generative design system, where I take a more abstract perspective. I begin with a distinction on the differences between a black box and a white box, postulating that the generative design system should work as a black box. Essential property of a black box is that its internal working is unknown and it is controlled only by its input, output and transfer characteristics (a relation between input and output). It is desirable that the internal working of the generative design system is unknown, because only such a system can be creative, i.e., can provide *unexpected* solutions. The input for the generative design system would be building environment (a set of building characteristics of the intended building model) and the output would be a building model that fits with the inputted set of characteristics. On that basis, I propose the concept of *indirect design* – a user should develop a building model controlling building characteristics rather than building elements.

The third part of this section addresses the thesis' contribution to the existing body of knowledge. The main contribution is the development of the elements and principles for the generative design system that are adjusted to the actual design practice. I hope that the theoretical framework provided by my thesis can be used for a further realisation of generative design software. In addition the assumption that a building model is an adapting complex system makes possible application of the principles developed in this thesis to a *common* architectural practice. Specifically, selected mechanisms of

Evolutionary Computing, such as a parallel-terraced scan, can be applied as a design strategy for facilitating design development.

In the final part, I propose directions for further research. The further research should involve realisation of the generative design system. One could gradually implement particular techniques of EC that thesis suggests. Emphasis should be put on techniques addressing multi-objective problems. Important aspect would be a plausible user interface and a functional visual working environment. In addition, the further research should study more in-details selected biological mechanisms, such as for example a self-organizational mechanisms of biological systems.

1. SYNTHESIS OF THE GENERATIVE DESIGN SYSTEM

1.1. FRAMEWORK FOR FUNCTIONING OF THE SYSTEM

This section describes a conceptual framework for functioning of the generative design system. It brings together all the discussed elements and principles and puts it in one functional scheme. Figure 30 shows overall functioning of the generative design system developed in the thesis.

The system's functioning is based on two mechanisms. The first mechanism is called *iteration* and it is controlled by a user. At the outset of the generative process, the user defines a building environment (design objectives) – a set of building characteristics that describe the intended buildings. These characteristics are grouped firstly, according to a design domain they address (society, economy, ecology), and secondly, according to the design constraint they address (site constraints, building codes, master plan regulations, client's intentions and architectural qualities).

The second mechanism is called *adaptation* and it is controlled by the evolutionary algorithm. In the adaptation process, the algorithm generates a building model that fits with the environment defined by the user. This generation process is based on a simplified mechanism of evolution by natural selection, and it proceeds as follows. Firstly, a number of random building models is generated. Next, each model is evaluated and best fitted with the building environment are selected. The evaluation requires specification of characteristics of a given building model. These characteristics are compared with the characteristics defined by the user as a building environment. A few building models which are best fitted so far are selected and copied, while the less fitted models are deleted. Finally the new instances of building models are modified. Evaluation, selection and modification repeats in a certain number of cycles resulting eventually in one best-fitted building model. The adaptation process is done.

Then, the system goes back to the iterative mechanism, involving the user. The user evaluates the generated best-fitted building model, deciding if the initial building environment (design objectives) were correctly specified.

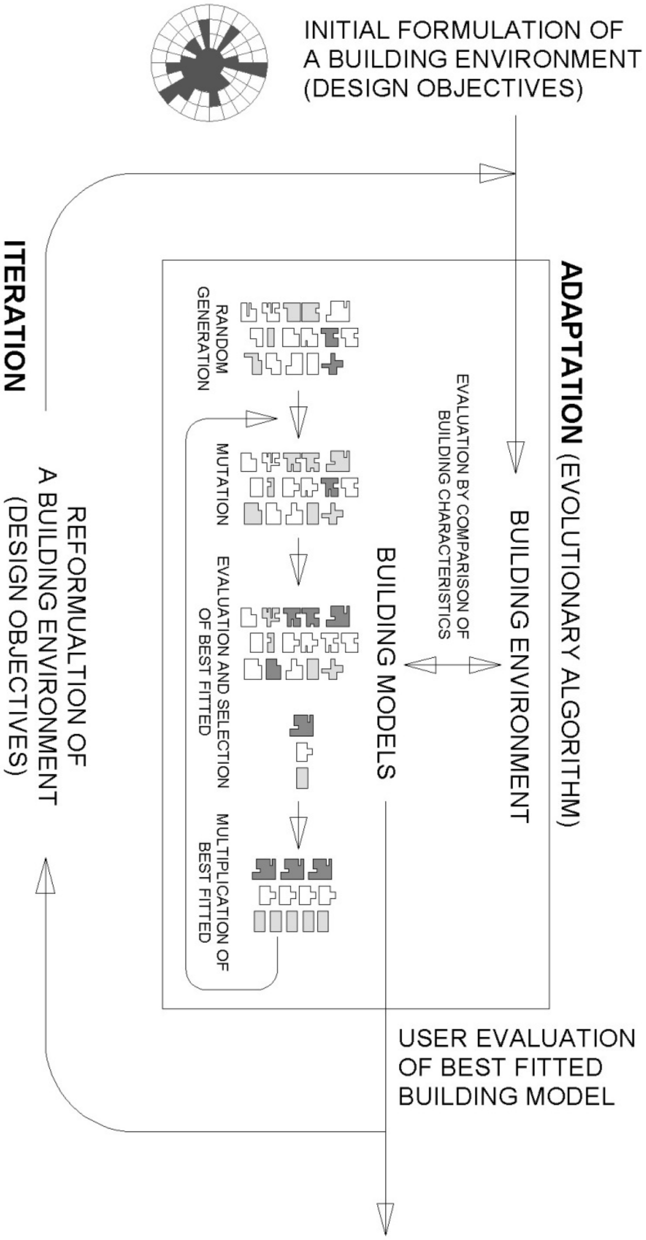


Figure 30. A synthetic scheme of the generative design system.

1.2. THE PRINCIPLES OF THE SYSTEM

In the following paragraphs, I list fifteen principles that were developed and argued for in the thesis. These principles should serve as a basis for realizing a functional generative design system.

The core element of the generative design system is the building environment. I believe, that appropriate implementation of this element is a decisive factor in constructing a viable design system. Thus, it is important to define the building environment in an inclusive and balanced way. I consider this principle central. Validity of this principle is supported by the analysis of four prototypes of generative design systems that I undertook in section IX.

PRINCIPLES RELATED TO DESIGN PROCESS

I derived two principles for the generative design system from the discussion about a design process.

1. A building model should be generated in a manner similar to *adaptation* – a gradual transformation of a building model towards increasing fitness with the building environment (design objectives). The way a building model is generated is derived from mechanisms of adaptation of complex systems. The generation of a building model is non-linear, involving combination of random search and ‘deterministic’ methods.
2. Beside adaptation, the generative process should involve *iteration*. By iteration I understand a cyclical reformulation of the building environment (design objectives) based on data resulting from the generated building model. Iteration stems from a ‘wicked’ nature of architectural problems, where a design problem (design objectives) is defined in a circular way, i.e. the solution affects the problem formulation. The functioning of the generative design system should be based on two mechanisms: adaptation and iteration. The adaptation should take place within the iteration.

PRINCIPLES RELATED TO THE BUILDING MODEL

3. A generic structure for building models should be flexible. The generative design system can represent only the building models that are anticipated in a predefined generic structure (it cannot represent all conceivable building models). The flexible generic structure for building models would increase their potential diversity. Implication

of the flexible generic structure would be a *large* search space for the evolutionary algorithm.

4. A building model should consist of elements which resemble real architectural components. That is, building elements should consist of geometric objects and *auxiliary information* about the kind of architectural component that is represented by the object. The information can include physical properties of architectural components, their function or how they relate to other components.
5. A building model should have a *hierarchic structure*. Some of its elements should have stronger interconnections than others and they should constitute groups. This would be beneficial in evolution of building models in the following way. The evolution involves mutation and recombination of building models. If selected building elements were strongly interconnected, making groups, then in the adaptation process there would be no need to decompose the adapting models to their basic elements. Because of this, the whole evolutionary process would be faster.

PRINCIPLES RELATED TO THE BUILDING ENVIRONMENT AND BUILDING CHARACTERISTICS

6. The building environment should be *inclusive* and *balanced*. Consequently, the generative design system should be able to consider a number of very different building characteristics, reflecting a typical ability of an architect in the actual design process. The building characteristics that the environment includes should be organized in terms of design *constraints* and *domains* they address. The design constraints include: site constraints, building codes, master plan regulations, client's intentions and architectural qualities. The design domains are drawn from the triple bottom line of sustainable architecture and they include: society, economy and ecology. Furthermore, the framework of sustainable architecture provides specific examples of building characteristics from each domain.
7. Instead of including one or a few building characteristics and simulating them to a high degree (*low resolution* and *high depth* of a building environment), it would be desirable to include more aspects and simulate them in less-details (high *resolution* and low *depth* of a building environment). Thus, specification of some complex building characteristics (such as for example energy consumption, ventilation or acoustic performance) should be implemented with an increasing level of accuracy, adjusted to the growing computational

capacity of machines. Alternatively, selected complex building characteristics could be omitted in the first implementations of the system and included later.

8. The building environment should offer a default parameter for each building characteristic it contains, in order not to necessitate a user to define a large number of building characteristic. The default set of parameters should be based on the framework of sustainable architecture.

PRINCIPLES RELATED TO THE EVOLUTIONARY ALGORITHM

9. The evolutionary algorithm should apply *mutation rate control.* It is more desirable to have a large mutation rate at the beginning of the generation process and then to focus on refining the solutions that have been found so far. Due to large mutation rates, a large part of the search space could be examined at the beginning of the search process, increasing probability of finding good solutions.
10. The algorithm should use *parameter adjustment*. The building model's development could be improved by application of a dynamic adjustment of such parameters as mutation rate control or probability of mutation and crossover. Practically, these parameters would be the subject of the evolutionary process. For example the level of decomposition of a building model can be a parameter that is adjusted dynamically. The decomposition can be deeper at the beginning of the adaptation and decrease gradually towards the end of the process.
11. The evolutionary algorithm should apply a *parallel-terraced scan*. This technique optimally allocates the available searching resources in order to find the best balance between exploration and exploitation of a search space. Thus, at the initial phase of the search (generation) process, building models should be very different from each other, exploring a possibly large space of design versions. Successively, the algorithm should focus on refining a few best-fitted building models. At the same time, the algorithm should allocate a fragment of the search resources on a random search for very different alternatives.
12. The algorithm should employ *penalty functions* – a method for eliminating deficient (for example spatially inconsistent) building models from evolving population.
13. The algorithm should use a *tournament selection*. It is a method for *relative* evaluation of building models. Instead of evaluating a building in absolute terms (for example by points) the algorithm

should indicate which model is 'better' in a direct comparison ('tournament') concerning selected building characteristic. For example the actual cost of a building does not have to be determined; instead, the algorithm can indicate which of two selected buildings would be *more* costly. Alternatively, the algorithm might use representational units for rating building models.

14. A *user intervention* in the process of building models evaluation should be allowed. The user should be able to monitor the adaptation process and intervene in defined periods to support the automatic evaluation of building models. This seems desirable because of the qualitative nature of some building characteristics (such as a building shape) which make automatic evaluation problematic.
15. *Multimodal problems*, *multi-objective problems* and the *Pareto front* are tactics from which the generative design system could probably benefit most. These tactics involve evolution that considers many different objectives (often opposing each other) at the same time. Design problems are a class of multi-objective problems, because they involve several design objectives (the building environment is inclusive). There are two main approaches to multi-objective problems. In the first approach, a multi-objective problem can be 'reduced' to a single-objective problem by a system of 'weights'. The generative design system would produce a single, best-fitted building model. In the second approach, a simulation of environmental niches can be applied, in which the building models evolve in parallel, adjusting to carefully selected subsets of design objectives. The generative design system would then offer a number of building models, which are best trade-offs between the subsets of design objectives (a Pareto front).

1.3. THE BENEFITS OF THE SYSTEM AND THE WAY IT CAN BE USED

It seems that nature, clients and architects can benefit from the generative design system proposed in the thesis.

Firstly, the selection of building characteristics that the system includes is based on the framework of sustainable architecture. Accordingly, the system suggests generation of building models that include the parameters that make the models environmentally and economically sustainable (for example, by default, a building model would aim at reduction of energy consumption), but also socially sustainable (the system would aim at diversity of form or accessibility of spaces).

Secondly, the client benefits, because the intensive exploration of a design situation that the generative design system offers, might reveal unanticipated development options. The system might be especially suitable for these types of buildings that require a substantial spatial analysis, like for example housing architecture. There, design objectives are often well-defined and the initial stage of design process involves extensive analysis that aim to find optimal correlation between a number of building characteristics. This analysis might include such building characteristics as: building's volume, number of floors, number and type of apartments, orientation of the apartments to sunlight and the like.

Lastly, the architect would benefit perhaps the most. The system aims at supporting architect in the first place, allowing for much more informed design practice. A role of the designer who uses the generative design system would depend on his or her approach to a design task. The system has three modi operandi. It can *solve* a design problem if it is well defined, *explore* a design problem, or generate building models that are sources of *inspiration*. These three modes are described more in-details in the next paragraphs.

PROBLEM SOLVING

In case, where the design intentions and the design requirements were rigid and well structured, the software would simply provide a solution for them. The user's role would be to input as much relevant data as possible in order to focus the generation process on the particular problem.

The design of a residential building could be an example. Often, the design objective is that the projected building contains a certain number of flats. If additionally the master plan regulations define the building's height, its area and its footprint, then the design task is largely constrained. The system's role would therefore be limited to finding the best distribution of flats (within the regulated building size). Additionally, the system could optimise the plan solution, improving for example the access to daylight, views and the like.

Such a ‘pure’ problem-solving situation is not very usual in architectural practice. Often, strictly formulated design requirements are either unfeasible, or they lead to unanticipated and undesired consequences. If they do so, then, the original problem formulation needs to be revised and some constraints have to be removed or adjusted. The search process becomes exploration.

EXPLORATION

According to Simon, the design process is an exploration of alternatives delineated by a set of constraints. To some degree, it is a process of ‘revealing’ the problem through design. It is worthwhile to quote here Alexander: “It is quite reasonable to think of the realisation [of a building model] as a way of probing the context’s nature, beyond the program but parallel to it” (Alexander, 1970: 91).

In the context of the generative design system, the desirable situation would be to keep the alternative solutions – they may both be attractive for some reason – rather than targeting at the apparently ‘best’ one. The software could serve as an exploratory tool, where a designer investigates a series of solutions. By differentiating input data and then studying output, the user can acquire information about the achievability of certain design goals or about the consequences of initially given requirements. The information acquired through such an ‘exploratory experiment’ gives more informed picture of the site. In other words, it gives an ‘insight’ into the site and it helps to ‘frame’ the problem: “Exploratory experiment is the probing, playful activity by which we get a feel for things. It succeeds when it leads to the discovery of something there” (Schön, 1982: 145). Based on the information obtained through generating and visualising alternative building models, the user can adjust the initial design requirements. The iteratively redefined input data would “stimulate the situation's back-talk [and make users] to appreciate things in the situation that go beyond their initial perceptions of the problem” (Schön, 1982: 148).

Exploration is a very common approach in architectural design. Architects often undertake design decisions only to see what consequences this may have. The situation is analogous to “(…) what an artist does when he juxtaposes colours to see what effect they make, and what a newcomer does when he wanders around a strange neighbourhood” (Schön, 1982: 145).

An example can be a situation when the user prioritizes a certain distribution of flats - for instance 140 flats, in which 25% is 2-rooms, 50% is 3-rooms, and 25% is 4-rooms. The generated solutions could show that to achieve 140 flats on a given site, one would need to build a very high building, which exceeds the cost of the whole project. In this case, the realisation of the initial intentions would turn out to be too expensive and thus unachievable, though one did not realize that at the beginning. Only the generation of a series of solutions would show the consequences of the initial intentions (a building that is too expensive). In the end, the original design intentions would need revision, and perhaps the number of flats or their sizes would have to be reduced. Optionally – operating in a dense urban environment – the user's interest might not be to maximise the number of flats, but to improve their usability in terms of access to daylight and good plan solutions. The priority of the system would then be to generate floor plans with sufficient daylight and to provide possibly comfortable plans.

Unlike a mere problem solving, exploration might lead to innovation. While problem solving "simply accepts the parameters of the problem given", exploration works by "a different, more experimental logic where, by rigorous analysis, design opportunities are discovered that can be exploited and transformed into design innovations." (Sharples et al., 2002: 6)

INSPIRATION

Finally, the user could use the system freely. For example, quasi-random forms may be generated; these unexpected forms would inspire the architect and provide him with new ideas. Here, the focus would be on the generative potential of the software rather than on design objectives. The building environment would serve as a 'pretext' for generating and exploring novel building models.

Today, many generative design systems concentrate on the generation of complex forms that are then freely interpreted by the architect (An example is prototype 3 - ref. section IX). These forms serve as a source of inspiration and have to be 'translated' into architectural language. Initially, the generated models do not have to explain or address the real architectural 'issues' as long as they inspire thinking and provoke discussion. Though the quotation below relates to scientific models, it illustrates very well building models:

> The main role of models is not so much to explain and to predict – though ultimately these are the main functions of science – as to polarize thinking and to pose sharp questions. Above all, they are fun to invent and to play with, and they have a peculiar life of their own. (Kac,1969, in: Weinberg, 2001: 43)

In this context, it is worthwhile to mention the concept of 'mis-use strategy'; it refers to the use of software in a different way than it was intended to be used. The use of the generative design system as a generator of free interpreted structures is a 'mis-use' strategy. It seems that the 'mis-use' strategy goes along with losing control over the design (Sevaldson, 2005: 346). In fact, purposefully decreasing control over the design is often considered as a prerequisite for creativity. 'Playing' with the initial parameters of the design system can provide a new extensive design material in the form of a collection of inspiring design proposals: "Losing control is a central aspect of a creative process (…) Strategizing the loss of control through digital generative design results in new kinds of generative source material" (Sevaldson, 2005: 349).

APPLICATION OF THE GENERATIVE DESIGN SYSTEM

The amount of data that the user introduces in the system delineates the number of solutions: the more data entered, the smaller the spectrum of solutions. By moving on the diagram below from the left hand side (much data entered) towards the right hand side (little entered data), one changes the system's mode of operation from the problem solution/ optimisation, through the design exploration to the inspiration (Figure 31).

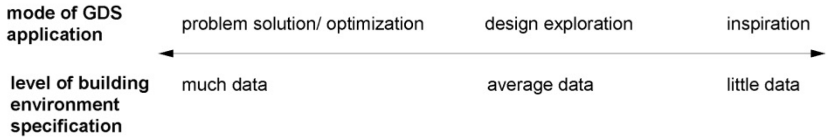


Figure 31. Three ways of applying the generative design system. In a well-specified building environment, the GDS works as a problem solver. In an undefined building environment, the GDS works as a generator of forms for inspiration. The intermediary position is the most typical: there is an average amount of data and the GDS can be used for design exploration.

ALLOCATING THE SYSTEM IN A DESIGN PROCESS

The generative design system is most useful in the early, conceptual stages of the design process. There are at least three arguments supporting this claim.

1. Referring to professional architecture practice, the two modes of using the system – exploration and inspiration – are present at the conceptual phase of the design process.
2. The design process in further phase becomes more technical and specific. For example, the particular technical design solutions depend on service provider and building contractor. All these activities go beyond the abilities of the proposed system, which focuses on generic solutions.
3. The speed of data processing is limited. The last stages of the design process demand operations on the detailed building models. Because the evolutionary algorithm performs a great number of calculations in order to determine a fitness function, and because the calculations are multiplied by a number of building models in a population and by a number of subsequent generations, the calculations for detailed building models would be extremely time-consuming.

1.4. CHALLENGES AND SHORTCOMMINGS OF THE SYSTEM

The most important challenge for the generative design system stems from the fact, that many of the building characteristics are qualitative – it is difficult to express them algorithmically and to implement them in a fitness function. The solution might be to involve the user in the evaluation process.

Moreover, evolutionary algorithms need much time to generate such complex objects as building models. The elaborated fitness function, where the complex building characteristics are calculated, could increase the time needed for building generation. A solution might be a reduction of the precision of building characteristics specification (reduction of the depth of the building environment). Then, the precision of the building characteristics specification can be increased gradually, considering the overall time of building model generation. Kolarevic and Malkavi (2005: 91) indicate a relevant approach to this problem in the following quotation: “Because simulation-based optimization can be time consuming, approximate-based methods which use functions derived from simulation responses to partially guide the search during the optimization process have been recently utilized.”[35]

As in the case of any architectural design software, the user interface is important. The generative design system is a complicated system, involving management of a large number of building characteristics. The interface that would allow a user to input design objectives easily and to monitor building model development in a comfortable way seems to be a significant challenge. Especially important seems to be the possibility to display the evolution process, to halt it, to visualise the key building models and their characteristics and to manipulate them. The difficulty of using the system should be not greater than that of commercial CAAD systems.

Some researchers criticise the use of AI (Artificial Intelligence) or particularly Evolutionary Computing in design. The main argument is that AI techniques function well only with well-defined problems, and that then they become a mere optimization technique (Sevaldson, 2005: 60). I think there are several approaches that can overcome this difficulty. One is to apply the iterative method to design generation, where redefinition of design objectives is a part of a process. In addition, the system should involve the user in the adaptation process, letting him or her control the qualitative building characteristics. Thus, a central mode of operation of this system would be a gradual and nonlinear approach to the solution rather than a straightforward problem solving.

Another counterargument emphasises EC’s inability to resolve potential design conflicts. Typically, the design process involves actors and experts representing a number of professions (structural engineers, heating/ ventilation/ air-conditioning engineers and the like). When a conflict between them arises, they discuss it and find a solution through negotiations, often bringing about a novel comprehension of the problem. The problem with AI applications is that since they are based on preconceived rules, it is not possible to solve the conflicts in a creative way.

The answer to this dilemma can be that as EC works as a black box, it is able to go beyond problem-specific knowledge and therefore beyond the

---

[35] See also: Choudhary, R. (2004).

preconceived rules. The multitude of perspectives (the fact that a building is described by characteristics of different categories) can be managed using EC techniques that address multimodal and multi-objective problems. One solution could be to put all the different categories under one framework and then control them by assigning priorities to each of them. The conflicting character of different design objectives was addressed in many part of this thesis, for example in section IV or in section VIII.

Unlike many applications using expert systems, in the generative design system proposed in this thesis, creativity arises from the concept of indirect design. Just as in the case of human intelligence, one cannot give a detailed account of how the black box (in general) or the human brain (in particular) solves the problem. However, adequately adjusted and indirectly controlled black boxes are capable to come up with smart and unexpected design solutions.

Another objection to the genetic algorithm is that it tends to produce repetitive forms and is unable to introduce a fundamental change: “[The forms generated] might be beautiful at first sight, [but] they all operate within narrow fields of formal types, which rapidly exhaust their potential of variation” (Sevaldson, 2005: 60). Here it seems that the problem lies in the flexibility of the building model structure, not in the nature of the genetic algorithm. The definition of the building model’s structure should be possibly flexible to provide diversification of generated models. In EC terms, the *solution space* should be possibly large (ref. section V and section VIII). Concluding from section IX, the fragmentariness is the common shortcoming of the generative design prototypes. In particular, the four prototypes discussed focus only on one or a few architectural aspects. This shortcoming seems to have its source in a narrow implementation of a building model, especially when compared to present BIM systems. The fact that these prototypes take into account only a narrow spectrum of building characteristics combined with a narrow implementation of the building model, results in the inability of these systems to introduce a fundamental change in the generated models.

This thesis defends the introduction of a more comprehensive approach, in which the building model structure would be more inclusive and in which a broader spectrum of building characteristics would be implemented in the system.

2. SUPPLEMENTARY CONSIDERATION

2.1. A BLACK BOX AND A WHITE BOX

In this section, I use the metaphor of a *black box* and a *white box* to summarize the most important mechanisms of the generative design system. Generally, a black box is a device that can be examined only through its input, output and so-called 'transfer characteristics' (which are defined through relations between the input and the output). In heuristic programming (neural networks, evolutionary computation, cellular automata, etc.), the term black box refers to algorithms, in which one cannot give a detailed account of how exactly the algorithm solves a problem. This is due to the fact that problem-specific or instance-specific knowledge is not implemented directly in the algorithm. Rather, the same construction of the algorithm is applied to a broad range of problems, and only selected parameters of the algorithm are adjusted to make it suitable to a specific problem. The direct inspection of a black box algorithm is difficult if not impossible, because the way the calculations are made does not resemble the way human beings solve problems in natural science or mathematics (i.e., through 'understanding' and 'conceptualising' a problem first and then solving it with equations).

In contrast, a white box is a device whose internal working can be examined. A white box's procedures are implemented explicitly, based on problem-specific knowledge. They are open to inspection, because their structure resembles the way the human mind conceptualises and solves problems.

According to the Weinberg's definition, a system is neither an organised simplicity (a machine) nor an unorganised complexity (an aggregate), but an organised complexity. While an organised simplicity (machine) works under a white-box principle, an organised complexity (system) works under a black-box principle. From this point of view, all systems are black boxes.

There are many examples of organised complexities: an ant colony, a neural system or a species subjected to natural evolution. It is difficult to give a detailed account of how an ant colony succeeds as a whole, manifesting a complex 'behaviour', even though such a complex behaviour is apparently based on a few simple rules. It is similarly difficult to give a *precise* account of how the human brain solves problems, though we know how neurons function[36]. It is the intricacy of the neural structure (relation between simple elements) that is responsible for producing the complex behaviour. Both the ant colony and the brain are the products of natural processes, and therefore to some degree their organisation is ‘alien’ to humans.

The reason for which black boxes are difficult to inspect is that they work as wholes. The functions in the black box are distributed over the whole black box mechanism. In some cases, a damaged neural network can continue to work after some time of re-training. It is because the network functions are localised both in the removed part and in the remaining part. The remaining part of the network takes over the functions of the removed part.

On the contrary, in a white box the functions can be easily localised in the structure. Therefore removing a certain part of the white box (e.g., removing a battery from a car) usually makes it inoperative.

Furthermore, it is possible to give a precise account of how a white box works. A change of one element in a white box structure changes its behaviour in a predictable way. For example, installing winter tires to a car would make driving on slippery surfaces safer. A change in behaviour is predictable, because the structure of the white boxes reflects the way human minds conceptualise and solve problems. Therefore the structure is pertinent to examination.

It is different when looking at the black boxes. In the case of the ant colony, one does not know what will be the impact of modifying one behavioural rule on the whole colony. Similarly, it is difficult to speculate (i.e. to anticipate without making actual experiments) what effect would have the modification of the way a neuron works on the whole brain. Neither can one say exactly what impact would a change in genetic mutation rate have on the process of natural evolution. Our observations and conclusions about black boxes are based mostly on experimenting, i.e., on manipulating the input and looking at the output. In the case of white boxes, the observations are based on the examination of the structure and its direct manipulation.

The black box and the white box mechanisms are approached either in terms of a system’s function or in terms of a system’s structure (Weinberg, 2001). The white box approach would focus on the system’s structure, while the black box approach would focus on the system’s function. According to

---

[36] Please note, that the case of human brain is simplified in the following paragraphs. My intention is to underline the fact, that to some extent it is difficult to inspect some parts of the human brain, because the structure of these parts is very complex (and thus resembles a black-box). I do not imply that the whole brain is a black-box and is not structured. Scientific evidence shows that the brain is structured and has localizable ‘functions’ (for example one area deal with vision, another with hearing, another with emotions, etc.) So in fact the human brain as a whole includes elements of a black-box and of a white-box.

Weinberg, the distinction appears under many names in the history of science:

> In biology we have the anatomists, who try to understand change through statics- behavior through being. On the other hand, we have the ethologists, who try to discover what is constant through what changes- being through behavior. At another level in biology we have the molecular biologists versus the systematists; in physics we have mechanics versus thermodynamics; in psychology, the physiologists versus the behaviorists; in art, the linear versus the painterly. (Weinberg, 2001: 252)

This distinction will be discussed more in the following paragraphs, in the context of the generative design system.

2.2. GENERATIVE DESIGN SYSTEM AS BLACK BOX

An evolutionary algorithm works as a black box and essentially it is not based on problem-specific knowledge. In principle, the algorithm consists in generating, modifying, and evaluating subsequent generations of individuals; and this over a number of cycles to find the best fitted individual. This principle can be applied to a broad spectrum of problems. However, some of the parameters of the algorithm need to be adjusted in order to be relevant to a particular task (e.g., generation of building models). Though the algorithm principle is clear, the way it transforms the particular building forms in the certain environment is complex. In this sense, the development of each particular building model is ‘hidden’ from the user and from the programmer (hence the term black box).

THE BLACK BOX ASPECT OF GENERATIVE DESIGN SYSTEM

The black box aspect is apparent when generating a building model. Neither the user nor the programmer has an overview of which parts of the building model exactly the algorithm transforms in order to attain the building model that fits the given design objectives. The programmer's knowledge about the general principles of the algorithm does not help in explaining the way a particular model evolves. The situation is similar to that of a brain scientist – though he or she might know the principles and general structure of a neural network, it does not help to explain the complexity, sophistication and ingenuity of every individual human thought. Actually, the fact that the building model's evolution is hidden is a crucial feature of the system. As a creative tool, the system's solutions should not be fully predictable (neither by the system's programmer nor by its user).

The role of the user is to enter design intentions and interpret the solution generated in the output. In some cases the generated model does not convince the user, who should therefore come back to the input and adjust the initial design intentions.

THE WHITE BOX ASPECT OF A GENERATIVE DESIGN SYSTEM

The problem-specific knowledge in the generative design system occurs on two levels: (1) the class to which the problem belongs (here the generation of a building model); (2) the specification of a particular building environment (design intentions and design requirements; see figure 32).

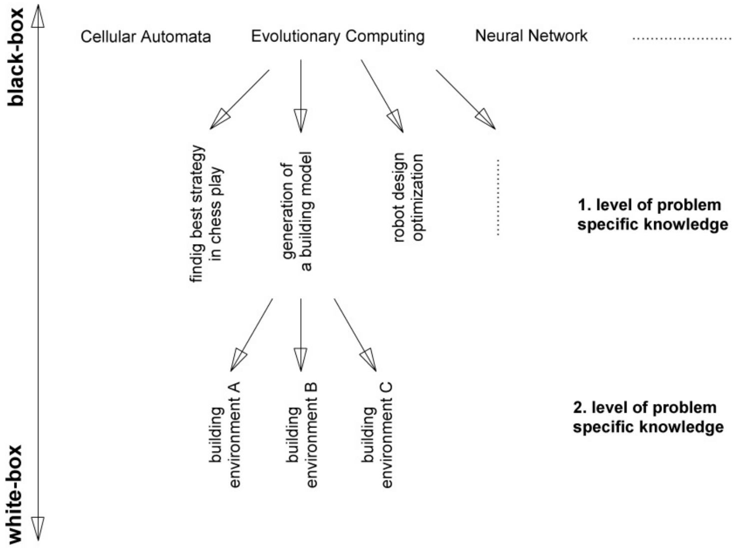


Figure 32. Two levels of problem-specific knowledge.

The first level of the problem-specific knowledge requires the definition of a general structure of a building model (phenotype), its encoding (genotype) and their relation (the growth function). The building model needs a definition of building elements and their relations, which is knowledge specific to the architectural domain. Moreover, the method of specifying building characteristics belongs to the first level of problem-specific knowledge. For example the calculation of energy consumption requires problem-specific equations, i.e., taken from thermodynamics. At this level, a programmer can inspect the calculation, because it resembles the structure of problem solving present in natural science. Because the programmer implements this information, the information can be later examined and revised by the programmer (e.g., to improve the calculations of energy consumption).

All parts of the algorithm related to the specification of a particular building environment are the second level of problem-specific knowledge. The definition of the problem becomes more specific here, as a concrete design situation is introduced in the algorithm. This knowledge is not implemented by a programmer, but by a user.

Figure 33 illustrates the efficiency of different methods of problem solving in terms of how they make use of problem-specific knowledge (Goldberg, 1989: 4). Problem-tailored methods are based on the knowledge adequate to a specific problem. They work well in a narrow area of unimodal problems, because the knowledge required for solving a specific unimodal problem is 'achievable'. Random and enumerative methods work equally inefficiently across a broad range of problems. Evolutionary Computing works well across a broad range of problems. Design problems – as a kind of multimodal problems - are marked with a dashed area on the right hand side. As the graph shows, Evolutionary Computing seems most effective for this kind of problems.

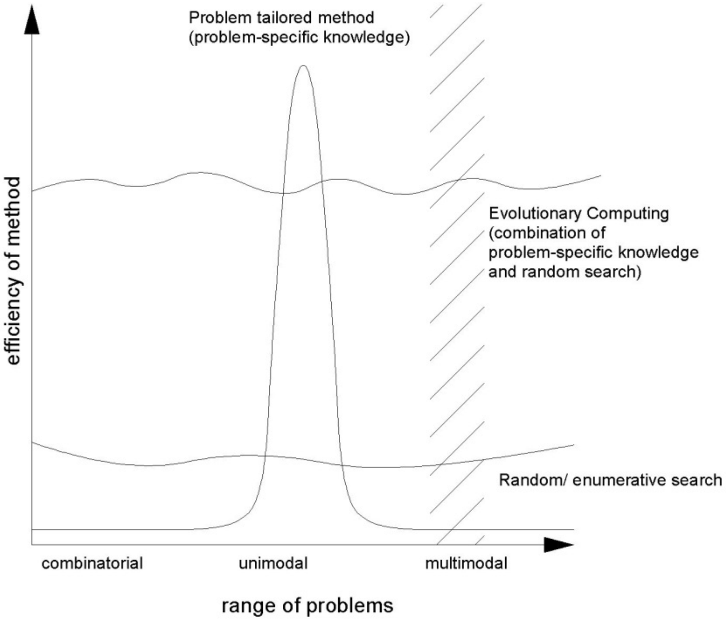


Figure 33. Problem-tailored methods work well in a narrow problem domain. Random and enumerative methods work inefficiently across a broad range of problems. Evolutionary Computing works well across a broad range of problems, including design problems (dashed area).

2.3. THE INDIRECT DESIGN

The black box approach to design is very similar to the behavioural concept of performance. The latter approach assumes that products or devices are measured and controlled through their behavioural (performative) specifications, regardless its physical structure (Williamson et al., 2003: 74). The approach can be compared to the way nature proceeds: in the natural evolution process, the organism's structural characteristics are of secondary importance as long as it performs well in a given environment, i.e. as long as it has suitable behavioural features that let him exist in this environment. Indeed, one can observe many structurally different organisms occupying one environment.

Applying this approach to the generative design system, the building model's behaviour (performance) finds its analogue in the complex building characteristics, such as for example energy consumption level, and the building model's physical structure finds its analogue in the configuration of the building elements. Furthermore, the generative design system evolves a building model's structure in a way that makes it fitted to the required behaviour (performance), defined by a user. So, the actual behaviour of a generated building model adapts towards the required behaviour.

The behaviour of a building model is defined by its structure. Specifically, a complex characteristic (behaviour) of a living organism is a consequence of its basic characteristics, which in turn are the products of its structure. The fact that birds can fly (a complex characteristic of birds) is – amongst others – a consequence of their low weight (a basic characteristic), which in turn is a product of the fact that they have wings, pneumatic bones and other similar structural adaptation features. The anatomists would focus on a bird's structure (wings and pneumatic bones) to understand its behaviour, while ethologists would look at a bird's behaviour (making nests on trees) to understand its structure.

The structure of an organism is less important to the ethologist, who focuses on behaviour. By analogy, the structure of a building model (i.e., the configuration of building elements) is of a secondary importance to the design system, which can accept any building structure as long as the building satisfies a set of behaviour criteria. In the design system, as "in the black-box view, adaptation is seen through (…) changes in behavior" (Weinberg, 2001: 252).

In this context, the black box approach is almost identical with the performance concept. In both approaches the transformation of a system's structure occurs through controlling system's behaviour rather than the direct inspection of its structure. When it comes to a building model, instead of operating directly on its elements (as it is usually done in classical CAAD systems) the user's role would be to define a set of desired building characteristics (desired building performance). The design system would therefore produce a solution that satisfies the required characteristics.

The approach presented above shifts from a traditional way of direct design, where a user draws and manipulates each individual building element, to an *indirect design*, where a user operates on building characteristics (building performance). Because the building characteristics express complex dependencies or relations amongst the building elements, operating on them must affect indirectly the structure of the building model. For example, an energy consumption characteristic is determined – among many others – by the relations of opaque walls and glazed surfaces. Given this, changing a 'parameter' of energy consumption should effect in reconfiguration of the opaque walls and glazed surfaces.

All the designs in which generation is automated, (i.e., not defined by the user) might seem to involve the indirect design. Especially it might seem that all the parametric designs in which the user makes objects by specifying their general parameters belong to the indirect design. A parametric design of a staircase is an example; the user specifies the number and size of steps, the height of a floor and the like, and the staircase is generated automatically. However, there is a difference between such a parametric design and the generative design system proposed here. In fact, the majority of parametric design systems follow a *white box* principle, as they apply problem-specific algorithms. Using this kind of algorithms makes the generation of a staircase predictable and limited to what the programmer had implemented. On the contrary, the evolutionary algorithm applied in the generative design system is only partly based on problem-specific knowledge, and the solution it generates can go beyond the boundaries of the specific problem definition.

Furthermore, the indirect design approach to architecture proposed here is very close to what is called a 'network thinking', where "focusing on relationships between entities rather than the entities themselves" is essential (Mitchell, 2009: 233). Building characteristics 'bond' building elements into relations and in this sense, operating on building characteristics means operating on relations. A given building characteristic can be realised with many configurations of the building elements. For example, there are many different buildings in which the average energy consumption is 170 kWh/m^2, or there are many buildings in which the living room is connected to the kitchen. On the other hand, a given form of a building can be described by many different characteristics. For example, a particular house (building model A) can consume 170 kWh/m^2 and have a living room connected to the kitchen, etc.

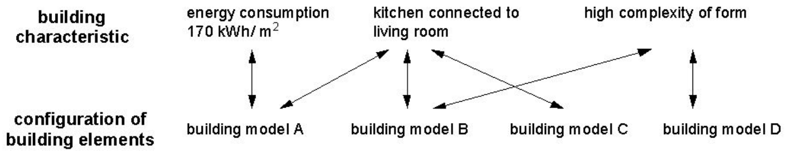


Figure 34. A network of connections between building characteristics (performance) and configuration of building elements (building's structure).

From this point of view, building structures and building characteristics constitute a complex network of dependencies. The implementation of the generative design system proposed here would require the definition of these connections, that is, the definition of the building characteristics.

3. CONTRIBUTION TO THE EXISTING BODY OF KNOWLEDGE

This thesis has two major outcomes. The first one is the development of the generative design system. The thesis proposes and develops constitutive *elements* of the system and the *principles* for its functioning. The synthetic scheme discussed at the beginning of this section integrates the elements and the principles and provides a graphical summary of my contribution. Later in this section I present a concept of *indirect design*, in which the generative design system generates building models by managing their characteristics. Originality of this concept consists in its holistic approach to design – it conceptualizes a building model as a complex system in which a number of characteristics should be considered simultaneously. The concept of indirect design implies that the generative design system should essentially work as a black-box.

Secondly, the thesis contributes to the theoretical debate on a generative approach to design. My original work is contained in the discussions that go along with the development of the principles. I use quotations from general domains of systems science, design theory and Evolutionary Computing, in order to apply them to the *specific* context of generative design. Theoretical discussion is supplemented by an analysis of existing prototypes of generative design systems. I believe, this leads to new conclusions and new understandings of the generative process.

DEVELOPMENT OF A GENERATIVE DESIGN SYSTEM

The main contribution of my thesis is a development of a comprehensive generative design system. This generative design system consists of four elements: building model, building characteristics, building environment and evolutionary algorithm. Its functioning is based on fifteen principles. I propose and develop these elements and principles throughout the thesis.

The essential element of my contribution is the characteristic of the building environment that I devise in section VII. My argument is that the building environment has to be both inclusive and balanced. Only then, the generative design system can generate workable building models. Consequently, I argue for the integrative approach to building model generation. I acknowledge the fact, that different building characteristics are integrated in one building model and this integration should be preserved in a generation process. Based on this assumption, I argue for a ‘high-resolution’ (including a high number of building characteristics) and ‘shallow’ (approximate specification of complex building characteristics) building environment rather that ‘low-resolution’ and ‘deep’.

I validate my argument in section IX, where I make a critical review of the prototypes of generative design systems. I recognize that building environments of these prototypes are focused on a few selected building characteristics: they are low-resolution and deep. In my opinion this is the limitation that seriously decreases usability of these systems in the actual design practice, where a number of diverse building characteristics has to be considered.

Several principles for the generative design system are developed on the basis of the inclusive building environment. For example, the application of multi-objective problems is especially suitable to the environment that includes many inter-related criteria. I show how the selected techniques of the multi-objective problems, such as niches or Pareto front can be applied to the comprehensive building environment.

Finally, the way a user would control the building model development is innovative. I called it *indirect design*, because the control over the design is obtained through manipulating building characteristics, not the building elements (as in conventional CAADs). Nowadays, the design process consists in first arranging building elements, and then deriving building characteristics (building volume, area of selected rooms, costs etc.) The indirect design is an inverse way – the definition of building characteristics affects the arrangement of building elements. This topic was discussed in the context of a black box and a white box earlier in this section.

It is worthwhile here to point at the difference between the concept of indirect design and the approach referred to as a 'performance based design'. The difference consists in the totality of the approach that I suggest. The 'performance based design' shares the common feature with the prototypes that I reviewed – it approaches designs in a fragmented way, as it usually focuses on a few building characteristics, which are examined very extensively, disregarding a number of others. The approach that I suggest focuses on the *integration* of the multiple and diverse building characteristics and on the examination of *interplay* between them.

THE DISCUSSION ACCOMPANING THE DEVELOPMENT OF ELEMENTS AND PRINCIPLES

The starting point of this thesis was a theoretic problem of how the complexity and the diversity of design aspects can be unified under one coherent theoretical framework. The development of the generative design system in many places of the thesis is accompanied by the extensive references both to the common design practice and to the complexity theory. Hence, the second purpose for developing the design system was finding new relationships between complexity theory and generative approach to design.

Specifically, many of the principles that are developed here primarily for the generative design system can be applied to a common design practice, only if a digital building model were replaced by a *mental* building model. Assuming that a mental building model is an adapting system, the design practice should combine random search with 'deterministic' methods. The design in this light would be an incremental search process for a satisfactory (never optimal) building model. Strategies like parallel-terraced scan can be successfully applied and further developed in conceptual design phases. At the beginning of the design process one should work with possibly different versions of design, recognizing the most promising solutions. Then – according to the available resources – one should select one or two that are so-far best and allocate the resources to refine these models. At the same time, one should ‘in parallel’ look at the alternative building models.

4. SUGGESTIONS FOR FURTHER RESEARCH

The further research should involve a realisation of the generative design system. It should start with constructing a mathematical representation of elements of a building model, their properties and their relations. Then, a list of building characteristics should be made. Implementation of the characteristics should start with basic ones. The building environment should include possibly different design aspects (according to the guidelines proposed in section about the building environment). Then, methods of deriving these characteristics from configuration of building elements should be implemented specifically for each characteristic. The list could be gradually developed and adjusted according to the working of the design system. The inclusive structure of the generative design system allows for its improvement. The building environment could be easily enhanced with new, more complex building characteristics. Alternatively, definitions of the characteristics that are already included could be enhanced.

Next step would consist in the implementation of particular techniques of EC mentioned in the thesis. Emphasis should be put on techniques addressing multi-objective problems. A comfortable management and inspection of ‘Pareto front’ of generated building model, should be a priority.

Designing a plausible user interface in details should be another important step in further system’s development. A visual working environment is a key element in a practical application of the generative design system. A user should be able to input and read information in:

- a numerical form (e.g. energy consumption levels, areas of rooms, volumes of rooms);
- a form of schemes and maps (e.g. rooms adjacency) and
- 2D and 3D illustrations (e.g. shape of a building site, shape of the surroundings, contour of an intended building, volumes of the immediate neighbourhood, volume of the intended building).

Further research could gradually go deeper into study of selected biological mechanisms, their generalization (area of complexity sciences) and application to the generative design system, which would improve functioning of the system. For example, a *growth function* was only mentioned in my thesis, but it is especially worthwhile area of further research. The growth of a building model consists in encoding its genetic code into an actual digital building model. This is a process analogical to a biological ontogenesis (a growth of an organism from the fertilized egg to the mature form). A more detailed theoretical study of self-organizational mechanisms of biological systems would be helpful. Specifically, one should investigate more in-details how a building environment can affect the process of building model's 'ontogenesis'.

In addition, adaptation of biological systems occurs not only during their evolutionary development (phylogenesis) and when they grow from fertilized egg to a grown-up organism (ontogenesis) but also during their adult life (homeostasis). It seems worthwhile to study the adaptation of a *grown-up* organism – a *physical* building, not only a building model. The emphasis could be on how a physical building adapt to changing physical environment (seasonal change of temperature and sun exposition, changing building surroundings, change functional need, etc.) and how a generative method of design can positively affect a further 'life' of physical buildings. In this context a theme of over-fitness can be discussed more in-details.

Bibliography

Ashby, W. R. (1960), *Design for a Brain*. New York, NY: Chapman and Hall.

Alexander, C. (1970). *Notes on the Synthesis of Form.* Cambridge, MA: Harvard University Press.

Alexander, C., Ishikawa, S., Silverstein, M. (1968). *A Pattern Language which Generates Multi-Service Centers.* Berkeley, CA: Center for Environmental Structure.

Alexander, C., Ishikawa, S., Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction.* New York, NY: Oxford University Press.

Alexander, C. (2004). *The Nature of Order.* Berkeley, CA: The Centre of Environmental Studies.

Asbjornsen, O., Keitsch, M. "System Thinking Revisited." (digital copy)

Bar-Cohen, Y. (Ed.) (2010). *Biomimetics. Nature-Based Innovation.* Boca Raton, FL: CRC Press.

Bentley, P. (Ed.) (1999). *Evolutionary Design by Computers*. San Francisco, CA: Morgan Kaufmann.

Bertalanffy, L. von (1968). *General System Theory: Foundations, Development, Applications.* New York, NY: George Braziller.

Brawne, M. (1992). *From Idea to Building: Issues in Architecture.* Oxford: Butterworth Architecture.

Buelow, P. von (2007). *Genetically Engineered Architecture. Design Exploration with Evolutionary Computation.* Saarbrucken: VDM Verlag Dr. Muller.

Cabrera, D. (2006). "Boundary Critique: A Minimal Concept Theory of Systems Thinking." ISSS research paper, http://www.isssbrasil.usp.br/isssbrasil/pdfs/2006-403.pdf

Caldas, L. and Norfold, L. (2001). "Architectural Constraints in a Generative Design System: Interpreting Energy Consumption Levels." 7th International IBPSA conference. Rio de Janeiro, Brazil.

Caldas, L. (2003). "Shape Generation Using Pareto Genetic Algorithms." CAADRIA conference. Bangkok, Thailand.

Caldas, L. (2005). "Three-Dimensional Shape Generation of Low-Energy Architectural Solutions using Pareto Genetic Algorithms." 23rd eCAADe conference. Lisbon, Portugal.

Chalmers, A. F. (1999). *What is This Thing Called Science?* Buckingham: Open University Press.

Choudhary, R. (2004) *A Hierarchical Optimization Framework for Simulation-Based Architectural Design.* University of Michigan. PhD thesis.

Chen, L. (2007). *Constructing Structural Shape Grammars.* CAADRIA conference. Nanjing: China.

Chen, S. Y. and Chiu, M. L. (2006). “Designing Smart Skins for Adaptive Environments: A Fuzzy Logic Approach to Smart House Design.” Computer-Aided Design & Applications 4(6): 751-760.

Corne, D. W. and Bentley, P. (Ed.) (2001). *Creative Evolutionary Systems.* San Francisco, CA: Morgan Kaufmann.

Coyne, R. (Ed.) (1990). *Knowledge-Based Design Systems* (The Teknowledge Series in Knowledge Engineering). CA: Addison-Wesley Pub.

Coyne, R. D. and Newton, S. (1989). “A Tutorial on Neural Networks and Expert Systems for Design.” Proceedings of the Australasian conference on expert systems in engineering, architecture and construction. University of Sydney. 321-337.

Creswell, J. (2003). *Research Design: Qualitative, Quantitative, and Mixes Methods Approaches.* Thousand Oaks, CA: Sage Publications.

Darwin, Ch. (2003) [1859]. *The Origin of Species.* New York, NY: Signet Classics.

Day, A. (1997). *Digital Building.* Oxford: Taylor & Francis Group.

Duarte, J. P. (2001). *Customizing Mass Housing: A Discursive Grammar for Siza’s Malagueira Houses*. Department of Architecture, Massachusetts Institute of Technology. PhD thesis.

Dounas, T. and Anastasios, M. K. (2006). “Generation of Alternative Designs in Architectural Problems Using Shape Grammars Defined With Animation Tools - A Computer Application of Shape Grammars Using Modelling and Animation Software”. eCAADe conference. Volos, Greece.

Dunin-Woyseth, H. and Michl, J. (Eds.) (2001). *Towards a Disciplinary Identity of the Making Professions. The Oslo Millennium Reader vol.4.* Oslo School of Architecture and Design.

Dunin-Woyseth, H. and Nielsen, L. M. (Eds.) (2004). *Discussing Transdisciplinarity: Making Professions and the New Mode of Knowledge Production. The Nordic reader vol. 6.* Oslo School of Architecture and Design.

Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Berlin: Springer.

Flood, I. and Christophilos, P. (1996). “Modelling Construction Processes Using Artificial Neural Networks.” *Automation in Construction* 4(4): 307-320.

Frazer, J. (1995). *An Evolutionary Architecture.* London: Architectural Association.

Galison, P. (1990). “Aufbau/ Bauhaus: Logical Positivism and Architectural Modernism.” *Critical Inquiry* 16: 709-52.

Gell-Mann, M. (1995). *The Quark and the Jaguar: Adventures in the Simple and the Complex.* London: Abacus.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley.

Groat, L. and Wang, D. (2002). *Architectural Research Methods.* New York, NY: Wiley.

Hak, T., Moldan, B. (Ed.) (2007). *Sustainability Indicators.* London: Island Press.

Hensel, M. (2013). *Performance Oriented Architecture.* Chichester: Wiley.

Hensel, M., Menges, A. and Weinstock, M. (2010). *Emergent Technologies and Design. Towards a Biological Paradigm for Architecture.* New York: Routlege.

Hofstadter, D. R. (2000). *Gödel, Escher, Bach.* London: Penguin.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.

Huang, Y. H. (2000). "Investigating the Cognitive Behavior of Generating Idea Sketches. Neural Network Simulation." CAADRIA conference. Singapore.

Jagielski, R. and Gero, J. S. (1997). "A Genetic Programming Approach to the Space Layout Planning Problem." CAAD Futures conference. Munchen, Germany.

Jones, J. C. (1992). *Design Methods.* New York, NY: Van Nostrand Reinhold.

Kac, M. (1969). "Some Mathematical Models in Science." Science: 16.

Kolarevic, B. (Ed.) (2003). *Architecture in the Digital Age: Design and Manufacturing.* New York, NY: Taylor & Francis.

Kolarevic, B. and Malkavi, A. M. (Ed.) (2005). *Performative Architecture: Beyond Instrumentality.* New York, NY: Taylor & Francis.

Kuhn, T. S. (1996). *The Structure of Scientific Revolutions.* Chicago, IL: University of Chicago Press.

Laszlo, E. (1996). *The Systems View of the World: A Holistic Vision for Our Time.* Cresskill, NJ: Hampton Press.

Lee, J. H. (2003). "Housing Types and Classification Systems in Case-Based Design." ANZAScA Conference. Sydney: Australia.

Lloyd, S. (2001). "Measures of Complexity. A non-exhaustive list." Institute of Electrical and Electronics Engineers Control Systems Magazine. Vol. 21.

Malkavi, A. M. and Augenbore G. (2003). *Advanced Building Simulation.* New York, NY: Spon Press (Taylor & Francis).

Malkavi, A. M., Srinivasan, R. S. et al. (2003). "Performance-Based Design Evolution: The Use of Genetic Algorithms and CFD." IBPSA conference. Eindhoven, Netherlands.

Maturana, H. and Varela, F. (1980). *Autopoiesis and Cognition: the Realisation of the Living.* Dordecht: D. Reidel Publishing Company.

McShea, D. W. (2001). "The Hierarchical Structure of Organisms: A Scale and Documentation of a Trend in the Maximum." *Paleobiology* 27 (2). 405-423.

Mei, C., McKay, A. et al. (2006). "Combining Evolutionary Algorithms and Shape Grammars to Generate Branded Product Design." *Design Computing and Cognition.* Dordrecht, Netherlands: Springer. 521-539.

Menges, A., Ahlquist, S. (Ed.) (2011). *Computational Design Thinking.* (AD Reader). Chichester: Wiley.

Mitchell, M. (2009). *Complexity: A Guided Tour.* Oxford: Oxford University Press.

Mitchell, W. J. (1990). *The Logic of Architecture: Design, Computation, and Cognition.* Cambridge, MA: MIT Press.

Nelson, H. G., Stolterman, E. (2003). *The Design Way. Intentional Change in an Unpredictable World.* Englewood Cliffs, NJ: Educational Technology Publications.

O'Neill, M. J. (1992). "Neural Network Simulation as a Computer-Aided Design Tool for Predicting Wayfinding Performance." New York, NY: John Wiley & Sons. 347-366.

O'Reilly, U. M. and Ramachandran, G. (1999) "Evolution as a Design Strategy for Nonlinear Architecture: Generative Modelling of 3-D surfaces." doi=10.1.1.56.840

Osborne, H. (1979). *Abstraction and Artifice in Twentieth-Century Art*. Oxford: Clarendon Press.

Oxman, R. E. (1988). *Expert System for Generation and Evaluation in Architectural Design.* Faculty of Architecture and Town Planning, Haifa. PhD thesis.

Polkinghorne, D. E. (2006). "An Agenda for the Second Generation of Qualitative Studies." *International Journal of Qualitative Studies on Health and Well-being*, 1(2). 68-77.

Popper, K. R. (1963). *Conjectures and Refutations: The Growth of Scientific Knowledge.* London: Routledge and Kegan Paul.

Rittel, H. W. J. and Webber, M. M. (1973). "Dilemmas in a General Theory of Planning." *Policy Sciences* 4 (2). 155-169.

Rosenberg, A. and McShea, D. W. (2008). *Philosophy of Biology: A Contemporary Introduction.* New York, NY: Routledge.

Salingaros, N. (1995), "The Laws of Architecture from a Physicist's Perspective." http://www.math.utsa.edu/ftp/salingar.old/Laws.html

Salingaros, N. (1997). "Life and Complexity in Architecture from a Thermodynamic Analogy." *Physics Essays*. Vol. 10. 165-173.

Salingaros, N. (2006). *Theory of Architecture.* Solingen, Germany: Umbau-Verlag.

Schodek, D. L. (2005). *Digital Design and Manufacturing: CAD/CAM applications in architecture and design.* Hoboken, NJ: Wiley.

Schön, D. A. (1983). *The reflective practitioner: how professionals think in action.* New York, NY: Basic Books.

Sevaldson, B. (2005). *Developing Digital Design Techniques. Investigations on Creative Design Computing.* Oslo School of Architecture and Design. PhD thesis.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press.

Sharples, Ch., Holden, K., Pasquarelli, G. (Ed.) (2002). *Versioning: Evolutionary Techniques in Architecture.* (AD). Chichester: Wiley-Academy.

Simon, H. A. (1957). “A Behavioral Model of Rational Choice.” *Models of Man, Social and Rational. Mathematical Essays on Rational Human Behavior in a Social Setting*. New York, NY: Wiley. 241-260

Simon, H. A. (1962). “The Architecture of Complexity.” Proceedings American Philosophical Society. 106. http://www.cs.brandeis.edu/~cs146a/handouts/papers/simon-complexity.pdf

Simon, H. A. (1981). *The Sciences of the Artificial.* Cambridge, MA: MIT Press.

Simon, H. A. (1990). *The Architecture of Complexity.* Philadelphia, PA: The American Philosophical Society.

Sirowy, B. (2010). *Phenomenological Concepts in Architecture*. *Towards a User-Oriented Practice. Oslo:* The Oslo School of Architecture and Design. PhD thesis.

Stake, R. E. (1995). *The Art of Case StudyRresearch.* Thousand Oaks, CA: Sage.

Stake, R. E. (2006). *Multiple Case Study Analysis.* New York, NY: The Guilford Press.

Sterman, J. D. (2000). *Business Dynamics.* Boston, MA: Irwin McGraw-Hill.

Stiny, G. and Gips, J. (1971). “Shape Grammars and the Generative Specification of Painting and Sculpture.” IFIP Congress 71. Ljubliana, Yugoslavia. http://shapegrammar.org/ifip/SGBestPapers72.pdf

Terzidis, K. (2006). *Algorithmic Architecture.* Amsterdam: Elsevier.

Terzidis, K. and Narahara, T. (2006). “Multiple-Constraint Genetic Algorithms in Housing Design.” Annual Conference of the Association for Computer-Aided Design in Architecture. Louisville, KY.

Thompson, D. A. and Bonner, J. T. (1966). *On Growth and Form.* Cambridge: Cambridge University Press.

Voigt, H.-M., Ebeling, W. (Ed.) (1996). *Parallel Problem Solving from Nature.* Conference proceedings*.* Berlin, Heidelberg: Springer-Verlag.

Vries, B. de, et al. (2004). ”Dutch Dwellings”. 7th International Conference on Generative Art. Milan, Italy.

Weinberg, G. M. (2001). *An Introduction to General Systems Thinking*. New York, NY: Dorset House.

Weinstock, M. (2010). *The Architecture of Emergence: the Evolution of form in nature and civilisation.* Chichester: Wiley.

Williamson, T., et al. (2003). *Understanding Sustainable Architecture.* London: Spon Press.

Woodward, C. and J. Howes (1997). *Computing in Architectural Practice*. London: Spon Press.

Yin, R. K. (2009). *Case Study Research: Design and Methods.* Los Angeles, CA: Sage.

APPENDIX 1 - AN INTERDISCIPLINARY DISCUSSION OF THIS PROJECT

The mains findings of this research were presented at a seminar at the laboratories of the Computer Science Department at the University of Oslo in March 2010. In this appendix I briefly summarise the feedback I have received, focusing on interdisciplinary communication.

Seven professionals participated in the seminar (software engineers and computer scientists), which was chaired by Professor Jim Tørresen. The presentation of the research took about 45 minutes and was followed by a 45 minute session for comments, questions and discussion.

As the participants were all specialists, it was not necessary to present the mechanisms of Evolutionary Computing. Yet, the same mechanism discussed earlier with architects and designers sounded unclear to them. The seminar confirmed that when it comes to programming techniques, what seems elementary for computer scientists is rather abstract for designers.

The computer scientists were particularly interested in the *realisation* of a generative design system, which was part of the presentation (ref. section IX). The discussion of the design system focused on the interplay between two conflicting *building characteristics*: energy consumption and daylight illumination inside a building. This interplay was processed by the genetic algorithm. The participants understood the nuances of the problem immediately and they came up with relevant comments. For example, they underlined that the distribution of sunlight and thus the energy consumption and the illumination levels depend on the geographic location of a building and a layout of windows. Besides, they were interested in diagrams showing the results of the genetic algorithm and in the 3D visualisations of the generated buildings.

One of the participants was very specific about a possible realisation of the system and he tried to propose to formulate the discussed aspects directly to a computable code. He was interested in a novel way of encoding building elements such as walls, windows, etc.

Even though the participants could readily understand the essential mechanism of the generative design system, the nature and complexity of architectural problems seemed ‘alien’ to them. They were unfamiliar with the requirements of the site analysis, constraints of the terrain shape, building codes, master plan regulations, etc. Moreover, they were not totally convinced by the importance of an inclusive approach – the need for considering a broad spectrum of design aspects which may differ in nature.

They tended to emphasise the elaborated forms of a building as if it were the most important aspect of architecture.

The complexity of design problems (and thus the complexity of their application in a generative design system) was the central point of the discussion we had. They pointed at a potential difficulty in defining a fitness function, because such a function would embrace so many different building characteristics. They suggested using a technique called multi-objective optimisation, where an algorithm deals with two or more conflicting objectives (ref. section VIII).

Another problem raised was the quantification of some of the building characteristics. Especially, the participants acknowledged that measuring the aesthetic properties of a building is difficult. Some participants claimed that it is not possible to formulate the visual aspects of a building in an algorithmic way. It seems to me, however that some of these aspects might be analysed mathematically (provided that they are approached as variables such as symmetry, proportions, optical balance, contrast, the centre of a composition, etc.).

Surprisingly, the present scientists emphasised the importance of an aesthetic approach in design, showing examples of generative systems that produce very complex, mathematical forms. There is an area of research that applies evolutionary computation to the generation of very complex, visually attractive forms. The generation of such forms indeed requires computers and it could not be achieved otherwise. Nonetheless, the originality of the approach presented in this thesis, is that it focuses on complexity of the problems present in the architectural *practice*. This thesis advocates a balanced architecture that is based on sustainable development guidelines, and in which the social, economic and ecologic aspects of the design ideally play an equal role.

APPENDIX 2 - DESIGN PROCESS AS A VERIFICAITION OF DESIGN HYPOTHESIS

The adaptive and iterative character of a design process can be approached from yet another angle. In this appendix I look at the design process from the viewpoint of the 'design hypothesis verification' proposed by Brawne (1992). The author attempts to rationally account for what a design process is, applying a modified version of Popper's theory of empirical falsification.

> Karl Popper puts forward a schema for the growth of theories which summarizes the view he has elaborated over a number of years and to which he returns again and again. It takes the following form: P_1 -> TT -> EE -> P_2 (Brawne, 1992: 58-59)

In this scheme, P1 stands for problem, TT for tentative theory, EE for error elimination and P2 for problem redefinition. In natural sciences, a problem is defined by a set of observable data - for example the positions of the planets in the solar system. A tentative theory is then proposed; it 'captures' the data in a simpler form of equations. In other words, the theory captures in a mathematical form the recognized patterns in the observed data. Instead of having a great number of records of subsequent planet positions, the relevant theory uses equations, which give a position of a given planet at a given time. The theory is never quite certain, i.e., one is never sure whether new observations will fit the theory or on the contrary refute it. Therefore the theory is a tentative one, a hypothesis, which is repeatedly tested against the new observations. Further, the more tests the theory passed, the better it is:

> Popper introduced the term 'corroboration' (…) 'for the degree to which a theory has been severely tested'. Moreover, because the testing of theories is an iterative process, raising new problems as we go along, 'the degree of corroboration of a theory has always a temporal index: it is the degree to which the theory appears well tested at the time t. (Brawne, 1992: 61)

The new, more precise observations might show the shortcomings of the theory, as well as point to new, unanticipated problems. Then, a process of error elimination begins, where the theory is being adjusted or replaced by a theory that better explains the data.

According to Kuhn, scientific theories are built in a framework of more general assumptions – scientific paradigms. Kuhn defines the scientific paradigm as a “universally recognized scientific achievements that, for a time, provide model problems and solutions to a community of researchers” (Kuhn, 1962: x). Especially, the paradigm specifies what is to be observed and how the observations should be measured and interpreted. In other words, the paradigm ‘frames’ the observations. So, in fact, the tentative theory is not tested against just observations, but it is tested against the observations structured or framed by a current paradigm.

This mechanism can be compared to the iterative character of a design process. A building model is a tentative theory – a proposed solution for a given design problem. As a tentative theory attempts to ‘capture’ the observations in a simpler, more elegant form of mathematical equations, the building model attempts to solve a design problems formulated as design intentions and design requirements. And as a quality of scientific theory is corroborated by a number of tests or scientific experiments, a building model and its components are tested throughout the design process by what Schön calls ‘mental experiments’ or by other kinds of simulations. While the scientific theory is evaluated against the observable data, the building model is evaluated in terms of how well it fits a building environment. The analogy goes even further: just as each ‘paradigm shift’ in natural science restructures the observations, in a design process each redefinition of the problem restructures the initial building environment.

In the following quotation, Brawne – in accordance with Schön - makes a direct link between a design process and a problem solving process based on decision-making. Although each design action is taken within a broader framework of problem definition, these actions have a certain degree of autonomy:

> The design process and its immediate outcome can be roughly equated with the tentative solution, the tentative theory step, in the move from P1 to P2. It represents the elaboration of a new and alternative hypothesis, irrespective of the degree to which it differs from the previous answer. Any discussion of the design process in these terms also presupposes that it is a problem-solving process in which decisions are possible and that these are not entirely predetermined by the definition of the problem. (Brawne, 1992: 72)

The Brawne's scheme that was quoted at the beginning of this appendix can be now supplemented – a tentative theory can now be replaced by alternative tentative solutions to a problem, each solution corresponding to an alternative building model:

$$\begin{array}{lll} & TS_1 & \\ P_1 \text{ ->} & TS_2 & \text{-> } EE \text{ -> } P_2 \\ & \ldots\ldots & \\ & TS_n & \end{array}$$

Figure 35. Scheme of a problem reformulation in a design process. P_1 is an initial problem formulation, $TS_{1\text{-}n}$ are tentative solutions for a problem, EE is an error elimination from the tentative solutions (evaluation) and finally, P_2 is a problem reformulation (author: Brawne, 1992).

The initial formulation of the problem formulation P_1 leads to a building model adaptation. In the adaptation process, the building model transforms following error elimination (TS_1, TS_2, TS_3… and EE) in order to solve the initial problem P_1. Each adaptation process is a source of new knowledge about the whole design situation and it leads to the new, hopefully better-formulated problem P_2. The shift from P_1 to P_2 is an iterative step.

> (…) in architecture, and probably all the visual arts, the sequence which Karl Popper suggested for science (P_1 ->TS -> EE -> P_2) generally holds good, providing us with the clearest understanding of a process which probably goes on continually from time to time erupts to create major shifts. (Brawne, 1992: 107)

In the above quotation, Brawne supports the tentative approach to architectural design, where each version of design scheme is an attempt to solve a problem formulated within a design context. Figure 36 illustrates these conclusions.

Symbol	**natural science**	**planning**	**architecture**
P1,2…..n	paradigm	problem in planning (socio-cultural problem)	building environment (design intentions and design requirements)
TS1 -	tentative theory (text, equations)	master plan (text, drawings)	building model (drawing, digital 3D etc)
EE -	experiment, test	discussion, "mental "experiment", simulation	test, simulation, "mental experiment"

Figure 36. Scheme of a problem reformulation in a design process. P_1 is an initial problem formulations, $TS_{1,2\ldots n}$ are tentative solutions for a problem P_1, EE is error an elimination from the tentative solutions and finally, $P_{2,3\ldots n}$ are problem reformulations.