THESIS ON CIVIL ENGINEERING F38

Autonomous Design Systems (ADS) in HVAC Field

Synergetics-Based Approach

DMITRI LOGINOV

TUT
PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Civil Engineering
Department of Environmental Engineering

Dissertation was accepted for the defense of the degree of Doctor of Philosophy in Engineering on November 15, 2012

Supervisor: Professor Emeritus Kaido Hääl, Department of Environmental Engineering, Tallinn University of Technology

Opponents: Professor Anatolijs Borodinecs, Riga Technical University, Latvia

PhD Uve Matson, Degensi OÜ, Estonia

Defence of the thesis: December 21, 2012

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

/Dmitri Loginov/

Copyright: Dmitri Loginov, 2012
ISSN 1406-4766
ISBN 978-9949-23-402-8 (publication)
ISBN 978-9949-23-403-5 (PDF)

EHITUS F38

Autonoomsed masinprojekteerimissüsteemid (ADS) kütte- ja ventilatsioonivaldkonnas

Sünergeetikapõhine lähenemine

DMITRI LOGINOV

TTÜ
KIRJASTUS

Loginov D. (2012) Autonomous Design Systems (ADS) in HVAC field. Synergetics – based approach.
Ph.D. Dissertation, Tallinn University of Technology

ABSTRACT

This thesis investigates the problem of modeling the creative component of the engineering design process in HVAC using synergetics-based methods. It is an interdisciplinary exploration between engineering design, theoretical physics (synergetics) and computer science.

The aim of the research was to answer the following research questions: (1) Can synergetics help to model the creative part of the engineering design process? (2) Can synergetic algorithms help to develop the Autonomous Design System (ADS) software? (3) Can the synergetic computer theory be used for the outer walls recognition mechanism in HVAC software on the AutoCAD® platform? We raised the hypothesis that synergetics can be used to model the creative component of the engineering design process.

The method to conduct the study included the following. First, the needed theoretical base was prepared by studying Haken's synergetic mechanisms, and particularly, the theory of synergetic computer. The next step was developing numerical models and their implementation on the AutoCAD platform. Then a further optimization was done to adapt these models for a particular HVAC ADS scenario. As a result, a software application which elaborates the synergetic model was completed. Its testing and implementation analysis should answer the research questions stated above and prove/disprove the research hypothesis. Thus the design was mainly constrained to laboratory experiments.

The results of this study may be grouped in two classes: theoretical and practical ones. In the theoretical part of the work a thorough analysis was performed regarding the use of synergetics and synergetic computer technology in engineering and HVAC design. Particularly, the problem of modeling the creative part of the engineering design process was analyzed and synergetics was proposed as a main tool for that purpose. The research has helped us to introduce the novel notion of the Autonomous Design System (ADS) and to define its properties. The standard model of the synergetic computer was analyzed and modified accordingly, to fit our purposes for the modeling of the creative components of HVAC ADS. The general theory of synergetics and self-organization was analyzed from the philosophical point of view. The results of the analysis of the general theory of the engineering design process are also presented in this thesis. The major practical outcome of this research was the development of the software framework based on synergetic computer implementation which may be run on the AutoCAD platform. This essentially constituted the development of a fully functional component of ADS, namely, the component capable of building's outer walls geometry recognition. The recognition algorithm uses the advanced synergetic computer model which is optimized for one particular design task and relies solely on vector graphics information, i.e., no additional parametric data is needed. This component can be used in real-life software applications (both on traditional ones and on BIM platforms, particularly in the HVAC domain) that use the visual recognition tasks in their work.

Thus, as a result of this investigation, the research questions may be answered in positive manner.

The hypothesis raised in the beginning of this research was proved. Thus, the result of this investigation was a successfully realized ADS system capable of autonomously recognizing and extracting the geometry of a given building.

Keywords: AI, HVAC, System science, ADS, CAD, Complex systems, Self-organization, Synergetics

Loginov D. (2012). Autonoomsed masinprojekteerimissüsteemid (ADS) kütte- ja ventilatsioonivaldkonnas. Sünergeetikapõhine lähenemine.
Doktoritöö, Tallinna Tehnikaülikool.

KOKKUVÕTE

Käesolevas töös on uuritud kütte ja ventilatsioonisüsteemide insenerliku projekteerimisprotsessi loomingulise osa modelleerimist, kus kasutatakse sünergeetikal põhinevaid meetodeid ning nende rakendamise võimalusi masinprojekteerimisel (CAD). Sünergeetika meetodid pärinevad H. Hakeni iseorganiseerumise teooria käsitlusest.

Antud töö on interdistsiplinaarne uuring insenerliku projekteerimise (kütteventilatsiooni valdkond, KV), teoreetilise füüsika (sünergeetika) ja arvutiteaduse piirimaadel.

Uurimuses tuli leida vastused järgmistele küsimustele.

- Kas insenerliku projekteerimisprotsessi loomingulises osas saab kasutada sünergeetikat?
- Kas autonoomse masinprojekteerimise (CAD) väljatöötamisel saab rakendada sünergeetikast pärinevaid algoritme?
- Kas sünergeetikapõhist infotehnoloogiat saab kasutada hoone välisseinte geomeetria äratundmiseks kütte ja ventilatsiooni projekteerimisel AutoCAD tarkvara abil?
- Püstitati hüpotees: sünergeetika põhimõtteid saab kasutada insenerlikus projekteerimisprotsessis (sealhulgas KV alal) loomingulisel modelleerimisel.
- Põhimeetod uuringu läbiviimiseks oli järgmine: esiteks valmistati ette vajalik teoreetiline baas, uurides H. Hakeni sünergeetikamehhanisme ja eriti sünergeetika arvutiteooriat. Järgmiseks sammuks oli arvuliste mudelite loomine ja nende rakendamine AutoCAD platvormil. Seejärel oli vaja neid mudeleid täiendavalt optimeerida ja kohaldada ühele konkreetsele autonoomse masinprojekteerimise rakendusjuhtumile kütte- ja ventilatsioonivaldkonnas. Uurimistöö tulemusel pidi valmima sünergeetikateoorial põhinev tarkvara. Nimetatud tarkvaramooduli testimine/realiseerimine pidi andma vastused töö alguses esitatud küsimustele ja tõestama või ümber lükkama püstitatud hüpoteesi. Seega oli uuring põhiliselt üles ehitatud laboratoorsetele eksperimentidele.

Antud töö tulemused võib jagada kahte rühma: teoreetilised ja praktilised. Töö teoreetilises osas on põhjalikult analüüsitud sünergeetika ja sünergeetikapõhiste infotehnoloogiate kasutamist insenerlikus projekteerimisprotsessis (sealhulgas kütte- ja ventilatsioonivaldkonnas). Eriti süvendatult on uuritud projekteerimisprotsessi loomingulise osa modelleerimise probleemi ja püütud seda lahendada sünergeetikal põhinevate töövahenditega. Sünergeetikateooriat ja projekteerimise loomingulist osa on uuritud üldisest ja filosoofilisest vaatenurgast. Samuti on analüüsitud üldisi insenerliku projekteerimisprotsessi formalisme ja antud lühike ülevaade tehisintellektist. Töös on esmakordselt kirjeldatud ja defineeritud CAD-süsteemi uudset alaliiki – autonoomset masinprojekteerimist *(Autonomous Design System, ADS)*. Pakutakse välja võimalikud küberneetilised (matemaatilised) mudelid süsteemi realiseerimiseks.

Töö praktilises osas on esmakordselt (ja teatud muudatustega) rakendatud sünergeetikapõhist arvuti algoritmi AutoCad platvormil. Algoritmi eesmärgiks on võimaldada vektorkujul esitatud andmete identifitseerimist, täiustades ja arendades edasi olemasolevaid sünergeetikapõhiseid infotehnoloogiaid. Töös kirjeldatakse viit gruppi uurimiste käigus leitud ja testitud klassikaliste arvutusmudelite täiendusi. Üks olulisemaid edasiarendusi, mis otseselt mõjutab vektorkujul objektide äratundmise korrektsust, on kord-parameetrite genereerimise meetodi asendamine nn koosinus-sarnasusel ehk korrelatsiooni koefitsiendil põhineva meetodiga. Edukalt on lahendatud autonoomse masinprojekteerimise süsteemi ühe võimaliku komponendi väljatöötamine ja rakendamine kütte ja ventilatsiooni projekteerimistarkvara automatiseerimisel AutoCAD platvormil. Nimelt võimaldab loodud komponent hoone geomeetria (välisseinad, aknad, ruum) autonoomset äratundmist ja nende omaduste eristamist ning vajadusel kasutamist hoone soojuskadude arvutamisel. Tarkvara komponenti on võimalik kasutada ka teistes CAD ja BIM süsteemides analoogsete, aga ka teistsuguste ülesannete lahendamisel. AutoCAD-i arendamisel on rakendatud kõige tõhusamat olemasolevat API tehnoloogiat ObjectARX. Seda kasutavad kogu maailma professionaalid vertikaalsete rakenduste loomisel Autodeski tootegruppidele ning ka Autodeski enda meeskond kasutab seda alusplatvormide (sealhulgas näiteks AutoCAD, Revit jt.) arendamisel.

Uurimiste tulemusena on esitatud küsimused saanud positiivsed vastused ja püstitatud hüpotees on leidnud tõestuse. Doktoritöö eesmärgid on saavutatud. Seega võib öelda, et antud töö on edukalt teostatud, täidab oma eesmärke ja annab positiivseid tulemusi.

Märksõnad: tehisintellekt, küte ja ventilatsioon, süsteemiteadus, ADS, CAD, keerulised süsteemid, iseorganiseerumine, sünergeetika.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The dissertation is a conclusion of my research work conducted in 2007-2012 at the Chair of Heating and Ventilation, in the Institute of Environmental Engineering of Tallinn University of Technology, and also the extension of the original research focused on automation of engineering design and started back in the year 2000.

Special gratitude is expressed to my supervisor Prof. Emeritus Kaido Hääl. Without his constant motivation and irreplaceable help and support, the studies would definitely not have been successful. Especially commendable is his understanding of the importance and promotion of interdisciplinary research. Interdisciplinary research preparation and education are central to future competitiveness, because knowledge creation and innovation frequently occur at the interface of disciplines. The given case is especially complex as it constitutes an exploration of several fields of expertise.

In addition, I would like to thank Prof. Teet-Andrus Kõiv, for his valuable comments, quick responses, help and hospitality during the years, and especially for his advice on writing scientific publications. I would like to express my gratitude to Prof. Ahto Kalja for inspiring me to the research of CAD systems using the AI technology and to Assoc. Prof. Heino Möldre for his help and support in AutoCAD related issues.

The work was financially supported by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa.

Finally, I would like to thank all of my family members for supporting me during the course of this work. You have been patient and understanding and I love you all. I would like to dedicate this dissertation to my parents, my wife Gayane, and children Anna and Boris. Their profound and unconditional love and belief in me has lit the path for me throughout my life. This dissertation is the fruit of my labor and their love.

During my studies I worked in engineering design and consulting companies where I had the opportunity to test and apply the results of my research to real-life situations. I believe that progress in applied sciences depends heavily on cooperation between companies and universities.

Tallinn, November 2012 Dmitri Loginov

1 INTRODUCTION

This chapter presents an overview of the thesis. First, background is given on the interdisciplinarity and the subject of the research. In the next section an example of the research methodology that has been discovered by the approach proposed in this dissertation is described. This will help to better understand the more abstract and formal discussions that follow. Then, the motivation for pursuing this research is given, followed by a formal problem statement. Next, the goal and objectives are formulated, the significance of the research is indicated, the approach and implementation are reviewed, and finally, the outcomes of and potential applications for this work are described.

1.1 Background

In order to succeed in today's global, competitive market, manufacturing industries and design departments need continuous improvements in their multidisciplinary design processes. These improvements should result in expending fewer resources on the design process while achieving better quality and more environmentally friendly products. The current approach for improving design processes is mostly based on intuitive observations followed by incremental changes to the existing methodologies. However, today's fast-paced world needs rapid incorporation of new technologies and methods into design methodologies. Recent advances in the application of Artificial Intelligence to design domain, Synergetic Computer theory, in particular provide an opportunity to accomplish this goal. The inter-disciplinary collaboration between Computer Science and Engineering Design provides the means to develop systematic and holistic approaches for constructing superior design methodologies and optimizing design processes.

The context of this research is the analysis of possible applications of synergetics (synergetic computer) in Heating Ventilation and Air Conditioning (HVAC) field. Specifically, this research addresses issues within the context of HVAC design process. As the practical side of this research, the software applications have been developed to illustrate the usage of synergetics in design process. The software application is based on AutoCAD platform and developed with ObjectARX®, which is object-oriented C++ application programming interface for developers to use, customize, and extend AutoCAD and AutoCAD-based products.

This research is an inter-disciplinary exploration between engineering design, theoretical physics and computer science. The positive side of an inter-disciplinary exploration is taking advantage of the power of the different disciplines. However, the inter-disciplinary aspect makes the research more difficult to start and also to present to discipline-based people.

In this work, the recent advances in Artificial Intelligence (AI) theories and techniques (synergetics, synergetic computer and Synergetic Neural Network or

SNN) are used alongside with the practical/theoretical and philosophical considerations of traditional AI technologies, e.g., such as artificial neural networks (ANN), genetic algorithms and knowledge-based systems.

The main rationale for the study is the need to fill the gap in modern CAD software. Namely, the creative part of the engineering design, contrary to the routine one, is poorly automated (if at all), and if automated, then by means of classical IT methods, which lead to aggregate growth of data, and thus, computer power needed to deal with such procedures. This actually kills creativity and downcasts creative problems to routine problems. The problems of creativity should be solved in a more elegant way. Hopefully, the answer to this is synergetics. We raise the hypothesis that synergetics can be used to model the creative component of the engineering design process. We investigate the possibility of modeling the creative component of the design process using the theory of synergetics, which addresses complexity by introducing models that are very close to natural systems, being at the same time relatively simple.

As synergetic models are closely connected to their representations, synergetics has solid philosophical motivation and holds promising paradigmatic importance for contemporary science. One of the additional benefits of this research is philosophical outlook on the problem, providing general framework and connecting this particular problem to the state-of-the-art trends and findings of the modern science of complexity.

1.2 The Methodology

The research methodology provides an answer to the question: *How should we conduct the research process in order to analyze the use of synergetics in engineering (HVAC) design?* Some of the questions that originate from the general question are as follows:

- Should it be a theoretical research, based on general knowledge and formal logic, or a practical one, in which specific knowledge is extracted from the real working system, or should we use a combination of them?
- What type of application (if any) should be used for analysis and testing of synergetic models?
- In what order should the research methods be used?
- What are the usefulness criteria for implementation of synergetic theory in the design process?

This research is intended to answer the following research questions. The questions are ordered by their scope and each preceding question may be broken down to the subsequent one.

1. Can synergetics help to model the creative part of the engineering design process?
2. Can synergetic algorithms help to develop the Autonomous Design System (ADS) software?

3. Can the synergetic computer theory be used for the outer walls recognition mechanism in HVAC software on the AutoCAD® platform?

The method to conduct the study includes the following. First, the needed theoretical base is prepared by studying Haken's synergetic mechanisms, and particularly, the mathematical theory of synergetic computer and also the philosophical aspects of synergetics and self-organization. It is important to note that synergetics, being perfectly describable by mathematics, provides a top-down approach to the problem of modeling the system's properties, meaning that first the desired properties of the system are selected, or inother words, these properties can be formulated in advance. This allowes us to first philosophically justify the desired system/model, and then describe and recreate its characteristics in mathematical (synergetic) form. The next step is developing numerical models and their implementation on the AutoCAD platform. Then, a further optimization is done to adapt these models for a particular HVAC ADS scenario. As a result, a software application which elaborates the synergetic model is completed. Its testing and implementation has to answer the research questions stated above and prove or disprove the research hypothesis. Thus the design is mainly constrained to laboratory experiments.

Below is an overview of the methodology discovered based on the approach proposed in this dissertation. This methodology answers the above questions, thus facilitating the process of analyzing the usage scale of synergetics in the field of HVAC design.

In the rest of this chapter we provide an overview of this dissertation in a more formal way by answering the ‘why’, ‘what’, and ‘how’ questions regarding the approach proposed.

One of the goals of this research has been to answer the research questions and prove or disprove the hypothesis raised. This has been accomplished through composing a numerical model and its testing and analysis. Thus, laboratory experiments have been chosen as the main tool for hypothesis testing. As the research chosen has a combined theoretical and practical design, there are no alternatives to practical/laboratory experimentation to judge the successfulness of our approach and models.

In order to successfully complete the research project we have organized the ***research process methodology*** into several consecutive steps (some major known difficulties are outlined in parentheses):

1. *Analysis* of the general synergetics theory to answer the (preliminary) question whether it is suitable for modeling the creative components of HVAC design. This step includes the philosophical consideration of the problem as well.

2. *Creation* of a general numeric model of a serial synergetic computer algorithm. (As the synergetics theory is available solely in the form of analytical mathematical expressions, casting it into numerical form suitable for computer applications has been a challenge.)

3. *Selection* of an appropriate HVAC design example suitable for a practical software application: research and find out a sample design routine to demonstrate usage of synergetics technology

4. *Adaptation* of this algorithm *for the AutoCAD* environment and *for the particular HVAC problem*, e.g., outer walls recognition process of the buildings heat loss calculation and its testing. (The main problem here is to find out how to suit recognition algorithm to ACAD entities/architecture properties as these are vector geometry data. Another problem is to find a mechanism for subclassing prototype/test patterns into smaller parts that could be recognized, as there are too many different floor plan shapes to use these as prototype patterns.)

5. Based on the previous steps, *creation of* the specific ADS *software component* that can be used in real-life applications and in future research The software program is intended to automatically identify and select the building's (or as a prototype, a single room's) outer walls. Such functionality is close to human visual perception and constitutes one of the properties of ADS and also mimics an engineer's creativity. This step is not obligatory, nor does it affect the general properties of the model. Therefore, this can be omitted or left for further research.

6. *Testing* of the developed *software component* and proposing the usage of the application module in real-life situations and/or design process scenarios

7. *Analysis* of this particular implementation of the synergetics theory: namely, the analysis and conclusions based both on the theory developed/considered and practical examples

8. *Final analysis* of the results, their description and documentation.

All the steps described are eligible only if the solution to each previous problem is positive. However, as this is a completely new approach, the result of the experimentation is hardly predictable and it may be even a negative one i.e. the disproval of any or even some of the sub/hypotheses. The methodology discovered may also be adjusted/changed during the course of the research.

1.3 Motivation

The motivation for this work in short has been *the need for better ways of doing design* in today's design/manufacturing companies. For many companies, this need is a matter of being able to compete, and thus, to survive, in today's fast-paced world. Methods and tools are needed to automate the design process and systematically generate better design methodologies at the same speed as new technologies are emerging. Improving design processes based merely on ad-hoc approaches and intuition are no longer adequate. New methods and techniques from the area of Artificial Intelligence in Design and Synergetics are at the stage of maturity where they can provide better alternatives for improving design processes.

Engineering design can be viewed as an articulate process composed of phases, where each phase represents a combinatorial action on the parts the composite object consists of. To realize an object meeting the desired market requirements, engineering designers have to deal with different kinds of knowledge about objects, or ontological knowledge (which is often represented in a declarative form) and dynamic knowledge about processes (which is often represented in "procedural terms").

There is a lot of CAD software on the market today that simplifies engineers' daily work and reduces human errors in the design process. Nevertheless, a lot of design errors are still generated during the process, which leads to errors in construction and sequentially to extra project costs, environmental impacts and so on. Furthermore, as the software becomes more and more complex and powerful and automates most parts of the process, taking away control from the human, it becomes hard to track down design errors during the process flow.

There are two major problems, solutions to which are intended to be given with this research. The first one is the existence of design errors occurring within the process of CAD development. The second is the absence of an adequate solution to modeling of the creative part of the engineering design process.

Both of these problems have been addressed by implementing a synergetic computer algorithm on CAD platform, i.e., implementing one of the components of the Autonomous Design System (ADS). Namely, it is intended to create a numerical model for the outer walls recognition mechanism in HVAC software on the AutoCAD platform.

Contrary to traditional (e.g., parametric models in BIM software) IT approach, the synergetic theory allows using simple Artificial Neural Networks (ANN) synergetics based models, minimizing data cumulation and facilitating faster design. Synergetics-based pattern recognition mechanism is closer to the natural human perception, being at the same time both philosophically and experimentally motivated. It constitutes a novel approach, which lies in accordance with general universal laws, discovered recently by the science of complexity and self-organization.

Engineering design process could be divided in two parts: the routine and the creative one. The routine part of the design process consists of the numerically analyzable elements. The numeric model could then be further improved and optimized. Here all the benefits and achievements of the digital revolution, including Artificial Intelligence (AI), can be used to automate the process of engineering design. On the other hand, the design process consists of the "creative" part which is not numerically describable by traditional numerical algorithms, at least not yet.

If we could invent the way to model/automate the most complex/challenging part of the design process, the creative component, the engineering efforts/errors could be minimized, and ultimately better designs (specifically in HVAC), which save investments and environment, could be created.

In this thesis, the problem of modeling the creative part of the engineering design process has to be analyzed from the perspective of synergetics. The characteristics of the creative tasks of engineering design have been defined and the novel notion of the Autonomous Design System (ADS) has been introduced.

The synergetic computer algorithms are optimized for a particular HVAC CAD case and the part of the real life software application prototype is created exploiting these principles.

ADS is considered an advanced CAD system that has Artificial Intelligence (AI) functionality and particularly the functionality to deal with creative components of the engineering design process.

Synergetics can be considered one of the modern, most promising research programs. It is oriented towards the search for common patterns of evolution and self-organization of complex systems of any kind, regardless of the concrete nature of their elements or subsystems.

Synergetics (in the meaning of H. Haken's school of thought) provides mathematical tools to cope with the self-organization phenomenon. These tools are based on the combination of the differential equations theory and the stochastic modeling.

The principles of synergetics (i.e. self-organization theory) are really of a universal nature and have paradigmatic importance in modern science. There is also a solid philosophical foundation to that phenomenon and this is a very promising and novel approach to the world life systems in general and philosophy of science in particular.

Hypothesis: synergetics (synergetic computer theory) may be successfully used to model the creative component of the engineering design process.

Sub hypothesis: synergetic algorithms (and self-organization paradigm in general) reflect the creativity of the human being. Thus, creativity is derived from self-organization.

The following is a summary of the motivations for conducting this research:

- **Need for Continuous Improvement and Design Error Reduction.** In order to succeed in today's global competitive market, companies need continuous improvements in their design processes. These improvements should result in expending fewer resources on the design process while achieving better quality and more environmentally friendly products.
- **Need for Rapid Incorporation of New Technologies**. New technologies (e.g., new materials, manufacturing processes, etc.) are emerging into design products at an increasingly fast pace. These new technologies can not only improve the quality of products, but can also provide better ways of conducting the design process. In this situation, we need to incorporate the new technologies and methods into design processes as quickly as they appear.
- **Need for Integration.** Integration of multidisciplinary and inter-process design is a means to enhance the quality of design, reduce the cost and the time to market, and incorporate environmental considerations into the design of the product. Integration reduces the number of failures and backtracking by facilitating information sharing, and thus saves

resources. On the other hand, integration provides collaboration between different participants (both from the same and different design disciplines) that as a result enhances the quality of design.

- **Need for Concurrency in Design.** "It is well known that concurrent decision making is an important and very desirable component of modern design methodology"[1]. A concurrent strategy, in contrast to a sequential strategy, carries out some of the problem-solving activities in parallel to each other. As a result, the design process speeds up, because the participants in the design do not have to wait in a line if they can make a contribution.
- **Need for Design Assistant (CAD) Tools.** There is a need for design assistant tools that can help designers in their everyday work. This need is constant since not all engineering routines are well automated yet (especially the creative parts) and there are always technology improvements that influence the design process. Moreover, nowadays it is becoming harder to improve the system performance of engineering devices based merely on advances in individual disciplines and non-distributed architecture. In other words, improvements in individual disciplines alone are not sufficient to affect the improvements in products and processes needed in the future. To achieve higher quality, system-oriented, holistic, multidisciplinary and distributed approaches to the design of engineered systems are needed that consume less resources [2].
- **Recent Advancements in Artificial Intelligence in Design and in the Science of Self-Organization (Synergetics).** Recent advances in the application of Artificial Intelligence in design, synergetics in particular, provide an opportunity to build superior design methodologies/systems and automate design processes. Theories and techniques from Artificial Intelligence that have recently become available enable engineering design researchers to take advantage of the computational power of computers in solving their problems. These advances allow formulating a new kind of CAD system – the Autonomous Design System or ADS, where the creative parts of the design process are automated as well. Therefore, the problem of modeling the creative component of engineering design has to be investigated.
- **Need for Testing the Hypothesis Raised.** It is intended to analyze and test the hypothesis and subhypothesis raised prior to conducting the research. For that purpose, both general and philosophical analyses on the subject are needed, as well as practical experimentation (mostly in the form of numerical modeling).

1.4 Problem

The main problem is the following:

"*There is no solution to the problem of how to automate/model the creative components of the engineering design process. There is no answer to the question whether this is possible at all.*"

Sub-problems:

"*The existence of (human) design errors during the flow of engineering design process that may result from poor/inadequate automation of this process.*"

"There is no real-life software application in the HVAC field that utilizes the synergetic models theory and has the properties of ADS on the market today and there are no evident applications of synergetics in HVAC that are publicly acknowledged, i.e., no analysis has yet been performed regarding this subject."

The following factors contribute to the difficulty of the problems:

- The current approaches for improving design processes are mostly based on intuitive observations followed by incremental changes to the existing methodologies. The current practices of multidisciplinary design are based on ad hoc strategies for handling the complexities that multiple points of view bring to the design process. These techniques solve the problem of complexity at the expense of giving up the potential advantages of diversity. The common methodologies for multidisciplinary design are based on compromising between different disciplines rather than collaborating between them.
- The creative parts of the design process are characterized by the higher intelligence needed to deal with them. Therefore, if we want to model that part of engineering design we need far more powerful AI technologies than those available today. It may even prove impossible to model/automate these creative components by means of today's computers. Some new revolutionary technology may be needed to do that. However, there is a belief that some of these components may still be approximated (to some degree) by mathematical methods that are readily available now (by their improvement) or by the newest methods that have emerged recently, or those, still under development at the moment. The AI tools and technologies in cooperation with synergetics, for instance, may help us to achieve that goal.
- One aspect of design that is usually neglected is that design knowledge is constantly evolving [3]. Designers must not only cope with a complex task, but they must track the evolution of a domain. In this situation, designers have to determine whether new knowledge is related to the body of existing knowledge, or whether this new knowledge reflects a more fundamental change in technology. The latter may have a large effect on the application's problem solving behavior.
- The number of specialists is increasing, while the number of generalists, capable of doing system integration, is decreasing. At the same time, the

knowledge burden on the designer keeps increasing as more materials and more options become available [2].

- One of the sub-problems addressed by this research is the fact that ANN algorithms popular today still rely heavily on the model of McCulloch and Pitts (1943), which has certain disadvantages. One of them is that there is no general theory of what a network can really do or how it could be trained in a reliable and fast manner. Neural nets are also prone to be trapped into so-called spurious states. Neither of these learning difficulties occurs in the synergetic computer algorithm, introduced and developed by Hermann Haken. Equations of synergetic computer may be solved on a serial computer, but they also provide us with the construction principle of a new type of parallel network, in which the individual nodes or neurons have quite different properties from those of the previous neural computers in the sense of McCulloch and Pitts.

Some additional researchable questions:
- Can synergetics really help to model the creative part of the design process?
- What exactly is ADS? What parts does it consist of?
- Do we actually need to model these creative components? Should it be left to the human (designer) instead?

1.5 Goal and objectives

The goal of this research is the following:

"To analyze the problem of modeling the creative parts of the engineering design in the HVAC design process by using the synergetics approach (theory)."

To achieve the above goal, we need a complete understanding of the theoretical aspects of the synergetics theory and design process formalisms in general. In particular, we need the understanding and practical experiences in the HVAC design process. Finally, it is good to prove (or to disprove) the theoretical conclusions with the practical model implementation, in our case, by means of developing a software solution based on synergetic models.

The objectives are to find out some of the most useful application domains of synergetics and synergetic computer in the given field, to optimize them and to test them in real-life situations. By doing this, we hope to prepare the theoretical and practical knowledge platform for future research and implementations. We need to perform this analysis in order to improve (hopefully) the design process characteristics (quality, speed, etc.) by means of synergetics. An overview of the technologies used in this research is given later in this paper.

By accomplishing these objectives we systematically overcome the complexities that the eccentricity of interdisciplinary research brings into our analysis efforts.

1.6 Approach

We propose an approach to the problem of analyzing the synergetics theory application possibilities and modeling the creative part of the engineering design based on practical implementation of synergetics, i.e., by making experiments. On the other hand, the approach to the second part of the problem, which is the creation of a practical synergetic computer technology application in ADS (the computer program), is to perform an analysis regarding the subject of this research. Thus, we are looking for a twofold solution.

However, there is an intention to concentrate on the theoretical aspect of the problem, leaving the practical part as a sort of testing mechanism to prove our analytical implications.

We will use deduction, induction, formal logic and other analytical methods to formulate theoretical conclusions. We will also perform a philosophical analysis on the subject.

The design and realization of a synergetics-based solution is a complex task. In this work, an attempt to add even more complexity to the synergetic system's usual attributes is made. This is accomplished by applying the synergetics technology to ADS system's software application, as we assume that synergetics may be used as a primary tool for modeling the creative components of engineering design.

1.7 Significance

The following is a summary of the significant aspects of the research, which partly coincide with the research's contributions.

- *Application of synergetics/synergetic computer theory in HVAC and on AutoCAD platform*. One particular design domain is analyzed and tested from the perspective of synergetics. There are no known applications of this technology in HVAC design and on AutoCAD platform yet. This work aims at researching that subject.

- *Introduction and explanation of the novel notion of the Autonomous Design System (ADS).* ADS are considered as advanced CAD systems that have the Artificial Intelligence (AI) functionality, and particularly, the functionality to deal with creative components of the engineering design process. The main tool to model the creative part of the ADS is synergetics.

- *Analysis of the problem of modeling the creative components of the engineering design process both from the technology, implementation, general and philosophical points of view.*

- *Proposal of one possible way of eliminating human errors in the design process.* This is accomplished through developing and formalizing an ADS system that uses synergetics to model the creative (i.e., non-routine) tasks of the CAD design process, thus providing a higher level of automation functionality and reducing the designer's potentially error-prone functional efforts.

- *Thorough and critical analysis of Hermann Haken's standard model of the synergetic computer.* Potential shortcomings of the classical model are discussed and possible ways of its improvement are highlighted.
- *Adaptation of the standard model of the synergetic computer to the AutoCAD environment and to one particular HVAC ADS usage case.* This means the theoretical adaptation of the classical algorithm and its realization in the form of a CAD software application.
- *Reduction of the time to market and saving resources.* To be able to compete, companies need not only continuous improvements in the quality of their products, but also enhanced performance of their design and manufacturing processes in order to reduce the cost and the time to market. Integration facilitates information sharing among multiple and often contradictory points of view. As a result, the number of failures and the amount of backtracking in the design process are reduced, thus saving resources and shortening the design time. Integration also provides collaboration between different participants that, as a result, enhances the quality of the design.

For example, the software agent's technology (or any other traditional cybernetic AI technology) may be used in combination with synergetics. This allows using the benefits of integration in the design process seamlessly.

- *Biasing the design process toward more environmentally friendly products.* As the alternative methods that are built into each part of the compound ADS model are tried in a preferential order, and as each method tends to contribute differently towards the final properties of the design, it is possible to bias the design process towards particular properties. Error reduction and overall design process optimization will ultimately lead to the preservation of natural recourses and environment.
- *Attacking the problem of integration in multidisciplinary design.* The number of specialists is increasing, while the number of generalists, capable of doing system integration, is decreasing. In addition, the knowledge burden on the designer keeps increasing due to a wider choice of materials and more options. Thus, it is becoming harder to solve integration problems in a traditional manner. The synergetics-based approach may be the answer to this problem.
- *Enabling designers to break out of disciplinary confines.* An increasingly specialized technological environment tends to force designers to concentrate on specialized aspects of the design process. This research allows designers to consider the design process as a whole. In addition, the outlook on the creative parts of the designer's work helps to acknowledge the engineering design process as being a large, complex system that exhibits the properties of self-organized systems and thus obeys the principles of synergetics.
- *Application of computers to new areas of engineering design.* Computers have mostly been used to support the manipulation and analysis of design product information. This work focuses on the design process, an aspect that has not benefited from computers very much. Regarding modeling (including both

theoretical and practical aspects) of the creative part of the design process, computers have not been used at all.

- *Incorporating new software methods.* Aiding design processes based on a synergetics paradigm is a new area of research that has a high potential for practical as well as theoretical impact on the design of products and buildings.

- *Interdisciplinary research and its impact on engineering design research.* This work benefits from the interdisciplinary contribution of the state-of-the-art developments in Artificial Intelligence, Theoretical Physics/Complex Systems and Engineering Design. According to a recent report by the Council of Graduate Schools [4] “interdisciplinary research preparation and education are central to future competitiveness, because knowledge creation and innovation frequently occur at the interface of disciplines”. This work covers all of these areas of research and is hence expected to have a strong impact.

1.8 Contribution of the research

These are the most prominent features of this research that are developed solely by the author of the research and thus constitute the valued, unique and direct contribution of this work.

As an additional contribution of the research, an application for a patent for the method of synergetic recognition of vector graphics objects in the AutoCAD environment was submitted a couple of months ago, and the probability of obtaining the respective patent is considered to be very high.

Contribution:

1. 1. Thorough analysis of H. Haken's synergetic computer theory and its adaptation for modeling creative components in the HVAC design process
2. Proving/disproving abovementioned hypotheses (through the numerical testing on the CAD platform)
3. Developing the conceptual framework for modeling the creative component of the engineering design process in the AutoCAD environment using synergetics-based approach
4. For the first time, implementation of the synergetic computer algorithm on the AutoCAD platform and in the HVAC field
5. Developing the fully functional component of ADS (e.g., a building’s outer walls geometry recognition, which relies solely on vector graphics, i.e., no additional parametric data are needed) that can be used in real-life software applications (both on traditional ones and on BIM platforms, particularly in the HVAC domain)
6. Introduction and analysis of the novel notion of the Autonomous Design System (ADS)
7. Developing a solid philosophical foundation and motivation for the study conducted

1.9 Outcome and potential applications

The potential outcome of this research is a thorough analysis of the problem of modeling the creative component of the engineering design in the HVAC design process, finding out most useful application cases and optimization of these applications. Such applications lead to design practices that consume fewer resources and provide better product quality.

The result of this research is expected to be a generic theoretical approach to the problem of defining and optimizing appropriate usage of creative parts of the CAD system, particularly as a part of HVAC ADS, and a practical implementation of a synergetic computer algorithm on the AutoCAD platform. There is an intention to create a computer program that utilizes the strengths of the synergetic computer and helps a designer of a HVAC system in his/her daily work. Particularly, it is intended to create a software application (as part of HVAC ADS) that automatically detects the building's/room's outer peripheries and extracts its geometry for, e.g., further heat loss calculation. The objective of such a program is to prove (or disprove) the theoretical implications made throughout this paper.

The primary outcome of this study is an academic one. The practical part of the research will concentrate mostly on testing synergetic computer algorithms in the HVAC field and on proving/disproving the hypotheses raised. However, it is expected that the computer program developed finds, after some minor modifications, immediate practical implementation in real-life design scenarios. Therefore, special attention is paid to the selection of the underlying computational algorithms and their optimization, to allow for designing a software application that is user friendly, efficient, intuitive and easy to use. Possible directions for future research:

- Improvement and further optimization of the synergetic computer algorithm. Introducing “arrow of time”, making it more “self-organizing”.
- Application of this algorithm in modeling of other creative parts of the design process.
- Making a fully functional real-life software application ready for the commercial use, implementing complete automation for outer-inner walls recognition mechanism and autonomous calculation of a building's heat losses based solely on vector data representation (i.e., visual perception).

The intended *results* of this research:

- If the methodology presented is successfully implemented, the results of the research are those stated in the contribution section of this thesis (in section 1.8). In any case we must get the answers to the research questions and prove/disprove the hypothesis.

1.10 Outline of the dissertation

This section provides an overview of how the rest of the dissertation is organized.

- Chapter 2 is an in-depth study of design formalisms and the problem of multidisciplinary and computer aided design. We review the major issues of the engineering design process including multidisciplinary design, integration among different disciplines/processes and concurrency among them.
- In Chapter 3 the generic issues concerning the AI domain are discussed in order to prepare the reader for a more specialized discussion.
- Chapter 4 gives a condensed general overview of synergetics and self-organization theories. Additionally, the problem of modeling the creative modules of the design process and the self-organization paradigm are described from the philosophical point of view. This chapter includes some theoretical contributions of the dissertation.
- Chapter 5 describes the fundamentals of the Synergetic Computer and Synergetic Neural Network (SNN) theories. An overview of the related research and literature is also presented.
- Chapter 6 defines and describes the notion of the Autonomous Design System (ADS) and its relation to synergetics and modeling the creative part of the design process. Here, theoretical conclusions are developed according to the goal of this research. This chapter also includes the theoretical part of the contribution of the dissertation.
- Chapter 7 describes the theory and practical experiments of the adaptation of the standard theory of the synergetic computer to AutoCAD and to HVAC ADS. This chapter includes both theoretical and practical contributions of this dissertation.
- Chapter 8 describes the implementation issues and the practical contributions of this dissertation that concern building a software application based on the synergetic computer paradigm as part of HVAC ADS.
- Chapter 9 comprises an overview of the data collected from experiments and analytical implications, presenting the results of data processing according to the approach proposed in order to fulfill the goal of the thesis.
- Chapter 10 closes the loop by revisiting the goal to see if it has been reached. It summarizes the results of this dissertation and makes some conclusions based on the results presented in Chapter 9.

2 ENGINEERING DESIGN PROCESS

2.1 Introduction

Designing is a complex human process that has resisted comprehensive description and understanding. All artifacts surrounding us are the results of designing. Creating these artifacts involves making a great many decisions, which suggests that designing can be viewed as a decision-making process. An abstract description of the artifact using mathematical expressions of relevant natural laws, experience, and geometry is the mathematical model of the artifact. This model may contain many alternative designs, so criteria for comparing these alternatives can be introduced in the model. Within the limitations of such a model, the best, or optimum, design can be identified with the aid of mathematical methods [5].

In this chapter, the engineering design process paradigm is considered from a general point of view. The modern understanding of the design process is explained in general, and civil engineering design process/ HVAC systems design in particular. The major issues of the engineering design process are reviewed, including multidisciplinary design, integration among different disciplines/processes and concurrency among them. The information presented in this chapter is based mainly on Cirrus Shakeri work “Discovery of Design Methodologies for the Integration of Multi-disciplinary Design Problems”.

2.2 Engineering design

2.2.1 The essence of engineering

What is engineering? There are several ways of answering this question. Engineers have some trouble answering it in a completely satisfactory way, particularly if the intention is to decide who is practicing engineering and who is not. All engineers will, however, agree with the statement that engineering is "turning ideas into reality”. Engineers turn ideas into useful products or systems, and the heart of engineering is the engineering process, sometimes called the *engineering design process*.

This is not to suggest that all engineers actually work at design. There are numerous types of jobs done by engineers. All are important in engineering and design is only one of these functions. Yet engineering design is the key activity, which distinguishes engineering from other occupations. Yet design is also a term to describe activities which are central to many other occupations, such as the design of clothes, advertisements, the external features of numerous consumer products from cars to cutlery, most of which involves consideration of aesthetics.

Aesthetics can be important in engineering too, but full-scale engineering design usually includes a good deal of *analysis*, much of it technical, based on an understanding of subjects such as mechanics, electronics, electricity, fluid dynamics, thermodynamics, and the like. Analysis is the application of science to design, applying the laws of science and mathematics in evaluating a design to ensure that it will work as intended. Such analysis will follow procedures and methods developed through research [6]. No product design can avoid testing to see if it functions as expected, and most engineering product design involves development, i.e., building prototypes, testing them, and then modifying the design before more testing is done, etc.

All major engineering projects are conducted by teams which have to be managed, and most engineering occurs in a business setting, so *good management* by *people who understand engineering* is vital to the success of the engineering business. This includes engineering which is done on a consulting basis, i.e., by engineers providing services to clients who are not their employers.

Of course, analysis, research, management, etc. can all be activities, which have nothing to do with engineering. For instance, these terms are used in connection with financial and stock market businesses. It is the connection to engineering design that makes these relevant to engineering. As far as the creative process is concerned, there are similarities between all occupations involving design. However, the engineering design process typically follows the whole sequence outlined here, and the technical knowledge and techniques employed are special to engineering. Not all engineering design will follow a particular sequence clearly or fully, and there are variations in the description of engineering design in different texts. What applies well as a description of design in one kind of engineering may not be common in another kind of engineering. Design is usually described as a linear sequence of steps each of which are performed once, but it can also be a cycle, which is followed more than once. Furthermore, parts of the sequence may involve loops, i.e., cycles of parts of the sequence. Designer(s) go over the same stages repeatedly, such as generating ideas, refining them, and then going back to generate more ideas and refining them a bit before making any choices. This section describes design in considerable detail, and several of the steps are described in some detail, considering also some of the technical analysis involved. The terms used to describe design may vary from case to case, from text to text, and not all stages in a design process may be followed clearly in a particular case.

Engineering design is sometimes confused with engineering drawing. While various forms of drawing are important in design and as tools for thinking through ideas, expressing them, and providing instruction for the production of artifacts, design itself is the production of solutions to specific problems. Often innovative solutions, i.e., solutions that are better than any of those used before, are aimed at to produce a new product or a better way of achieving some goal.

2.2.2 Need and opportunity

Engineering deals with ideas, which arise in response to a need or opportunity, and the two can often be distinguished between, though the engineering response to each is much the same. If a number of houses is built together, as in a new sub-division, you have to provide the houses with water and ways of disposing of sewage. Thus, there is a need for a water supply system and a sewage system, and the design of the water supply and sewage systems will be in response to this need.

On the other hand, many consumer products are first developed as an opportunity. For instance, nobody thought there was a pressing need for a motor car when the first cars were designed and produced. The earliest cars were in fact little more than toys. Everyone traveled on foot, on horse or by horse drawn carriages, or by steam train, and by boat where possible. Producing cars was a technical challenge and an opportunity to make money, yet the only needs satisfied in the early years of automobiles were sporting or recreational needs. Of course, owning a car is now a necessity for many, and the production of cars is still a business opportunity. For that matter, when a city has a need for a water supply, a sewage system, a bridge, etc, this presents an opportunity for the engineer to design the artifact, and for the contractor to build the structure, to fill the need.

2.2.3 Main design stages

2.2.3.1 Problem identification, definition, specification and information gathering

Design is often described as starting with a "problem definition", deciding why the engineer wants to design something. The reason is usually that there is a need or an opportunity. This need or opportunity amounts to a problem, which the engineer must solve, a task to accomplish. The first step in engineering design is to be clear on what this need or opportunity is, what the task is. If an engineer has to design a car, what does the car have to be able to do? The same would be true if designing a stepladder, or a coffee percolator. It is also important to understand clearly what the designed object or process has to accomplish and what may be the restrictions for the designer, i.e., the criteria the item has to meet and what constraints will restrict the design, production, and operation of the item? Thus, there are essentially two parts to a set of *problem specifications* in a *problem definition*: recognition of the *need or opportunity* and identification of the *requirements and limitations*, also known as the *criteria and constraints*. This stage is also often referred to as *problem identification.* The term "problem" does not necessarily imply that there is any major difficulty involved, but simply that there is a task to be done. Some of the specifications may be obvious, others may require some research to determine, i.e., *information must be gathered* on the intended use of the product, what conditions it will be used in, what is necessary and what is available in parts and materials, and how much it may cost. There may be designs, which have worked in the past, which can be studied in order to find out how they can be improved.

Studying the need or opportunity and developing a problem definition are vital steps in a successful design. If the engineer does not know clearly what he/she should be achieving he/she is unlikely to succeed. It is not so much the general idea that counts but the detail. For instance, it may be obvious that a bridge is needed across a river. The real issues are what kind of traffic the bridge should be able to carry, what effect, if any, it can have on water flow, and on traffic on the river, what the soil and weather conditions are (firmness of foundation and pier locations, rain fall, winds, flooding), how long it should last, what materials are available, what access to the site is like, what kind of technology is available for construction, who is going to maintain it, what the maintenance costs may be. There will be certain criteria that must be met, i.e., limitations on what will work, including perhaps a maximum price, restrictions on dimensions, safety or environmental issues.

2.2.3.2 Alternative designs, preliminary ideas, ideation

There is a big difference between a mathematical problem, and an engineering problem. A mathematical problem has one or more very definite solutions, such as two roots to a quadratic equation. No numbers other than the two roots are correct solutions to this problem, and thus are unique solutions. In most of engineering tasks, however, there are no unique solutions. There is no unique solution to the HVAC system design, for instance. There is no unique way of designing any car, hair dryer, pop can, bridge, building, computer, VCR, electrical power station, etc. In any such case there will be several feasible designs, i.e., several *alternative* designs. One of these may be the best of the lot, but there is no knowing if it is the best possible design, and what seems the best today will be superseded by another design tomorrow.

Unlike solving many mathematical problems, there is no set way of reaching the solutions of an engineering design problem, i.e., no formulae or unique steps, which will lead directly to the solution. The first step is typically simply a casting around for *preliminary ideas*, maybe based on existing designs, or maybe ideas that come from completely different areas of technology. This process is sometimes referred to as "ideation". Often quite different ways of designing a product will suggest themselves and this is the first step in innovation. Besides, very little technical knowledge is needed to come up with ideas, and many inventors are not highly technically qualified. This stage in design is often called "brain-storming". It is, of course, not unusual for some ideas to develop from a previous version of a device, sometimes changes are based on previous experience which has suggested ways of improving the item. Sometimes it is a matter of increasing or decreasing the size of the item, or simply finding ways to reduce manufacturing costs while otherwise leaving the item unchanged. For instance, the number of screws or bolts needed to assemble a ventilation unit's case has been reduced over time, reducing the effort and time needed to assemble them. While there are situations where an existing design is simply adopted and adapted, and to do otherwise would be wasteful of time and resources, if not committed to an existing design the engineer should try to approach each problem with an open mind looking for as many alternatives for the whole concept as possible, as well as for details which might provide a good solution.

The client may have made a basic choice before involving an engineer in the design process. In this case, the engineer's (designer's) freedom of choice may be limited to components and details. Yet, there are often numerous possible combinations of the components as alternatives but no great differences in the principles involved. Furthermore, the choice of a component depends on what is available, and this too is typical of engineering.

Here and further in this work, the term "engineer" will be used in a design process context, meaning the person who creates some kind of design specification, optimization, system analysis, etc. Likewise, the term "designer" is used in an engineering domain context.

2.2.3.3 Evaluation of alternatives, refinement, selection of optimum design, optimization

Once alternative designs have been developed, each should be studied, i.e., evaluated using appropriate criteria. Generally these criteria would be based on the problem specifications. An engineer should be thinking ahead to the evaluation stage when developing a problem definition, deciding how to choose between alternative designs. It may be advisable to select two or three alternatives as likely candidates, leaving the final choice until after each of the shortlisted ones has been analyzed and improved to the extent possible. That means each of the shortlisted alternatives is refined and studied in some detail, perhaps drawing them to scale for comparison. This is called *optimization*, a term used at any stage where improvements are made and also used for changes made once a prototype has been tested.

2.2.3.4 Analysis

Very often ideas are generated for a design without much consideration for whether any of the ideas will really work, or work well.

The Merriam-Webster dictionary gives the following explanation for the term *analysis*:

- Separation of a whole into its component parts
- The identification or separation of ingredients of a substance
- An examination of a complex, its elements, and their relations

If a design is studied using principles and formulae from science and practical experience, this is called analysis. There are various ways for performing an analysis. Graphics can be used for some aspects, others involve calculations, perhaps, with specialized computer software to ensure the item will work and last as long as needed. This analysis will be based on formulae, tables, graphs, and various design codes, i.e., principles based on science and *practical* *experience* accepted by the profession, and any pertinent authority concerned with safety, environmental protection, and durability. If this is done on several alternatives, a decision follows on which one to implement.

2.2.3.5 Decision and implementation of optimal design

Once the best of the alternatives has been chosen, this has to be described in very specific detail so that it can be realized. The various alternatives are typically developed and described using sketches and sometimes models, but the drawings needed for the implementation of the optimal design are critical, as they have to convey specific instructions to those who make the item.

2.2.3.6 Contract specifications and drawings

In large-scale civil engineering, the alternative chosen is usually described in detail in *drawings* and written *contract specifications (or explanatory letter)*, which are specifications for the contractor chosen for the job to work from. These contract specifications are in fact a key part of the contract between the owner of the project and the contractor carrying out the work, and constitute the final stage of the design process, and are not to be confused with the specifications or problem statement described as an early part of the design process. In practice, both the owner and the contractor may want to make some changes to the design while construction proceeds. The owner may decide that the structure can be improved, or the contractor may suggest improvements based on previous experience or alternative component parts, but essentially construction proceeds on the basis of the best alternative in the concept and the component parts. Once construction is complete, all the changes are included in "as built" drawings.

2.2.3.7 Prototypes, optimization, product specifications and drawings

With consumer products, and electronic and mechanical products, it is also usual to produce *detailed product drawings and specifications* of the chosen alternative, which is then constructed as a *prototype* and tested. Based on these tests the design is modified to improve the product, and then the new prototype is tested. These tests may be typical of expected normal use, or may be more elaborate and stringent. This cycle may be repeated several times. This is called *optimization*. Only when the producer is satisfied with the prototype, does mass production commence. This is the process followed by regular car manufacturers as well as the makers of items such as hair dryers, and a similar process is followed in the development of software. In civil engineering there are usually no prototypes, i.e., a building is not tried out before construction (in recent years, however, digital prototyping has been widely used in the domain, e.g., BIM technology, 3D modeling, etc.). However, civil engineers and architects involved in this kind of design, do build on their *experience*, so that ideas used in a building or other structure may be used more widely or used differently in subsequent designs.

2.3 Design process basics

"A *design process* is the series of activities by which the information about the designed object is changed from one information state to another. That is, a design process solves, or resolves, a design problem" [7]. To improve the design process we need prescriptions that advocate how design should be done in particular circumstances [8].

Models of design process can be categorized into three categories: *descriptive, prescriptive, and computational*. The descriptive models simply describe the sequence of activities that typically occur in designing. Prescriptive models attempt to prescribe a better or more appropriate pattern of activities [9]. A computational model expresses a method by which a computer may perform the design process [8].

2.3.1 Models of design process

2.3.1.1 Knowledge-based design

"Engineering design can only be knowledge-based, or else it is guesswork.... And if an intellectual task is knowledge-based, then the knowledge employed can be explicitly identified, organized, codified, studied, and experimented with using knowledge-based computer systems as an experimental apparatus" [7]. A knowledge-based design paradigm applies highly specialized knowledge from expert sources to the synthesis or refinement of a design or a design process [10].

The development of knowledge-based systems for design, especially of mechanical systems, is increasing. The expectation is that these computer systems can improve the quality of design and shorten the design time [11]. These are, for instance, computer *expert systems* that utilize the highly specialized engineering design knowledge or expert knowledge.

A knowledge-based view of design is more appealing to human designers than other methods, such as optimization. Another advantage is that certain tasks are easier to be modeled in knowledge-based systems than using a mathematical model.

2.3.1.2 Systems science approach

In the systems approach the need for a close study of the problem environment is emphasized. Such a study takes place before specifying design requirements. The following are the models for design that would result from Systems Science theories:

- Black Box Theory: This theory models the design as the mapping of the requirements to the design descriptions with respect to the environment.
- State Theory: This theory models the design process by a vector of characteristic attributes representing the internal state of the process. The design process will be transition of current state to the next one.
- Component Integration Theory: This theory models the design by decomposing it into components whose input-output mappings are known. The behavior of the system is derived from the interaction between these components.
- Decision Theory: In this approach all activities of modeling, design, and analysis are modeled as decision making activities. Each decision is made in a systematic way, taking into account dependencies between actions.

2.3.1.3 Problem solving approach

The problem solving approach seeks to reduce a design problem to subproblems recursively until subproblems solutions are directly known. Subproblem interdependencies are formulated as constraints to be satisfied.

Due to the interdependencies between subproblems, a search process is needed to determine in what order the subproblems should be solved. The search is formulated in terms of problem states: given an initial state, the attributes of a goal state, and a set of state change operators, problem solving involves determination of a sequence of operators that transform the initial state to the goal state. The path of search may be controlled and constrained in several ways (e.g., breadth first, depth first, heuristic, dependency-directed backtracking).

2.3.1.4 Algorithmic approach

The algorithmic approach views design as a finite deterministic process. Cases where the entire design process is algorithmic are rare. However, parts of most design processes are algorithmic, especially where the emphasis is on numerical analysis and optimization. Optimization techniques apply where design problems can be formulated in the standard mathematical form of objective functions and constraint equations [12].

2.3.1.5 Axiomatic approach

“The key concepts of axiomatic design are: the existence of domains, the characteristic vectors within the domains that can be decomposed into hierarchies through zigzagging between the domains, and the design axioms. The design world of the axiomatic approach is made up of domains. There are four domains: the customer domain, the functional domain, the physical domain, and the process domain. Axioms are general principles or self-evident truths that cannot be derived or proven to be true except that there are no counter-examples or exceptions” [13] Suh identifies two axioms by examining the ubiquitous, common elements present in good product, process, or system designs [13]:

- Axiom 1: The Independence Axiom: The independence of Functional Requirements (FR) must be always maintained, where FRs are defined as the minimum number of independent requirements that characterize the design goals.
- Axiom 2: The Information Axiom: Among those designs that satisfy the Independence Axiom, the design that has the highest probability of success is the best design. The design with minimum information content has the highest probability of success.

2.3.2 Classes of design

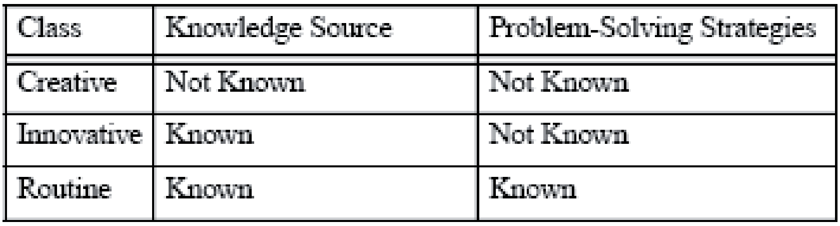
Many different classifications have been proposed for design, including: Preliminary, Conceptual, Functional, Innovative, Creative, Routine, Embodiment, Parametric, Detailed, Redesign, Non-routine, and Configuration [14].

Brown and Chandrasekaran have proposed the following three classes for design [15]:

- Class 1 Design (Creative Design): In this class of design neither the knowledge sources nor the problem-solving strategies are known in advance. An average designer in industry will rarely, if ever, do class 1 design. This type of design often leads to a major invention or completely new products.
- Class 2 Design (Innovative Design): What makes this type of design class 2 and not class 1 is that the knowledge sources can be identified in advance, but the problem solving strategies cannot. This type of design will require different types of problem solvers in cooperation and will certainly include some planning.
- Class 3 Design (Routine Design): The choices at each point in the design process may be simple, but that does not imply that the design process itself is simple, or that the components so designed must be simple. “We feel that a significant portion of design activity falls into this class” [14].

The main point, which is often overlooked, is summarized in Table 2.1, [14]:

Table 2.1. Classes of design



Class	Knowledge Source	Problem-Solving Strategies
Creative	Not Known	Not Known
Innovative	Known	Not Known
Routine	Known	Known

Note that in our consideration the creative design or the creative component of the design process has a different meaning; see next sections for more information (e.g., section 4.4).

Design processes can be classified along a different axis that describes what sorts of decisions are being made. “One end of the axis represents conceptual design and the other end represents parametric design. This axis shows the abstractness of the decisions being made, and reflects the notion that more constraints are added to the solution as the design activity progresses. For many design problems, the Conceptual-Parametric axis represents the flow of time during the design activity, with earlier decisions falling toward the left and later decisions falling toward the right” [14].

Consider the space that would result from the two orthogonal axes described above for classes of design, Figure 2.1.

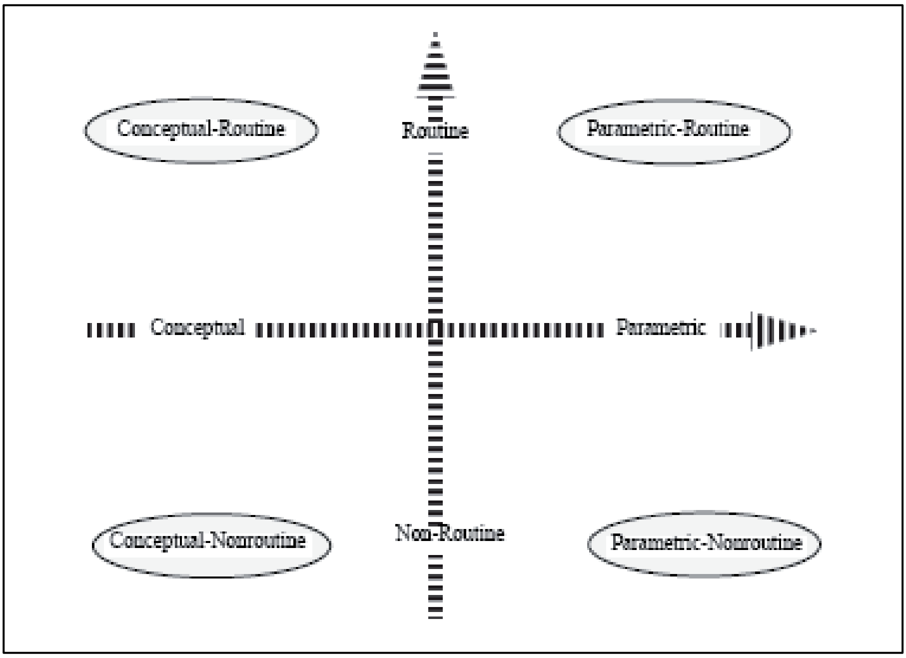


Fig. 2.1. Classification of Designs (According to [14])

A description of each of the four resultant categories can be found in [14].

2.3.3 Design methodology

The definition of “methodology” in Webster’s Dictionary is:

A body of methods, rules, and postulates employed by a discipline;

a particular procedure or set of procedures; the analysis of the principles or procedures of inquiry in a particular field.

A design methodology is a scheme for organizing reasoning steps and domain knowledge to construct a solution [16]. It provides both a conceptual framework for organizing design knowledge and a strategy for applying that knowledge [17].

"A design *methodology* is a prescription for a process intended to solve a specified design problem type. A design *method* is a procedure for implementing a step in a methodology." [7]

Design methodology is different from the *Design* itself. Design is primarily concerned with the question of '*what to design*' to satisfy some specified need. Design methodology, however, is primarily concerned with the question of '*how to design*'. Good methodologies allow us to better model, teach, and aid and automate finding solutions to 'what to design'. Design methodology is thus a vehicle for the evolution of design activity from an art or skill to a science. Insofar as design activity (as in industrial product design) is the natural testing ground for design methodology, we propose that the term *engineering design research* or *research in engineering design* be reserved for research in design methodology where the object is not a product but the knowledge of how to design products.

Design methodology is the answer to the question of "how to design". A design methodology provides methods for decomposing a design problem into sub-problems, ordering the design tasks, generating partial designs, composing more complete designs from partial designs, evaluating partial designs, and discovering and resolving conflicts.

A design methodology is a problem-solving model at an abstract level. A problem solving model is a scheme for organizing reasoning steps and domain knowledge to construct a solution to a problem. A problem-solving model provides both a conceptual framework for organizing knowledge and a strategy for applying that knowledge. [17]

Design methodology provides the design process knowledge that is the knowledge about how to carry out the design process to advance the design situation towards a solution [18].

2.3.3.1 Better design methodology

A better design methodology has at least the following properties:

- takes less time, causes fewer failures;
- produces better designs (better quality, simpler designs);
- works for a wide range of design requirements;
- integrates different disciplines;
- conducts design in a concurrent fashion;
- consumes less resources: time, money, expertise;
- requires less information (see Axiom 2 above).

2.3.4 Design methods

A *method* is defined in the Webster dictionary as follows:

- a procedure or process for attaining an object;
- a systematic procedure, technique, or mode of inquiry employed by or proper to a particular discipline or art;

- a systematic plan followed in presenting material for instruction;
- a way, technique, or process of or for doing something;
- a body of skills or techniques;
- a discipline that deals with the principles and techniques of scientific inquiry;
- orderly arrangement, development, or classification: plan;
- the habitual practice of orderliness and regularity.

"Design methods are any procedures, techniques, aids or 'tools' for designing. They represent a number of distinct kinds of activities that the designer might use and combine into an overall design process" [19]. All design methods have two principal features in common: One is that design methods formalize certain procedures of design; the other is that design methods externalize design thinking [19].

"Design goals have different design methods associated with them, which specify alternative ways to make decisions about the design parameters of the goal. These methods capture the knowledge about the possible values of properties of components, as well as knowledge about the behavior of components. The role of the design methods is then to generate partial designs." [11]

Cross [19, pp. 34-36] has categorized design methods to the following types: methods for exploring design situations, methods for searching for ideas, methods for exploring problem structure, and methods of evaluation.

Another classification of design methods is the following:

- algorithmic versus non-algorithmic
- theory-based versus experience-based
- iterative versus explicit
- analysis versus synthesis

A design method might be a combination of different types of methods such as a generator method plus an evaluation method. In addition, a design method might include different methods of the same type, such as different generator methods based on different technologies, etc.

2.3.4.1 Granularity of design methods

Smaller design methods have many benefits over large ones. There are some criteria for a small design method. The first is that a small design method is the one that makes fewer decisions. A decision is assigning a value to a design parameter. Additionally, a small design method uses less external information (e.g., fewer number of design parameters as input), and produces fewer numbers of design parameters as output.

If there are other methods that use the outputs of a large design method, they have to wait until the whole sequence of calculations for that method is finished. This prevents the other methods from having immediate access to those outputs. If any of the outputs violates a design constraint, it will not be revealed until the whole chain of parameters are produced by that method.

This principle of dividing larger methods into smaller ones is applicable in, e.g., the *multi-agent* engineering systems or in ADS systems. Different software agents exploit different smaller design methods toward a "larger" design goal. Therefore, the design methodology consisting of smaller design methods is better than the one consisting of fewer large design methods. Consequently, the multi-agent design system is better than a single-agent system. It is possible to deduce a more general conclusion from the above: *multidisciplinary design* is better than mono-disciplinary.

In this research, we use the benefits of this principle in order to improve the design process, and consequently, the output of that process.

2.3.4.2 Design approach

Within each design method we may define multiple design approaches. "There might be several kinds of artifacts, based on different technologies that can exhibit the same functionality" [11].

For example, one approach to designing a ventilation system might be to use a step ventilator speed controller, while another approach would be to use a thyristor regulated speed controller. Both approaches can be used to design a ventilator control schema.

A Design Method is a unit of procedural (operational) knowledge about how to produce values for some design parameters. A design method contains the knowledge about what approaches can be used to produce values for design parameters. These approaches might be ordered based on their priority over the others. A design approach might contain equations, look-up tables, heuristic rules, optimization algorithms, etc. that actually produce the values.

2.3.5 Design dependencies

Dependencies provide the knowledge for ordering design tasks, which comprises one part of a design methodology. They also provide a source of decomposition knowledge.

In addition to design dependencies that are extractable from design methods, there are other types of dependencies based on analytical or statistical studies. This type of dependency is mostly based on first principles and can provide valuable information for developing a design methodology.

"Lack of knowledge of dependencies, due to lack of a global model, is an underlying cause of conflict" [14]. Therefore, dependencies help to develop design methodologies that produce fewer conflicts.

The types of dependencies that two design methods might have are as follows:

- *Completely Independent:* Two design methods are completely independent if they share neither input nor output, Fig. 2.2 (a).
- *Loosely Independent:* Two design methods are loosely independent if they only share one or more of their inputs, Fig. 2.2 (b).

- *Reading Dependency:* Two design methods have reading dependency if at least one of them uses the other one's output as an input, Fig. 2.2 (c).
- *Conflict:* Two design methods have 'conflict' type of dependency if both produce the same output, Fig. 2.2 (d).

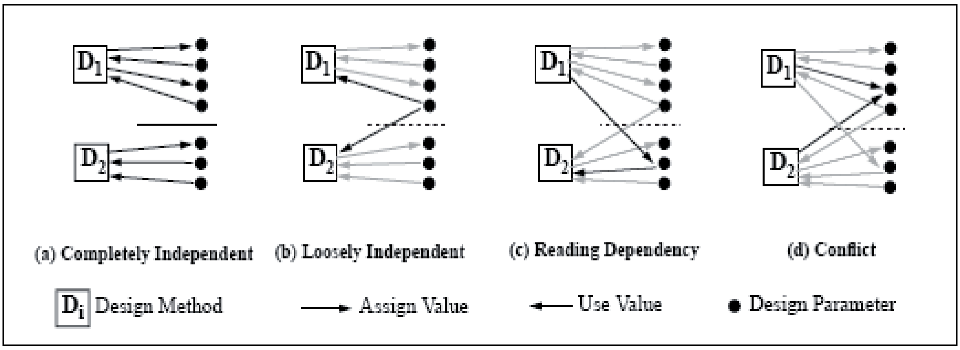


Fig. 2.2. The type of relationship between design methods

2.3.5.1 Sequencing design tasks

Eppinger proposes the following categorization of sequencing between design tasks [20]:

- Dependent Tasks: If task B simply requires the output of task A (or vice-versa), then the two tasks are dependent and are typically done in series, Fig. 2.3 (a).
- Independent Tasks: Tasks A and B are entirely independent if they can be performed simultaneously with no interaction between the designers, Fig. 2.3 (b).
- Interdependent (Coupled) Tasks: If task A needs information from task B, and also task B requires knowledge of task A's results, then the two tasks are interdependent, Fig. 2.3 (c).

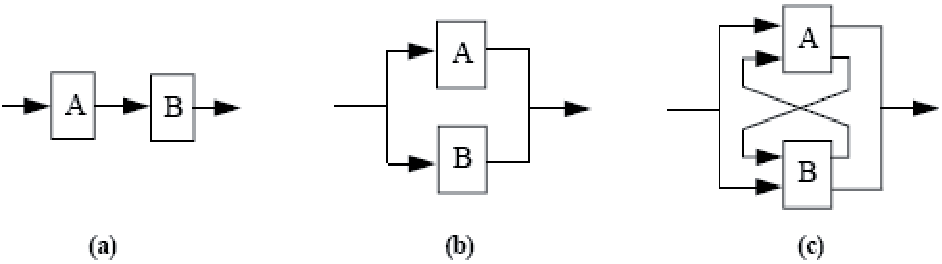


Fig. 2.3 Sequencing of design tasks [20]

2.3.6 Conflict resolution

Two types of conflicts can be identified in design: Domain Level versus Control Level conflicts. “Domain level conflicts concern conflicting recommendations about the actual form of the design, while the control level conflicts concern conflicting recommendations about the direction the design process should take in trying to create a design” [20].

According to Klein [21] the relevant literature on conflict resolution can be grouped into three categories:

- Development-Time Conflict Resolution: Systems of this type require that potential conflicts be “compiled” out of them by virtue of exhaustive investigation when they are developed.
- Knowledge-Poor Run-Time Conflict Resolution: In this approach conflicts are allowed to be asserted by the design agents as the system runs, and then resolved by some kind of conflict resolution component, e.g., backtracking.
- General Conflict Resolution: Work in this class comes closest to providing conflict resolution expertise with first-class status. Such systems provide categories of conflicts and have associate solutions or conflict resolution methods.

2.4 Multidisciplinary design

2.4.1 Introduction

Multidisciplinary designs are very complex processes that consume a lot of time, money, expertise, information and other resources. Complexity originates from the diversity of disciplines that each possesses a different point of view regarding the design problem. As a result, different disciplines adopt different and often contradictory goals and constraints, while they have to share resources such as budget, time, expertise, and information.

Although diversity is the source of complexity, it can be turned into a source of advantages. Having representation from different functional areas in multidisciplinary teams is beneficial to the design [22]. Diversity in disciplines brings multiple sources of knowledge, problem-solving techniques and expertise to the design process and also modularizes the design knowledge.

In this study the principles of multidisciplinary design are used, aiming at developing a software application. A computer program is developed based on different technologies and disciplines. These disciplines include engineering science, HVAC in our case, artificial intelligence, and computer science. There are numerous technologies and fields of expertise used in this research implementation, for instance, synergetics and complex systems, ObjectARX, neural networks, C++, MFC, just to name a few.

This research is focused on analyzing the problem of modeling the creative part of engineering design, the synergetic computer’s application opportunities in the HVAC design process and developing a software program based on this analysis. In this section, however, we consider multidisciplinary design as a general activity and give an overview from the traditional point of view, i.e., interaction between different participants (different knowledge expertise).

2.4.2 Integration and optimization

In order to take advantage of diversity in multidisciplinary design, different disciplines should collaborate with each other in adopting common goals, sharing resources, exchanging information, and resolving conflicts. “Engagement of different cooperative agents in the design problem solving can solve the design problem faster than either a single agent or the same group of agents working in isolation from each other. As a matter of fact, that cooperation leads to improvements in the performance of a group of individuals underlies the founding of the firm, the existence of scientific and professional communities, and the establishing of committees charged with solving particular problems” [23]. Such a collaboration strategy between different disciplines is called *integration of multidisciplinary design*.

“Engineering design should always be thought of as a multi-disciplinary activity. Indeed, it must include consideration of all disciplines: it is omnidisciplinary” [24]. Recently, there has been increasing recognition that multi-disciplinary design is important. A large amount of very good research has been focused on Multi-disciplinary Design Optimization (MDO) [24]. MDO tries to produce an effective product by recognizing and using appropriate combinations of parameters to be controlled and optimized by the designer.

MDO is based on mathematical modeling of the design problem in terms of objective functions and then their minimization. The problem is that a mathematical model for the design product is not available until the very end of the process, when the conceptual and embodiment design are complete. Additionally, for many cases a mathematical model cannot comprehensively include all design concerns, and ignores those characteristics that cannot be mathematically modeled.

However, in multidisciplinary design problems the values of design parameters may determine what design method will be employed, as methods may have applicability conditions. As different design methods may introduce different dependencies, dependency chains, and potentially problem decompositions, can be dynamically determined. This means that the sequencing of design tasks can also be dynamically determined.

2.4.3 Characteristics of multidisciplinary design

The following characteristics of multidisciplinary design contribute to the problem of producing better design. These are the most important barriers to integration of different disciplines.

2.4.3.1 Different points of view

The notion of ‘points of view’ is very important in multidisciplinary design. It is due to the difference in points of view of multiple disciplines that makes multidisciplinary design interesting and challenging [26].

2.4.3.2 Departmentalization of disciplines over time

Another challenge is that different disciplines have developed their own terminology and conceptualizing of the world separately from other disciplines because of the historical facts. As a consequence, there are not many techniques for understanding and collaboration among different disciplines. There could be different views among members of a team, but at the same time there could be well-defined terminologies and conceptualizing that allows the members to collaborate with each other in achieving a common goal.

Different disciplines conceptualize and represent their knowledge differently from the others. Boundaries are built around disciplines with special internal languages and no means for communicating with the outside world. As a consequence, it becomes difficult for the participants to communicate their points of view, let alone collaborate with each other or resolve their conflicts [26].

2.4.3.3 Built-in goals

Different disciplines tend to accumulate knowledge independently. As a result, they tend to have built-in goals that are often in conflict with global goals of the design. Ignoring the conflicts between local and global goals leaves the behavior of the system to the dynamics of self-design as determined by the structure of the system itself [27].

Below is an example of how the disciplines involved in ventilation system design adopt different goals:

- *mechanics:* keeps the system static (air ducts)
- *aerodynamics:* keeps the air velocity within a given range (diameter of the ducts increases)
- *economics:* keeps the price of the system low (diameter of the ducts should be minimum!)
- *thermodynamics:* avoids the appearance of the condensate
- *fire safety:* equips the system with fire safety features

As can be seen, the goals may also be in conflict with each other. The designer’s task is to resolve the conflicts by means of systematically optimizing the design values.

2.4.3.4 Focused expertise of the disciplines

The points of view of different disciplines are sharply limited due to highly focused expertise in their fields. As a result, integration techniques that are solely based on disciplinary knowledge become fragile, because they fail to apply as soon as the conditions change slightly [26].

2.4.3.5 Need for broad range of expertise

The required knowledge for doing multidisciplinary design is distributed among different fields of science and engineering so that no single person is able to possess all the required expertise. “Large-scale engineering projects typically involve up to 300 different specialty design firms, suppliers, and contractors. Therefore, many different types of professionals must interact and communicate with one another, which in many cases can result in conflicts” [28]. As a consequence, a broad range of expertise needs to be combined in order to develop an integrated design methodology.

2.4.3.6 Disciplinary design in big chunks

Disciplinary designs are processed in large segments that make integration very difficult because they hide valuable information that is necessary for integration (such as decisions that may lead to conflicts) from the rest of participants. Additionally, considering the iterative nature of design, it is costly and time consuming to repeat disciplinary designs in big chunks. This is because in every iteration the designers have to redo the big chunks of disciplinary design in their entirety.

2.4.3.7 Complexity of interactions

The interactions between different disciplines are complex because they are multifaceted, meaning that multiple disciplines might be interested in one parameter at the same time. Besides, the number of interactions is very large and they may change depending on the type of the problem.

2.4.3.8 Large number of iterations

Departmentalization increases the number of conflicts between disciplines, hence increasing the number of iterations required for finding a solution. A large number of iterations consumes more time and other resources in the design process.

2.4.3.9 Counter-intuitive behavior

“It has become clear that complex systems are counter-intuitive, that is they give indications that suggest corrective action which will often be ineffective or even adverse in its results” [27]. Multidisciplinary designs are a type of complex systems with counter-intuitive behavior. “Intuition fails to hold true when the constraints become active; it is then that the real interaction among design groups occurs” [29]. Therefore, integration in multidisciplinary design cannot be done based on intuitive approaches, and comprehensive studies are needed in order to develop solutions to the integration problem [26].

2.5 Computer Aided Design

Britannica Online gives the following definition for Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM):

> *Integration of design and manufacturing into a system under direct control of digital computers.*

CAD systems use a computer with terminals featuring video monitors and interactive graphics-input devices to design such things as machine parts, patterns for clothing, or integrated circuits. CAM systems use numerically controlled machine tools and high-performance programmable industrial robots. Drawings developed during the design process are converted directly into instructions for the production machines, thus optimizing consistency between design and finished product, and providing flexibility in altering machine operations. These two processes are sometimes grouped as CAE (computer-aided engineering).

Computer-aided design (CAD) is the use of a wide range of computer-based tools that assist engineers, architects and other design professionals in their design activities. It is the main geometry authoring tool within the Product Lifecycle Management process and involves both software and sometimes special-purpose hardware. Current packages range from 2D vector based drafting systems to 3D parametric surface and solid design modelers, and BIM (Building Information Modeling).

CAD is sometimes translated as "computer-assisted", "computer-aided drafting", or a similar phrase. Related acronyms are CADD, which stands for "computer-aided design and drafting", CAID for Computer-aided Industrial Design and CAAD, for "computer-aided architectural design". All these terms are essentially synonymous, but there are some subtle differences in meaning and application.

2.5.1 Capabilities

The capabilities of modern CAD systems include:

- Reuse of design components
- Ease of modification of designs and the production of multiple versions
- Automatic generation of standard components of the design
- Validation/verification of designs against specifications and design rules
- Simulation of designs without building a physical prototype
- Automated design of assemblies, which are collections of parts and/or other assemblies
- Output of engineering documentation, such as manufacturing drawings, and Bills of Materials

- Output of design data directly to manufacturing facilities
- Output directly to a Rapid Prototyping or Rapid Manufacture Machine for industrial prototypes

2.5.2 User Interface

The user interface (UI) is a fundamental part of any CAD application. A user interface can be defined as *"the parts of the program that link the user to the computer and enable him to control it"* [30, p. 13]. These 'parts' of the program include both hardware and software. The hardware parts are input devices such as tablet, mouse, joystick, light-pen and keyboard. The software part includes, for example, menu bars, icon menus, pallets and command languages.

The user interface determines the way in which user and computer interact. Learning time and program efficiency depend very much on a good user interface design. These are the goals that a well-designed user interface should fulfill [31]:

- SPEED OF LEARNING. How long a new user takes to achieve proficiency with a system.
- SPEED OF USE. How long experienced users require to perform a task.
- ERROR RATE. The number of user errors per interaction
- RAPID RECALL. The time a user needs to work again on the system after not having worked with it for some time.
- ATTRACTIVENESS. How attractive is the overall design for the user.

2.5.3 Building Information Modeling (BIM)

The short characteristics and the definition of BIM technology are outlined in this subsection. Logically, BIM may be considered as a successor of CAD technology.

Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition. (National BIM Standard - United States) [32]

Traditional building design was largely reliant upon two-dimensional drawings (plans, elevations, sections, etc.). Building information modeling extends this beyond 3-D, augmenting the three primary spatial dimensions (width, height and depth - X, Y and Z) with time as the fourth dimension and cost as the fifth. BIM therefore covers more than just geometry. It also covers spatial relationships, light analysis, geographic information, and quantities and properties of building components (for example manufacturers' details).

BIM involves representing a design as combinations of “objects” – vague and undefined, generic or product-specific, solid shapes or void-space oriented (like the shape of a room), that carry their geometry, relations and attributes. BIM design tools allow extraction of different views from a building model for drawing production and other uses. These different views are automatically consistent, being based on a single definition of each object instance [33]. BIM software also defines objects parametrically; that is, the objects are defined as parameters and relations to other objects, so that if a related object is amended, dependent ones will automatically also change [33]. Each model element can carry attributes for selecting and ordering them automatically, providing cost estimates as well as material tracking and ordering [33].

For the professionals involved in a project, BIM enables a virtual information model to be handed from the design team (architects, surveyors, civil, structural and building services engineers, etc.) to the main contractor and subcontractors and then on to the owner/operator; each professional adds discipline-specific data to the single shared model. This reduces information losses that traditionally occurred when a new team takes 'ownership' of the project, and provides more extensive information to owners of complex structures.

BIM is often associated with Industry Foundation Classes (IFCs) and aecXML - data structures for representing information. IFCs have been developed by buildingSMART (the former International Alliance for Interoperability), as a neutral, non-proprietary or open standard for sharing BIM data among different software applications (some proprietary data structures have been developed by CAD vendors incorporating BIM into their software).

Poor software interoperability has long been regarded as an obstacle to industry efficiency in general and to BIM adoption in particular. In August 2004 the US National Institute of Standards and Technology (NIST) issued a report [34] which conservatively estimated that $15.8 billion was lost annually by the U.S. capital facilities industry due to inadequate interoperability arising from "the highly fragmented nature of the industry, the industry’s continued paper-based business practices, a lack of standardization, and inconsistent technology adoption among stakeholders".

An early example of a nationally approved BIM standard is the AISC (American Institute of Steel Construction)-approved CIS/2 standard, a non-proprietary standard with its roots in the UK.

There have been attempts at creating a BIM for older, pre-existing facilities. They generally reference key metrics such as the Facility Condition Index (FCI). The validity of these models will need to be monitored over time, because trying to model a building constructed in, say 1927, requires numerous assumptions about design standards, building codes, construction methods, materials, etc., and therefore is far more complex than building a BIM at the time of initial design.

BIM is a relatively new technology in an industry typically slow to adopt change. Yet many early adopters are confident that BIM will grow to play an even more crucial role in building documentation. Proponents claim that BIM offers:

1. Improved visualization

2. Improved productivity due to easy retrieval of information
3. Increased coordination of construction documents
4. Embedding and linking of vital information such as vendors for specific materials, location of details and quantities required for estimation and tendering
5. Increased speed of delivery
6. Reduced costs

Green Building XML (gbXML) is an emerging schema, a subset of the Building Information Modeling efforts, focused on green building design and operation. gbXML is used as input in several energy simulation engines. Yet with the development of modern computer technology, a large number of building energy simulation tools are available on the market. When choosing which simulation tool to use in a project, the user must consider the tool's accuracy and reliability, considering the building information they have at hand, which will serve as input for the tool. Yezioro, Dong and Leite [35] developed an artificial intelligence approach towards assessing building performance simulation results and found that more detailed simulation tools have the best simulation performance in terms of heating and cooling electricity consumption within 3% of the mean absolute error.

US NIST Cloud Computing Security Architectures might be used by emerging enterprises to geo-spatially (geographically) connect individuals (corporations) with actionable building information data. These NIST architecture models define compute environments wherein private (securities available only to user), public (securities open to all in-network users), community (securities offered by a project leader) and hybrid (securities offered by a corporation) objects (folders and files) can be previewed, linked, opened, printed, exported, edited, saved, renamed, copied and deleted) and exchanged (uploaded to another network or downloaded from another network).

Chuck Eastman [33], for instance, notes that BIM tools are as different from CAD tools, in the same way that a slide rule is different from a computer, or as a set of toy soldiers is different from a battle-oriented computer game. BIM supports on-line simulation of a design, on-line simulation of construction - called 4D CAD, on-line simulation of a building’s operation, mechanically as well as the people organizations within it. The BIM processes provide better building products at lower costs to the owner. A growing number of case studies have shown the benefits to users who have used a building model to apply BIM technology. Building models and BIM technology will certainly become the standard representation and practice for construction within most of our lifetimes [33].

3 ARTIFICIAL INTELLIGENCE

3.1 Introduction

The background of artificial intelligence (AI) has been characterized by controversial opinions and diverse approaches. Despite the controversies, which have ranged from the basic definition of intelligence to questions about the moral and ethical aspects of pursuing AI, the technology continues to generate practical results. With increasing efforts in AI research, many of the prevailing arguments are being resolved with proven technical approaches. Synergetics and the technology of synergetic computer, the main subject of this research, when applied together with the cybernetic models, constitute one of the most promising branches of AI.

"Artificial intelligence" is a controversial name for a technology that promises much potential for improving human productivity. The phrase seems to challenge human pride in being the sole creation capable of possessing real intelligence.

It is being shown again and again that AI may hold the key to improving operational effectiveness in many areas of applications. Some observers have suggested changing the term *artificial intelligence* to a less controversial one such as intelligent applications (IA). This refers more to the way that computer and software are used innovatively to solve complex decision problems.

Natural intelligence involves the capability of humans to acquire knowledge, reason with the knowledge, and use it to solve problems effectively. By contrast, *artificial intelligence* is defined as the ability of a machine to use simulated knowledge in solving problems.

3.2 Origin of artificial intelligence

Many great philosophers and mathematicians, including Aristotle, Plato, Copernicus and Galileo, have sought the definition of intelligence over the ages. They attempted to explain the process of thought and understanding. The real key that started the quest for the simulation of intelligence did not occur, however, until the English philosopher Thomas Hobbes put forth an interesting concept in the 1650s. Hobbes believed that thinking consists of symbolic operations and that everything in life can be represented mathematically. These beliefs led directly to the notion that a machine capable of carrying out mathematical operations on symbols could imitate human thinking. This is the basic driving force behind the AI effort. For that reason, Hobbes is sometimes referred to as the grandfather of artificial intelligence.

While the term "artificial intelligence" was coined by John McCarthy relatively recently (1956), the idea had been considered centuries before. As early as 1637 Rene Descartes was conceptually exploring the ability of a machine to have intelligence when he said:

"For we can well imagine a machine so made that it utters words and even, in a few cases, words pertaining specifically to some actions that affect it physically. However, no such machine could ever arrange its words in various different ways so as to respond to the sense of whatever is said in its presence – as even the dullest people can do.”

Descartes believed that the mind and physical world are on parallel planes that cannot be equated. They are of different substances following entirely different rules and can thus not be successfully compared. The physical world (i.e., machines) cannot imitate the mind because there is no common reference point.

The 1800s saw advancement in the conceptualization of the computer. Charles Babbage, a British mathematician, laid the foundation for the construction of the computer, a machine defined as being capable of performing mathematical computations. In 1833, Babbage introduced an analytical engine. This computational machine incorporated two unprecedented ideas that were to become crucial elements in the modern computer. First, it had operations that were fully programmable, and second, it could contain conditional branches. Without these two abilities the power of today’s computer would be inconceivable. Due to a lack of financial support Babbage was never able to realize his dream of building the analytical engine. However, his dream was revived through the efforts of later researchers. Babbage’s basic concepts can be observed in the way that most computers operate today.

Another British mathematician, George Boole, worked on issues that were to become equally important. Boole formulated the laws of thought that set up rules of logic for representing thoughts. The rules contained only two-valued variables. By this, any variable in the logical operation could be in one of only two states: yes or no, true or false, all or nothing, 0 or 1, on or off, and so on. This was the birth of digital logic, a key component of the artificial intelligence effort.

In the early 1900s, Alfred North Whitehead and Bertrand Russell extended Boole’s logic to include mathematical operations. This not only led to the formulation of digital computers, but also made possible one of the first ties between computers and thought process.

However, there was still no acceptable way to construct such a computer. In 1938, Claude Shannon demonstrated that Boolean logic consisting of only two-variable states (e.g., on-off switching of circuits) can be used to perform logic operations. Based on this premise, ENIAC (Electronic Numerical Integrator and Computer) was built in 1946 at the University of Pennsylvania. ENIAC was a large-scale, fully operational electronic computer that signaled the beginning of the first generation of computers. It could perform calculations 1000 times faster than its electromechanical predecessors. It weighted 30 tons, stood two stories high, and occupied 135 square meters of floor space. Unlike today’s computers, which operate in binary codes (0s and 1s), ENIAC operated in decimal (0, 1, 2, …, 9) and required 10 vacuum tubes to represent one decimal digit. With over 18,000 vacuum tubes, ENIAC needed a great amount of electrical power, so much that it was said that it dimmed the lights in Philadelphia whenever it operated.

3.3 Human intelligence versus machine intelligence

Two of the leading mathematicians and computer enthusiasts between 1900 and 1950 were Alan Turing and John von Neumann. In 1945, von Neumann insisted that computers should not be built as glorified adding machines, with all their operations specified in advance. Rather, he suggested, computers should be built as general-purpose logic machines capable of executing a wide variety of programs. Such machines, von Neumann proclaimed, would be highly flexible and capable of being readily shifted from one task to another. They could react intelligently to the results of their calculations, could choose among alternatives, and could even play checkers or chess. This represented something unheard of at that time: a machine with built-in intelligence, able to operate on internal instructions.

Prior to von Neumann's concept, even the most complex mechanical devices had always been controlled from the outside, for example, by setting dials and knobs. Von Neumann did not invent the computer, but what he introduced was equally significant: computing by use of computer programs, the way it is done today. His work paved the way for what would later be called artificial intelligence in computers.

Alan Turing also made major contributions to the conceptualization of a machine that can be universally used for all problems based only on variable instructions fed into it. Turing's universal machine concept, along with von Neumann's concept of a storage area containing multiple instructions that can be accessed in any sequence, solidified the ideas needed to develop programmable computer. Thus, a machine was developed that could perform logical operations and could do them in varying orders by changing the set of instructions that were executed.

Due to the fact that operational machines were now being realized, questions about the "intelligence" of the machines began to surface. Turing's other contribution to the world of AI came in the area of defining what constitutes intelligence. In 1950, he designed the Turing test for determining the intelligence of the system. The test utilized the conversational interaction between three players to try to verify computer intelligence.

The test is conducted by having a person (the interrogator) in a room that contains only a computer terminal. In an adjoining room, hidden from view, a man (person A) and a woman (person B) are located with another computer terminal. The interrogator communicates with the couple in the other room by typing questions on the keyboard. The questions appear on the couple's computer screen and they respond by typing on their own keyboard. The interrogator can direct questions to either person A or person B, but without knowing which is the man and which is the woman.

The purpose of the test is to distinguish between the man and the woman merely by analyzing their responses. In the test only one of the people is obligated to give truthful responses. The other person deliberately attempts to fool and confuse the interrogator by giving responses that may lead to an incorrect guess. The second stage of the test is to substitute a computer for one of the two persons in the other room. Now the human is obligated to give truthful responses to the interrogator while the computer tries to fool the interrogator into thinking that it is human. Turing’s contention is that if the interrogator’s success rate in the human/computer version of the game is not better than his success rate in the human/human version, then the computer possesses “intelligence”. Turing’s test has served as a classical example for artificial intelligence proponents for many years.

By 1952 computer hardware had advanced so far that actual experiments in writing programs to imitate thought processes could be conducted. The team of Herbert Simon, Allen Newell and Cliff Shaw was organized to conduct such an experiment. They set out to establish what kinds of problems a computer could solve with the right programming. Proving theorems in symbolic logic such as those set forth by Whitehead and Russell in the early 1900s fit the concept of what they felt an intelligent computer should be able to handle.

It quickly became apparent that there was a need for a new higher-level computer language than was currently available. First, they needed a language that was more user-friendly and could take program instructions that are easily understood by a human programmer and automatically convert these into machine language that could be understood by the computer. Second, they needed a programming language that changed the way in which computer memory was allocated. All previous languages would pre-assign memory at the start of a program. The team found that the type of programs they were writing would require large amounts of memory and would function unpredictably.

To solve the problem, they developed a list processing language. This type of language would label each area of memory and then maintain a list of all available memory. As memory became available it would update the list and when more memory was needed it would allocate the amount necessary. This type of programming also allowed the programmer to be able to structure his or her data so that any information that was to be used for a particular problem could be easily accessed.

The end result of their effort was a program called Logic Theorist. This program had rules consisting of axioms already proved. When it was given a new logical expression, it would search through all the possible operations in an effort to discover a proof of the new expression. Instead of using a brute force search method, they pioneered the use of heuristics in the search method [36].

The Logic Theorist that they developed in 1955 was capable of solving 38 of 52 theorems that Whitehead and Russell had devised. It did them very quickly. What took Logic Theorist a matter of minutes would have taken years if it had been done by simple brute force on a computer. By comparison the steps that it went through to arrive at a proof to those that human subjects went through showed that it had achieved a remarkable imitation of the human thought process. This system is considered the first AI program.

4 GENERAL ANALYSIS OF SYNERGETICS AND SELF-ORGANIZATION THEORY

This section provides an introduction to the concepts of synergetics, self-organization and the problem of modeling the creative part of the engineering design process. A brief philosophical analysis of the related issues is also presented here.

4.1 Synergetics and self-organization

Synergetics (Greek: "working together") is an interdisciplinary field of research originated by Hermann Haken in 1969. Synergetics deals with material or immaterial systems, composed of, in general, many individual parts. It focuses its attention on the spontaneous, i.e., self-organized emergence of new qualities which may be structures, processes or functions. The basic question dealt with by synergetics is: are there general principles of self-organization irrespective of the nature of the individual parts of a system? In spite of the great variety of the individual parts, which may be atoms, molecules, neurons (nerve cells), up to individuals in a society, this question could be answered in the positive for large classes of systems, provided attention is focused on qualitative changes on macroscopic scales. Here "macroscopic scales" means spatial and temporal scales that are large compared to those of the elements. "Working together" may take place between parts of a system, between systems or even between scientific disciplines. See, e.g., [37] and [38] for a further reading.

A simple example of self-organization is a case of a fluid heated from below which may form patterns in the form of hexagons or rolls based in an upwelling of the fluid (Fig. 4.1).

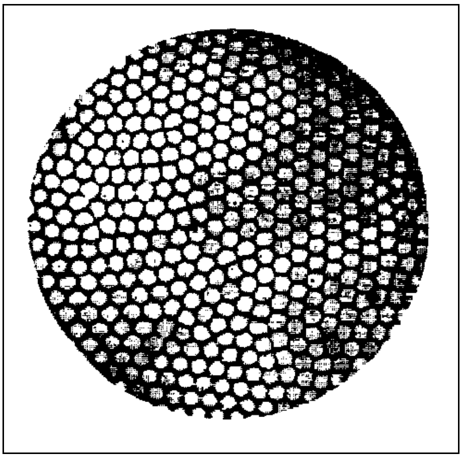


Fig. 4.1 Top view of a liquid in a circular vessel heated from below

When the temperature gradient exceeds a critical value, hexagonal cells are formed. In the middle of each cell the liquid rises, sinking back down at the edges of the hexagon. Further examples are the production of coherent light of lasers in physics, the macroscopic rings or spirals formed in chemical reactions in chemistry, and morphogenesis during the growth of plants and animals in biology.

The sunflower head, for instance, is composed of two counter rotating spirals, which must hit under a quite specific angle (Fig. 4.2).

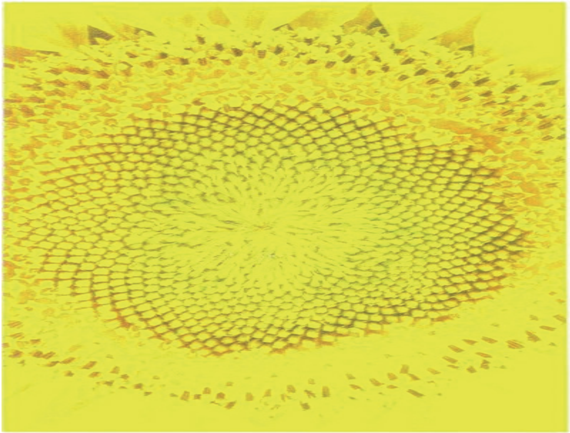


Fig. 4.2 The sunflower head is composed of two counter rotating spirals

The reason for this is not yet fully understood. Another example is in behavioral patterns which may range from the gaits of horses to the specific movements of human beings. In all of these cases the systems acquire their structures as a result of rather unspecific changes of their environment, for instance when the fluid is heated more strongly, or when the concentration of a chemical is changed and so on. In other words, the structures evolving in the system are not prescribed in a specific manner from the outside. More precisely, the system forms its new structure by self-organization.

4.2 Major researchers in the field

In this subsection the "key players" in the field of synergetics and the theory of self-organization are listed. They are from different fields of expertise: from the domains of mathematics and theoretical physics to those of philosophy and the philosophy of science. Below they are named, and a brief overview of their contribution is given.

4.2.1 Mathematical concepts (synergetics)

Hermann Haken (b. July 12, 1927) studied mathematics and physics at the Universities Halle (1946-1948) and Erlangen (1948-1950), receiving Ph.D. in mathematics (group theory) in 1951. He is professor emeritus and the chair for theoretical physics at the University of Stuttgart.

Hermann Haken has more than 30 awards and honorary degrees, including Max Born Prize and Medal by the British Institute of Physics and the German Physical Society (1976), Great Order of the Federal Republic of Germany with star (1986), Max Planck Medal (1990), Medal 2000: Outstanding People of the 20th Century (Cambridge, England). He is the member of Bavarian Academy of Science, German Academy of Natural Sciences Leopoldina, the Order "Pour le merite", Heidelberg Academy of Sciences, Academia Europaea (London), Academia Scientiarum et Artium Europaea (Salzburg), Lorenz-Oken-Medal of the Society of German Natural Scientists and Medical Doctors, and multiple "synergetics" societies.

Hermann Haken's major contributions are in the field of laser physics and nonlinear optics, solid state physics, statistical physics, group theory, and bifurcation theory. However, he is most known as the father of synergetics - the science of self-organization.

Dr. Haken is the author of 20 textbooks and monographs on atomic and molecular physics, quantum field theory, solid state physics, principles of brain function, information theory, computing, and synergetics. He is the editor of Springer Series in Synergetics, volumes 1-84.

A selection of some of his recent years' publications is given below:

Towards a Unifying Model of Neural Net Activity in the Visual Cortex. (to be published)

A Coherent Walk in Solid State Physics (to be published)

Some Thoughts on Modelling of Brain Function (to be published)

Synergetics of brain function (to be published)

Pattern Recognition and Synchronization in Pulse-Coupled Neural Networks (to be published)

Synergetics, Introduction to, in: Encyclopedia of Complexity and System Science, R. Meyers, Springer (2009)

Synergetics, Basic Concepts, in: Encyclopedia of Complexity and System Science, R. Meyers, Springer (2009)

Towards a Unifying Model of Neural Net Activity in the Visual Cortex, in: Cognitive Neurodynamics, Vol. 1, Springer (2007)

Handeln und Entscheiden in komplexen Systemen with G. Schiepek, in: 2. Symposium zur Gründung einer Deutsch-Japanischen Akademie für integrative Wissenschaft, J.H. Röll Verlag (2006)

Beyond Attractor Neural Networks for Pattern Recognition, in: Nonlinear Phenomena in complex Systems, Vol. 9, Nr. 2 (2006)

The interdependence between Shannonian and Semantic information, in: Complexity, Cognition and the City with J. Portugali, Springer (2006)

Synergetics on its Way to the Life Science, in: Complexus mundi: emergent patterns in nature, M.M. Novak, World Scientific Pub Co Inc (2006)

4.2.2 Philosophical study of the phenomenon of self-organization

Leo Näpinen has studied the concepts of organization and self-organization from the philosophical point of view. He has also laid the major philosophical foundation for and analyzed the concept of synergetics. He holds a Ph.D. from the Institute of Philosophy and Law at the Latvian Academy of Sciences (Riga) since 1984. The author of this dissertation had the honor to personally discuss with him the problems of modeling the creative part of the engineering design with synergetics tools, and also to listen to his graduate course on the self-organization topics/paradigm in modern science cognition.

Honours & Awards: 2008, Tallinn University of Technology's Letter of thanks.

Field of research: Culture and Society, Philosophy (philosophy of science, philosophy of chemistry, science studies; philosophical foundations of theories of self-organization; ideas of Ilya Prigogine).

Additional information: Supervised dissertations: Patrick Rang. BA dissertation "Ideas of self-organization in philosophy and science". Defended in 2000.

Directed (as a second supervisor) also Patrick Rang's Master’s degree dissertation.

Projects: collective project "Philosophy and science" (directed and realized at the Institute of International and Social Studies in 1995 - 1996); individual project "Paradigm of self-organization in exact sciences and its philosophical foundations" (realized in the Chair of Philosophy of the Tallinn University of Technology 1997 - 2000); collective project "Philosophical foundations of the non-classical science" (realized in the Chair of Philosophy of the Tallinn University of Technology 2001 - 2008).

Grants: three individual grants from the Estonian Science Fund, including the individual grant project no. 1814 "The relationship between man and the world in synergetic understanding of the world" (final report defended at the Institute of International and Social Studies in 1998).

Doctoral teaching: doctoral course "The paradigm of self-organization in modern sciences" (for doctoral students of the Tallinn University of Technology since 2000).

Has made reports on 34 international or national conferences (Beijing, 2005, 2007; Salzburg, 2006; Knoxville, 2005; Durham, 2004; San Francisco, 2004; Oviedo, 2003; Madrid-San Sebastian, 2000; Moscow, 1988, 1989, 1990; Kiev, 1989; Minsk, 1989, 1990; Riga, 2003, 2008; Vilnius, 2006; Tallinn, 1984, 1989, 1990, 1991, 2007, 2009, 2010; Tartu, 1982, 1983, 1984, 1994, 2001, 2003, 2007, 2008; Nancy, 2011).

Francis Paul Heylighen is a Belgian cyberneticist investigating the emergence and evolution of intelligent organization. He presently works as a research professor at the Vrije Universiteit Brussel, the Dutch-speaking Free

University of Brussels, where he directs the transdisciplinary research group on "Evolution, Complexity and Cognition" and the Global Brain Institute. He is best known for his contributions to the evolutionary-cybernetic worldview developed in the Principia Cybernetica Project, the modelling of the Internet as a Global brain, and the theories of memetics and self-organization.

His research focuses on the emergence and evolution of complex, intelligent organization. Applications include the origin of life, the development of multicellular organisms, knowledge, culture, and societies, and the impact of information and communication technologies on present and future social evolution.

Heylighen's scientific work covers an extremely wide range of subjects, exemplifying his intellectual curiosity and fundamentally transdisciplinary way of thinking. In addition to the topics mentioned above, his publications cover topics such as the foundations of quantum mechanics, the structure of space-time, hypermedia interfaces, the psychology of self-actualization and happiness, the market mechanism, formality and contextuality in language, causality, the measurement of social progress, the mechanism of stigmergy and its application to the web.

This impressive variety of work is held together by two basic principles. The relational principle notes that phenomena do not exist on their own, but only in relation (connection or distinction) to other phenomena. They thus only make sense as part of an encompassing network or system [39]. The evolutionary principle notes that variation through (re)combination of parts and natural selection of the fitter combinations results in ever more complex and adaptive systems. This principle is a direct application of Universal Darwinism, the idea that Darwinian mechanisms can be extended to virtually all disciplines and problem domains.

The two principles come together in Heylighen's concept of a distinction dynamics [40]. In his analysis, classical scientific methodology is based on given, unchanging distinctions between elements or states. Therefore, it is intrinsically unable to model creative change. Yet the evolutionary principle makes distinctions dynamic, explaining the creation and destruction of relations, distinctions and connections, and thus helping us to understand how and why complex organization emerges.

Moreover, any system must be adapted to its environment, which implies that it is able to react adequately to changes in that environment. This is the origin of mind or intelligence, as the system should be able to select the right actions for the given conditions. These "condition-action" relations are the basis of knowledge. As systems evolve, their adaptiveness tends to increase, and therefore also their knowledge or intelligence. Thus, the general trend of evolution is self-organization, or a spontaneous increase in intelligent organization.

Heylighen has published over 100 papers and a book. A selection of them is provided below:

Heylighen, Francis P. (2013) Self-organization in Communicating Groups: the emergence of coordination, shared references and collective intelligence. In A. Massip-Bonet & A. Bastardas-Boada (Eds.), Complexity perspectives on language, communication, and society, (pp. 117–150). Springer.

Heylighen, Francis P. (2011) Conceptions of a Global Brain: a historical review. Evolution: Cosmic, Biological, and Social, eds. Grinin, L. E., Carneiro, R. L., Korotayev A. V., Spier F. (pp. 274 – 289). Uchitel Publishing.

Heylighen, Francis P. (2008) Cultural evolution and memetics. Encyclopedia of Complexity and System Science, B. Meyers, ed.: Springer

Heylighen, Francis P. (2007) The Global Superorganism: an evolutionary-cybernetic model of the emerging network society. In: Social Evolution & History. Vol 6 No. 1,p. 58-119

Heylighen, Francis P. (2007) Accelerating Socio-Technological Evolution: from ephemeralization and stigmergy to the global brain. In: Globalization as an Evolutionary Process: Modeling Global Change. edited by George Modelski, Tessaleno Devezas, and William Thompson, London: Routledge. ISBN 978-0-415-77361-4. p. 286-335.

Heylighen, Francis P. (2001) The science of self-organization and adaptivity. The Encyclopedia of Life Support Systems 5 (3), 253-280

Heylighen, Francis P., Joslyn, C. (2001) Cybernetics and second order cybernetics.. Encyclopedia of physical science & technology 4, 155-170

Heylighen, Francis P. (1999) Collective Intelligence and its Implementation on the Web: algorithms to develop a collective mental map. Computational & Mathematical Organization Theory 5 (3), 253-280

Heylighen, Francis P. (1999) The growth of structural and functional complexity during evolution. The evolution of complexity, 17-44

Heylighen, Francis P. (1992) A cognitive□systemic reconstruction of Maslow's theory of self ‑ actualization. Behavioral Science 37 (1), 39-58

Ilya Romanovich Prigogine (25 January 1917 – 28 May 2003) was a Belgian physical chemist and Nobel Laureate noted for his work on dissipative structures, complex systems, and irreversibility.

Prigogine is best known for his definition of dissipative structures and their role in thermodynamic systems far from equilibrium, a discovery that won him the Nobel Prize in Chemistry in 1977. In summary, Ilya Prigogine discovered that importation and dissipation of energy into chemical systems could reverse the maximization of entropy rule imposed by the second law of thermodynamics [41]. Dissipative structure theory led to pioneering research in self-organizing systems, as well as philosophical inquiries into the formation of complexity on biological entities and the quest for a creative and irreversible role of time in the natural sciences. His work is seen by many as a bridge between natural sciences and social sciences. With professor Robert Herman, he also developed the basis of the two fluid model, a traffic model in traffic engineering for urban networks, in parallel to the two fluid model in Classical Statistical Mechanics. Prigogine's formal concept of self-organization was used also as a "complementary bridge" between General Systems Theory and Thermodynamics, conciliating the cloudiness of some important systems theory concepts with scientific rigor.

Work on unsolved problems in physics. In his later years, his work concentrated on the fundamental role of Indeterminism in nonlinear systems on both the classical and quantum level. Prigogine and coworkers proposed a Liouville space extension of quantum mechanics aimed to solving the arrow of time problem of thermodynamics and the measurement problem of quantum mechanics.[42] He also co-authored several books with Isabelle Stengers, including End of Certainty and La Nouvelle Alliance (Order out of Chaos).

The End of Certainty. In his 1997 book, The End of Certainty, Prigogine contends that determinism is no longer a viable scientific belief. "The more we know about our universe, the more difficult it becomes to believe in determinism." This is a major departure from the approach of Newton, Einstein and Schrödinger, all of whom expressed their theories in terms of deterministic equations. According to Prigogine, determinism loses its explanatory power in the face of irreversibility and instability. Prigogine traces the dispute over determinism back to Darwin, whose attempt to explain individual variability according to evolving populations inspired Ludwig Boltzmann to explain the behavior of gases in terms of populations of particles rather than individual particles. This led to the field of statistical mechanics and the realization that gases undergo irreversible processes. In deterministic physics, all processes are time-reversible, meaning that they can proceed backward as well as forward through time. As Prigogine explains, determinism is fundamentally a denial of the arrow of time. With no arrow of time, there is no longer a privileged moment known as the "present," which follows a determined "past" and precedes an undetermined "future." All of time is simply given, with the future as determined or undetermined as the past. With irreversibility, the arrow of time is reintroduced to physics. Prigogine notes numerous examples of irreversibility, including diffusion, radioactive decay, solar radiation, weather and the emergence and evolution of life. Like weather systems, organisms are unstable systems existing far from thermodynamic equilibrium. Instability resists standard deterministic explanation. Instead, due to sensitivity to initial conditions, unstable systems can only be explained statistically, that is, in terms of probability.

Prigogine asserts that Newtonian physics has now been "extended" three times, first with the use of the wave function in quantum mechanics, then with the introduction of space-time in general relativity, and finally with the recognition of indeterminism in the study of unstable systems.

A selection of his publications:

Prigogine, Ilya (1957). The Molecular Theory of Solutions. Amsterdam: North Holland Publishing Company.

Prigogine, Ilya (1961). Introduction to Thermodynamics of Irreversible Processes (Second ed.). New York: Interscience. OCLC 219682909.

Glansdorff, Paul; Prigogine, I. (1971). Thermodynamics Theory of Structure, Stability and Fluctuations. London: Wiley-Interscience.

Prigogine, Ilya; Herman, R. (1971). Kinetic Theory of Vehicular Traffic. New York: American Elsevier. ISBN 0-444-00082-8.

Prigogine, Ilya; Nicolis, G. (1977). Self-Organization in Non-Equilibrium Systems. Wiley. ISBN 0-471-02401-5.

Prigogine, Ilya (1980). From Being To Becoming. Freeman. ISBN 0-7167-1107-9.

Prigogine, Ilya; Stengers, Isabelle (1984). Order out of Chaos: Man's new dialogue with nature. Flamingo. ISBN 0-00-654115-1.

Prigogine, I. "The Behavior of Matter under Nonequilibrium Conditions: Fundamental Aspects and Applications in Energy-oriented Problems: Progress Report for Period September 1984 - November 1987", Department of Physics at the University of Texas-Austin, United States Department of Energy, (7 October 1987).

Prigogine, I. "The Behavior of Matter under Nonequilibrium Conditions: Fundamental Aspects and Applications: Progress Report, April 15, 1988 - April 14, 1989", Center for Studies in Statistical Mathematics at the University of Texas-Austin, United States Department of Energy, (January 1989).

Prigogine, I. "The Behavior of Matter under Nonequilibrium Conditions: Fundamental Aspects and Applications: Progress Report for Period August 15, 1989 - April 14, 1990", Center for Studies in Statistical Mechanics at the University of Texas-Austin, United States Department of Energy-Office of Energy Research (October 1989).

Nicolis, G.; Prigogine, I. (1989). Exploring complexity: An introduction. New York, NY: W. H. Freeman. ISBN 0-7167-1859-6.

Prigogine, I. "Time, Dynamics and Chaos: Integrating Poincare's 'Non-Integrable Systems'", Center for Studies in Statistical Mechanics and Complex Systems at the University of Texas-Austin, United States Department of Energy-Office of Energy Research, Commission of the European Communities (October 1990).

Prigogine, I. "The Behavior of Matter Under Nonequilibrium Conditions: Fundamental Aspects and Applications: Progress Report for Period April 15,1990 - April 14, 1991", Center for Studies in Statistical Mechanics and Complex Systems at the University of Texas-Austin, United States Department of Energy-Office of Energy Research (December 1990).

Prigogine, Ilya (1993). Chaotic Dynamics and Transport in Fluids and Plasmas: Research Trends in Physics Series. New York: American Institute of Physics. ISBN 0-88318-923-2.

Prigogine, Ilya (1997). End of Certainty. The Free Press. ISBN 0-684-83705-6.

Kondepudi, Dilip; Prigogine, Ilya (1998). Modern Thermodynamics: From Heat Engines to Dissipative Structures. Wiley. ISBN 978-0-471-97394-2.

Prigogine, Ilya (2002). Advances in Chemical Physics. New York: Wiley InterScience. ISBN 978-0-471-26431-6. http://www3.interscience.wiley.com/cgi-bin/bookhome/93517918/ProductInformation.html. Retrieved 2008-07-29.

Editor (with Stuart A. Rice) of the Advances in Chemical Physics book series published by John Wiley & Sons (presently over 140 volumes)

4.3 Literature review and discussion

As it was already mentioned, the major mathematical and theoretical physics background of this research has been laid down by H. Haken in several books and journal publications. The most important of them, published in Springer Verlag Complexity Series, are discussed below. Prior to this, an introduction to complexity is given.

Complex Systems are systems that comprise many interacting parts with the ability to generate a new quality of macroscopic collective behavior the manifestations of which are the spontaneous formation of distinctive temporal, spatial or functional structures. Models of such systems can be successfully mapped onto quite diverse “real-life” situations like the climate, the coherent emission of light from lasers, chemical reaction-diffusion systems, biological cellular networks, the dynamics of stock markets and of the internet, earthquake statistics and prediction, freeway traffic, the human brain, or the formation of opinions in social systems, to name just some of the popular applications. Although their scope and methodologies overlap somewhat, the following main concepts and tools can be distinguished: self-organization, nonlinear dynamics, synergetics, turbulence, dynamical systems, catastrophes, instabilities, stochastic processes, chaos, graphs and networks, cellular automata, adaptive systems, genetic algorithms and computational intelligence.

The two major book publication platforms of the Springer Complexity program are the monograph series “Understanding Complex Systems” focusing on the various applications of complexity, and the “Springer Series in Synergetics”, which is devoted to the quantitative theoretical and methodological foundations. In addition to the books in these two core series, the program also incorporates individual titles ranging from textbooks to major reference works.

In [43], for example, a contemporary insight on the problem of brain modeling is given. The human brain is the most complex system known of. It consists of about 100 billion neurons that interact in a highly complicated fashion with each other. Neurons are nonlinear elements. Most of them are able to produce trains of individual spikes, by which information between the neurons is exchanged. In addition, it is by now generally believed that correlations between spike trains play an important role in brain activity. Research on the human brain has become a truly interdisciplinary enterprise that no longer belongs to medicine, neurobiology and related fields alone. In fact, in the attempts to understand the functioning of the human brain, more and more concepts from physics, mathematics, computer science, mathematical biology and related fields are used. This list is by no means complete, but it reflects the aim of the book. It shows how concepts and mathematical tools of these fields allow for treating important aspects of the behavior of large networks of the building blocks of the brain, the neurons.

[44] is a thorough analysis of the concept of information and its interrelations with self-organization theory. One of the most fascinating fields of modern science is cognitive science which has become a meeting place of many disciplines ranging from mathematics over physics and computer science to psychology. Here, one of the important links between these fields is the concept of information which, however, appears in various disguises, be it as Shannon information or as semantic information (or as something still different). So far, meaning seemed to be exorcised from Shannon information, whereas meaning plays a central role in semantic (or as it is sometimes called "pragmatic") information. In the new chapter 13, it is shown, however, that there is an important interplay between Shannon and semantic information and that, in particular, the latter plays a decisive role in the fixation of Shannon information and, in cognitive processes, allows a drastic reduction of that information. A second, equally fascinating and rapidly developing field for mathematicians, computer scientists and physicists is quantum information and quantum computation. The inclusion of these topics is a must for any modern treatise dealing with information. It becomes more and more evident that the abstract concept of information is inseparably tied up with its realizations in the physical world.

In [45] a mathematical theory concerning the concept of synergetic computer is derived (from the basic equations of self-organization). This book presents a novel approach to neural nets and thus offers a genuine alternative to the hitherto known neuro-computers. This approach is based on the author's discovery of the profound analogy between pattern recognition and pattern formation in open systems far from equilibrium. Thus, the mathematical and conceptual tools of synergetics can be exploited, and the concept of the synergetic computer formulated. A complete and rigorous theory of pattern recognition and learning is presented. The resulting algorithm can be implemented on serial computers or realized by fully parallel nets whereby no spurious states occur. The recognition process is made invariant with respect to simultaneous translation, rotation, and scaling, and allows the recognition of complex scenes. Oscillations and hysteresis in the perception of ambiguous patterns are treated, as well as the recognition of movement patterns. A comparison between the recognition abilities of humans and the synergetic computer sheds new light on possible models of mental processes. The synergetic computer can also perform logical steps such as the XOR operation. The new edition includes a section on transformation properties of the equations of the synergetic computer and on the invariance properties of the order parameter equations. Further additions are a new section on stereopsis and recent developments in the use of pulse-coupled neural nets for pattern recognition.

[2] is a fundamental monograph explaining the basic principles of self-organization and its mathematical theory, i.e., synergetics. The spontaneous formation of well-organized structures out of germs or even out of chaos is one of the most fascinating phenomena and most challenging problems scientists are confronted with. Such phenomena are an experience of our daily life when we observe the growth of plants and animals. Thinking of much larger time scales, scientists are led into the problems of evolution, and, ultimately, of the origin of living matter. When we try to explain or understand in some sense these extremely complex biological phenomena, it is a natural question, whether processes of self-organization may be found in much simpler systems of the unanimated world. In recent years, it has become more and more evident that there exist numerous examples in physical and chemical systems where well-organized spatial, temporal, or spatio-temporal structures arise out of chaotic states. Furthermore, as in living organisms, the functioning of these systems can be maintained only by a flux of energy (and matter) through them. In contrast to man-made machines, which are devised to exhibit special structures and functionings, these structures develop spontaneously—they are self-organizing. It came as a surprise to many scientists that numerous such systems show striking similarities in their behavior when passing from the disordered to the ordered state. This strongly indicates that the functioning of such systems obeys the same basic principles. In this book it is intended to explain such basic principles and underlying conceptions and to present the mathematical tools to cope with them.

The question is not whether the creative component could be modeled on today’s computers using common, traditional computational methods, because the simple and obvious answer to this is that it could not. The question is rather if some new computational methods could be invented that allow for modeling the creative components of the design process. According to [46] it is possible by investigating the social-biological functionality of the human species. The authors note that deeper computational studies of biological and cultural phenomena are affecting the understanding of many aspects of computing itself and are altering the way in which human perceives computing proper.

The authors propose to model human intelligence by modeling individuals in a social context, interacting with each other. The important point is that while interacting, they have to change their thinking process not just the content. The authors oppose that to the software agents systems, which, according to them, are only capable of exchanging information while their own state remains unalterable. We argue here against the view of software agents paradigm, since agents may have dynamic structures, which are capable of learning and improving their functionality over time (they could be linked to artificial neural networks, for instance, or to other AI systems [47], they may be a part of Complex Adaptive System (CAS), as described in [48]). The social behavior greatly increases the ability of organisms to adapt. Minds arise from interactions with other minds.

An interesting insight into the problem has been given by Jorma Tuomaala [49], who considers creativity as the product of human subconscious mind and intuition. He proposes that the process of formalization of creativity is based on intuition and a combination of conscious and subconscious mind. The author does not call that exactly self-organization, but it is obvious that his methods of "capturing" creativity are tightly bound up with the notion of self-organization and synergetics (theory of self-organization).

In his book he attempts to handle the intuitive process and the possibilities of its controlled use based on his own experiences as a mechanical engineering designer. He examines in what ways one might use one's subconscious and conscious mind together in engineering design. He notes that this study is completely fictitious as far as subconscious mind is concerned. Nevertheless, he has tried to develop a model of it that may increase the understanding of the function and may even be useful in practice. He has also tried to expand the discussion to the area of literature and performing art, because he sees engineering design, as well, as a deeply human activity giving one all the possibilities to grow as a person (such approach is also articulated in the self-organization paradigm where someone creates arts, i.e., finds order in chaos). He has also tried to build a generally applicable method of intuitive creative work. According to J. Tuomaala, creativity is needed in the creation of anything new. He explains that the ideas produced intuitively usually seem to be, despite all arguments, of the best quality. This arouses the question if the use of intuition along with systematics should mainly be seeked in (mechanical) engineering and if

a) the time needed to find a solution can then still be defined;
b) an intuitive solution can be considered as optimum;
c) the time used for solving the problem affects quality:
d) it is possible to lead the intuitive process.

Finally, it could be asked if intuition could prove a useful instrument in creative engineering design, if it were possible to model it using some computational algorithm.

Another attempt to formalize the creative part of the design process was done by MG Taylor Corporation and explained by Bryan Coffman in 1996, see [50]. Very interestingly, MG Taylor Corporation seems to have been dealing with creative design formalization since 1982, but until today no working examples of an autonomous engineering design system have been created (i.e., a system that were capable of modeling the creative part as well), although there have been attempts to create a conceptual framework and general methodology to aid the design of complex adaptive systems using the principles of self-organization [48].

Ongoing research in the AI domain and in the field of general technology shows that traditional methods of solving engineering problems based on formal logic and systematical approach are shifting toward the new unrevealed, presently undocumented features of human mind and intelligence (closer to the characteristics of self-organization?). There are neural networks, which try to copy the functionality of biological brain cells – neurons, fuzzy logic and modeling (for a contemporary research on fuzzy dynamic systems see, e.g., [51]), expert systems, evolutionary programming/computing, knowledge-based systems, swarm and genetic algorithms, etc. In that sense, this can be compared to the paradigm shifts that occurred in the 20th century when the new age science transformed from its classical period (Galileo-Newton physics) to non-classical (quantum mechanics, static laws and systems) and post non-classical (open non-linear systems, etc.) forms.

4.4 Modeling of the creative part of the engineering design process using synergetics approach

In this section, the problem of modeling the creative part of the engineering design process is analyzed from the synergetics perspective. The analysis is performed from a general point of view. The characteristics of the creative tasks of engineering design are defined and the novel notion of the Autonomous Design System (ADS) is introduced. ADS is considered as an advanced CAD system that has the Artificial Intelligence (AI) functionality and particularly the functionality to deal with creative components of the engineering design process. A couple of cybernetic models which can be further optimized by the methods of synergetics are proposed (see also in section 6). The presented discussion forms theoretical foundations and philosophical motivation for ongoing research in this field. This work constitutes the introduction to the extension of the author's original research in the field of CAD systems optimization.

Engineering design can be viewed as an articulate process composed of phases, where each phase represents a combinatorial action on the parts the composite object is comprised of. To realize an object meeting the desired market requirements, engineering designers have to deal with different kinds of knowledge about objects, or ontological knowledge (which is often represented in a declarative form), and "dynamic" knowledge about processes (which is often represented in "procedural terms") [52].

Synergetics can be considered as one of the modern, most promising research programs. It is oriented towards the search for common patterns of evolution and self-organization of complex systems of any kind, regardless of the concrete nature of their elements or subsystems.

These parts of the design process that are numerically analyzable can be modeled numerically. The numeric model can then be further improved and optimized. All the benefits and achievements of the digital revolution, including Artificial Intelligence (AI), can be used to automate the process of engineering design.

On the other hand, the design process consists of the "creative" part that is not numerically describable by traditional numerical algorithms, at least not yet.

As today’s traditional CAD (Computer Aided Design) systems are based on numerical (digital) computational machines (i.e., personal computers), there are no ready standard solutions for automating these creative parts of the design process. Note that “routine” parts of the engineering design process can be modeled, divided in parts and automated relatively simply, and there are several examples on the market today.

On the contrary, there is only one stage in the true model of the creative process. At the simplest level, creativity is the act of being and doing folded into a state of flow called life. Naturally, all of our time is spent in a state of flow, despite claims to the contrary in the popular press. Even when analyzing a problem, we are doing something, and employing tools of some sort. A rapidly evolving picture of what we want to do is simultaneously embraced, and that unfolds just before we do it. What the popular press describes as a state of flow, occurs when the execution of the creative process becomes jubilant, and consequently high performance.

The creative process is divided into pieces in an effort to understand and picture the complexity of the entire process. Yet the pieces are not actually executed in some sort of lock step fashion. It is convenient and instructive to perceive that creativity has certain stages and that everyone can emotionally, physically and mentally relate to these stages, but to hold any model of the creative process as a precise description of creativity, and to force others to adhere strictly to its application is foolish. Stuart Kauffman uses an expression to describe the difficulty of modeling any living system: "the algorithm is incompressible." In other words, there is no shorter method, routine or program to describe life or living systems than life or the living system itself. Models are representations of reality, but they are not the reality itself. There is no algorithm or equation that creativity can be forced into that is shorter than the creative act itself [50]. However, there is a hope that the model can still be approximated to its representation by introducing (and using) the emergent AI technology, its tools and algorithms.

The creative parts of the design process are characterized by the higher intelligence needed to deal with them. Therefore, if the aim is to model that part of engineering design, far more powerful AI technologies are needed than those existing today.

It may even prove impossible to model/automate these creative components by means of today’s computers. Some new revolutionary technology may be needed to do that. However, there is a belief that some of these components may still be approximated (to some degree) by mathematical methods that are readily available now (by their improvement) or by the newest methods that have emerged recently, or those, still under development at the moment. The AI tools and technologies in cooperation with synergetics, for instance, may help us to achieve that goal.

Synergetics (in the meaning of H. Haken’s school of thought) provides mathematical tools to cope with the self-organization phenomenon. These tools are based on the combination of the differential equations theory and the stochastic modeling. The principles of synergetics can be used in a great variety of scientific disciplines ranging from theoretical physics to musical [53] and social sciences. First, synergetics is considered from the general, epistemological point of view.

Synergetics reflects the surrounding natural systems in a sense of soft or coherent action principle. The natural phenomena develop along the evolutionary paths according to evolutionary principles. There are no hard or external actions which can successfully drive (manage) the complex natural system in its development pathway. From that can be learnt how to arrange the activities in order to achieve optimal results. It turns out that managing influence must not be energetic, but rightly topologically organized according to the general and universal laws of self-organization. There must be certain organization of actions. It is the topological configuration, the symmetric "architecture" that is important, not the intensity of the influence. Synergetics defines how it is possible to multiply reduce time and required efforts to generate, by a resonant influence, the desirable and, what is no less important, feasible structures in a complex system. These principles are equally applicable to the case of modeling the complex parts of engineering design process.

Leo Näpinen, for example, stresses the importance of participative constructive activity as follows [54]. "The cosmos is filled with the creativity of the process of endless transformations, and human creativity derives from the creativity of the cosmos itself. Human constructive activity is justified indeed — but not a dominative construction. Instead, it has to be participative construction."

The routine parts (numerically describable) of the engineering design process can be successfully modeled with the help of cybernetics. It is really the art of combinatorial manipulation and constructing (constructive rationality) to fulfill the goal, using the already known or novice technology, IT in this case. As it is based on cybernetics, it boils down to organizational theories, contrary to the self-organization paradigm, and therefore is not the subject of interest of this work.

From the concept point of view, the notions of organization and self-organization can be described as follows. The concept of organization denotes the process that leads to the rise of goal-oriented structures due to conscious human goal-directed action or some external ordering influence, and the concept of self-organization denotes the process that leads to the rise of goal-oriented structures beyond conscious human goal directed action or some external ordering influence. Although the term "self-organization" is widely used (and more appropriate) in the field of synergetics, it has been utilized in cybernetics as well. In cybernetics, however, it has a different meaning (from the philosophical point of view). In cybernetics and systems engineering self-organization is understood as an effect of an external ordering factor (e.g., self-organizing map in [55]). In synergetics self-organization is understood as the rise of harmonious behavior distinguished from man's intervention and from external (with regard to the system) ordering factors. External factors (e.g., strong non-equilibrium) are indispensable for self-organization, but only as conditions, not as ordering forces.

Hopefully, it is possible to imitate the creativity (at least to some degree) by means of synergetic modeling. A question arises if modeling the creative part of the (engineering) design process is also possible. To answer this question, the synergetic approach must be analyzed and compared with traditional information technology modeling instruments, e.g., with cybernetics.

In cybernetics, as well as in synergetics, the objective processes are modeled in order to control them. The cybernetic models make it possible for man to strive for the desirable results using the program created by him. The synergetic models take into account that the programs form in the course of self-organization [56].

All exact sciences (and also the traditional scientific cognition) are model-based. They are exact only within that model. Therefore, it is not possible to explore or predict or study the real world adequately by means of “exact” sciences by definition. Exact sciences can be used to explore models. CAD systems and ADS (Autonomous Design System, ADS is considered to be a further development of a conventional CAD system, which takes into account the creative component of the design process) frameworks are examples of design system models. Both cybernetics and synergetics are exact sciences as well. So these disciplines can only be used for the development and research of models of the underlying real world’s phenomena and not for the investigation of the real world itself. It must be underlined that in exact sciences the approach to the interaction between organization (management) and self-organization does not go (and due to the specificity of exact sciences must not go) beyond certain boundaries.

The limits mean that exact sciences in their models of influence upon self-organization give only such recommendations according to which the future state of an object of management is given from the outside. Exact sciences do not make any contribution to the opening of the creative potential of the elements of the system [56]. So standalone synergetic methods (a kind of exact science) cannot be used to explore the creative potential of the system (and self-organization). As synergetics is an exact science and is based on mathematics, it has known limitations in its capability to explore the real world. Yet it can still be used to create better models of real life systems, but not to understand these systems completely. On the other hand, building more adequate models of the environment leads to a better understanding of the environment itself. Therefore, this may lead to a new level of understanding, to help form a new paradigm and from within it - to model even more precisely, closer to the real world.

Synergetics models better than cybernetics the processes of the real world, which is ultimately a self-organizing system. So principles of synergetics can be used, in conjunction with traditional computing technology, to model some aspects of the real systems. It is worth showing how creativity is understood in synergetics. The meaning of the word creative is the unpredictability and unavoidability of the unknown. The creative chaos is a field of unknown and unpredictable chances. The meaning of the word is closely related to such concepts as the non-equilibrium condition and conditions close to equilibrium.

Synergetics also accentuates one necessary condition of self-organization: the order arises from chaos only under the condition of strong non-equilibrium. It is necessary to strictly distinguish the chaos under conditions close to equilibrium (in which, generally speaking, self-organized structures can only decompose) from the chaos under strongly non-equilibrium conditions (in which composing of structures through self-organization can take place) [56]. The former type of chaos is non-creative, the latter is creative.

In engineering design process theory the meaning of the word "creative" (and "creativity") is slightly different. Here the word "creative" denotes a non-routine part of the design process. Contrary to the routine procedures, where inputs and outputs of the system are known or predictable, *the creative part of the design process* deals with output data that is mainly unknown, although the field of possibilities (possible outputs) is generally defined. This is true in ordinary design scenarios where the ultimate goal of the design procedure is known. When the output data of the system is completely undefined and unknown, then the system generates some new design information (i.e., invention mechanism). Note that in the majority of cases the input data is defined (both in ordinary design scenarios and in invention apparatus). The modeling of the technical invention processes is even more complicated (if not impossible) than imitating the creative part of the conventional design process (i.e., the process where the field of the possibilities of the output information is defined). There is a hope that using the methods of synergetics and the philosophy of self-organizing systems the problems of modeling creative design can be addressed in a more precise and better manner. The new science which accepts creativity based on chance and irreversibility in nature, and considers the fundamental indeterminacy of the whole history of nature and of human society should evolve to acknowledge the potential of this approach.

Basically, a model can be considered as an idealized version of the real system. The model is always simpler and more primitive than the real system. Nowadays, the traditional tool for creating engineering design models is the Computer Aided Design (CAD) system. For creating a new CAD system, CAD programming is used. Thus, CAD programming is essentially a construction of a model (computer program) for a model (CAD application) of a model (engineering design, project) of the system (e.g., engineering installation). Such cascading of models occurs, e.g., when programming under some existing CAD platform, for instance, under AutoCAD. On this level of abstraction, the model itself is very precise (it is nested into the surrounding model, etc.) and perfectly describable by mathematics.

The aim is to try to add to this model the properties/specifications of the *self-organizing* systems' behaviors. The author does not really think that the model will be capable of substituting the engineer completely in the process of producing creative design. Nevertheless, there is a hope that the model built in the spirit of synergetics may facilitate the emergence of the elements of creativity in engineering design in which the human participates as well. It is likely that these models, in cooperation with the operator (engineer), can function more effectively in creating new designs. Moreover, the engineer and the model in conjunction virtually constitute a self-organizing system and the number of degrees of freedom in that resultant system is bigger than in each of its separate parts. Thus, the probability of emergence of interesting and usable design scenarios is larger. It must be stressed here that it is impossible to predict whether the useful design cases ever emerge as a result of using the synergetic model. It is also impossible to specify when and what kind of outputs will be created by the model. This would be a kind of system with a rather probabilistic behavior, therefore, in theory, it could even downgrade (to some degree) the developments of the design process, but, nevertheless, even in that case it will still be operating according to the principles of self-organization. It is even possible that a wrong output (as it may seem at the time) will be considered a better one over time. The big question is how to compound such a system that it could be *maximum self-organizing*, because it still has to be constructed, i.e., the system does not emerge as a result of self-organization in principle.

This raises a question if passive behavior were wiser – not to construct anything, but be inactive and wait till the systems arise by themselves. Another option could be to create some very simple systems with minimum "dominative construction" attributes and let the general outer self-organizing world finish the model according to its intrinsic implicit laws (as known from synergetics, it seems that namely simpler laws drive complicated phenomena).

4.5 Philosophical outlook on the subject

In this section some concluding philosophical remarks on the subject are given. This section is a logical continuation of the preceding discussion, as the subject researched is closely connected with both the problem of modeling creativity (and engineering creativity) as well as the philosophical consideration of creativity and self-organization.

In his book "All Life is Problem Solving" [57], K. Popper emphasizes the value of free opinion formation, which, in conjunction with simple clear language (analogy to simple "laws" that drive complex phenomena?), may be considered as a characteristic of self-organization (in social context). "Why does simplicity of language matter so much to Enlightenment thinkers? Because the true Enlightenment thinker, the true rationalist, never wants to talk anyone into anything. No, he does not even want to convince: all the time he is aware that he may be wrong. Above all, he values the intellectual independence of others too highly to want to convince them of important matters. He would much rather invite contradiction, preferably in the form of rational disciplined criticism. He seeks not to convince but to arouse—to challenge others to form free opinions. Free opinion formation is precious to him: not only because this brings us all closer to the truth, but also because he respects free opinion formation as such. He respects it even when he considers the opinion so formed to be fundamentally wrong. “

It is possible to perform experiments with some candidate synergetic models in order to select the most appropriate one (it must be remembered, however, that in experimental situations, the determinants of the organization or process are contingent on the subject, i.e., the experimenter; for a detailed description of the philosophical interpretation of human constructive activity (based on Aristotle’s four causes that in unity form the philosophical category of self-organization) see, e.g., [58]). Again, what are the criteria for the selection and are these criteria adequate enough? Self-organization is impossible to describe adequately in detail, how is it then possible to define adequately its characteristics for the selection criteria?

The properties of self-organizing systems in general could be discovered (if ever) using the methods of historical cognition (although in relatively small isolated groups/systems it is possible to use methods of classical exact sciences for that purpose). These methods are closely connected with the notion of time. The analysis of the historical phenomena is not possible without any knowledge on their past, as they develop in the process of irreversible evolution. Thus the future is unknown and unpredictable.

From the philosophical point of view, it seems that the hypothetical-deductive knowledge in principle cannot explain historical phenomena. It is still possible to use the hypothetical-deductive theories of exact sciences in the modeling of some aspects of historical phenomena. It must be regarded as the cooperation of reconstructive (historical, descriptive-theoretical) and constructive (hypothetical-deductive) approaches but not as an attempt to replace one by the other [56]. It is also possible to use the reconstructive approach in modeling some parts of the cybernetic systems. This can also be called cooperation. From the perspective of modeling the creative part of the engineering design process, this type of cooperation seems to be more appropriate. The irreversible “arrow of time” (i.e., the reconstructive approach) has to be introduced into cybernetic models (e.g., by cooperation techniques; in this case it will be a rather synergetic model; this approach may need different computational algorithms or a different paradigm not invented yet) or, at least, connecting the computational system based on cybernetics with the synergetic model has to be attempted. This should improve the initial models’ architecture, their functionality and bring these models closer to the real life phenomena. The question is whether a synergetic system that has characteristics of irreversible “arrow of time” could be modeled by means of traditional computer (i.e., serial computational machine), which is based on reversibility principles.

As I. Prigogine showed in the language of mathematized science, the real situations are orientated in time, the states and laws are closely connected with one another, and the initial conditions of the system emerge as a result of its previous evolution [56]. For a critical analysis of I. Prigogine's attempts to use exact science (based on mathematics) for understanding the natural-historical phenomena see, e.g., [59]).

I. Prigogine has written that irreversibility can no longer be identified by mere appearance that would disappear if the observer/human had perfect knowledge. Instead, it leads to coherence, to effects that encompass billions and billions of particles. He noted that humans are actually the children of the arrow of time, of evolution, not its progenitors.

It seems that most of the self-organizing systems have their inner goal, towards which they are constantly evolving, but making this goal clear to humans is a very hard (if not impossible) task. The essential characteristic of a self-organizing system is its autonomous purposive behavior. The characteristics of a self-organizing system cannot be constructed according to an external purpose (i.e., from the outside of the system). "Self-organizing systems … have their own (i.e., autonomous) goals…" [60].

Stephen J. Gould in his works (e.g., [61]) stresses the importance of the historical character of life. As mentioned earlier, different life forms may be considered as an example of self-organizing systems. The important property of self-organizing systems is their historical character of evolution. He also notes in his definitions that the human species is not necessarily the highest expression of life on our planet, and therefore, to put it objectively, humans should not expect their cognition mechanism to be the supreme and right one. He notes that in order to understand the events and generalities of life's pathway, it is necessary to go beyond principles of evolutionary theory to a paleontological examination of the contingent pattern of life's history on our planet, which is the single actualized version among millions of plausible alternatives that happened not to occur. Such a view on the history of life is highly contrary both to conventional deterministic models of Western science and to the deepest social traditions and psychological hopes of Western cultures for a history culminating in humans as life's highest expression and intended planetary steward. Stuart A. Kauffman [62], [63], one of the leading figures in the study of self-organization and complexity nowadays, has pointed out that the evolution of the whole world appears to be a combination of selection and self-organization. Thus, the understanding of evolution only by natural selection is incomplete.

Assuming that the self-organization paradigm is true, the human mind, among all other known systems in our world, must function according to it as well. From that it can be concluded that the human mind's creativity has to work in accordance with synergetic models as follows. In synergetics, models should ideally reflect the transitions between different qualitative states by positive feedback (a system exhibiting positive feedback, in response to perturbation, acts to increase the magnitude of the perturbation). These transitions are possible only if the influence of external environment on the system is so changeable that amplification of the fluctuations may cause the system to move so far away from the equilibrium that it cannot return to the former state, and new possibilities of development (i.e., creativity?) may appear.

These qualitative transitions are simultaneously both determined and undetermined. The fundamental objective indetermination lies in their basis. It is not determined into which qualitative state from some (or many) possibilities the system will actually go after the selection. Only the field of possibilities is determined. [48] These model characteristics should comply with the theory of self-organization or synergetics that was introduced and thoroughly developed by Hermann Haken.

It has to be remembered (while constructing, modeling some natural phenomena) that the scientist himself and his activity together with its products can now be treated as part of the modeling process, of nature. Nature is understandable as a living being who, thanks to the conceptual and technical idealization, is indeed predictable, even transformable and manipulable, but only locally, partially and relatively [64].

As Zwierlein notes [65], no question can be approached from a neutral or objective standpoint. Every questioning grows out of a tradition and its underlying pre-understanding that opens the space of possible answers. To grow and to expand the horizons does not mean to surpass the condition of having a background of pre-understanding in principle. Human will always operate within the framework the "Lebenswelt" provides. It is definitely *impossible* that human understanding will ever be *neutral* or *objective* or *complete*.

We agree with Zwierlein at that point, but does it mean that mankind will never be capable of understanding the surrounding integral world completely? Should the problem of understanding be approached from another perspective that is not based on traditional cognition (i.e., logical, cumulative way of thinking) or, as Zwierlein and Kant note, is ultimately rooted in anthropology (scientific understanding)? For instance, the Taoist approach (meditative practices, etc.) can be tried, but, although the so-called enlightened adepts are reputed to realize (specifically, to realize not to understand – in Taoist practices these words have different meanings) the true meaning of life and integral world (Lebenswelt?), the question still remains whether this realization is the true one. How can this be checked against the truth and what is the truth itself? Moreover, every Zen master has still got a slightly different realization (understanding) of things, although they insist that the core understanding is the same and one. All these points just amplify doubts and uncertainties about human cognition mechanisms.

Hayek [66] suggests that while known biological species have adapted to fixed and rather limited environmental "niches" beyond which they cannot exist, human and some animals, rats, for instance, have been capable to adapt almost everywhere on the planet. Hardly could this have been achieved by individual adaptability alone. The author points out that our civilization owes the overall success of the human species to the social ability and cooperation of individuals (which derives from self-organization?).

In conclusion of the above, we present a rather philosophical outlook on the problem. What would be the directions for further evolution of the mankind? Some think that acceleration of the development of technocracy (in the meaning of hypothetic-constructive-deductive way of cognition, opposed to self-organization thinking) is an answer. This way of thinking really seems to dominate in today's scientific and domestic (everyday) domains. The alternative understanding, on the contrary, respects the appearances of Mother Nature in all aspects of reality and promotes the soft management and participative construction in accordance with the global self-organization. As Leo Näpinen notes [67], the big question is whether the human mind (spirit) will develop belonging to the general determinacy (comprising the creative chance) derived from the integrity of the world, i.e., belonging to the self-organization; or the human mind will restrict itself to splitting the integral world into pieces and manipulating with them, i.e., will restrict itself to organizing the organizations.

Going back to creativity modeling, we should take into consideration that in order to model the human creative process, the analogy principle can be used only to a certain degree. This means that there is no precise information on how this process occurs in reality (e.g., dreams, emotions etc). Modern science provides only some possibly true facts (knowledge) that can be relied on. Therefore, it is possible to get only an approximation of the real system (artificial human creativity). Another point is that the aim is not an examination of how this creativity really works (i.e., the objective of the research is not to ultimately expose the mechanism of creativity, but to build the mathematical/synergetic model that is relatively creative (for a mathematical model's "creativeness", in the sense of engineering design creativity, see above)). Instead the goal is to model this phenomenon and use it in practical applications, which may help to do better design (engineering) work and automate the engineering design process. On the other hand, in order to model the system successfully, it is useful to know how the real system works, at least on the conceptual level.

In addition, in all of these possible implementation examples (some of which are highlighted in section 6), as they are based on synergetic models, it is possible to use the principle of new mereology - the philosophical study of wholes and parts, which states that in dissipative structures (i.e., self-organizing systems) parts are modified by their composition into a whole. The existing versions of mereology rely on the assumption that parts are not changed by being associated into wholes. To put it simply, the sum of the single components' properties is not equal (in the qualitative sense) to the compound properties of these components (as a whole). In synergetic models combining parts of the information (composing them into a whole) may lead to the emergence of new properties of the resultant compound system.

There is a need for further and better research on the phenomenon of self-organization in natural systems and in synergetic models. Future developments in the science of self-organization are likely to focus on more complex computer simulations and mathematical methods. However, the basic mechanisms underlying self-organization in nature are still far from clear, and the different approaches need to be better integrated [68].

Investigating the principles of self-organization, however, is not a simple task and needs careful and reason approaches. While approaching this conception, it must be remembered that human does not necessarily have to be the life's utmost creation on this planet nor does human understanding of the surrounding world have to be unconditionally adequate.

5 ANALYSIS OF THE SYNERGETIC COMPUTER THEORY

5.1 Standard synergetic computer model

An overview of the synergetic computer standard or classical model is presented in this section and is based mainly on the information presented in [45], (for in-depth discussion and theorem proofs, refer to [45]). As it has already been noted, the notion of the synergetic computer and its theoretical consideration was introduced in 1989 by Hermann Haken, who discovered the profound analogy between pattern formation and pattern recognition. The concept of synergetic computer was developed by the use of the synergetics theory, which was already available at this time. The synergetics theory was also developed by Hermann Haken.

For a relatively long period of time the field of computers has been dominated by serial computers based on the concept of the universal Turing machine and on the von Neumann architecture. Serial computers may process numbers as well as symbols, and they are thought to be universally applicable, at least in principle. In practice, however, there are some limitations, which become evident when certain specific tasks are to be addressed by computers. For example, in vision modeling an enormous number of bits have to be processed, and if real time processing is required, even faster computers are too slow. Quite evident that biology has masterly solved such kind of problems. In spite of the fact that neurons are slow, having processing times of the order of milliseconds, human is able to recognize patterns within a fraction of a second. Thus, the brain works by means of a different principle, which can be only parallel processing. So the question arises of how to construct the parallel computing model. A possible answer may be the Hillis machine [69] (or hypercube), but this has the drawback that it requires heavy programming. As nature has shown, there must be ways in which learning, or in other words self-programming, occurs rather easily. This has led scientists to devise neural computers constructed in a manner analogous to the nets of neurons of animal or human brain.

The majority of present day concepts rely heavily on the model proposed by McCulloch and Pitts (in 1943) [70], who represented neurons as two-state elements which receive inputs and have outputs. A neuron is activated only if the sum of the input signals exceeds a specific threshold value. Early attempts to realize these basic ideas, in particular by Rosenblatt [71], who constructed the perceptron, were not followed up for a while because Minski [72] had shown that the perceptron could not learn certain logical tasks, such as the “exclusive or”. Over the last decades, however, there has been an enormous revival of this field initiated by several groups of researchers. Their concepts are still based on the fundamental idea of McCulloch and Pitts but with some slight modifications;

in particular, with respect to the shape of the threshold curve. At present, no unique opinion exists among the experts as to how far this approach may go. On the one hand, there are some beautiful results, for instance, those of Sejnowski [73], who trained the neural net to learn a spoken language so that the net could perform at the same level as children in the first or second year of school. On the other hand, there is no general theory of what a network can really do, or of how it can be trained in a reliable and fast manner. In fact, learning is still a major problem and, at present, predictions for the future of these devices are difficult to make. Thus the novel concept of a synergetic computer must be viewed against this background.

The concept of the synergetic computer stems from the interdisciplinary field of synergetics, which will be discussed in more detail below. The synergetic computer utilizes far-reaching analogies between spontaneous pattern formation and pattern recognition. In the following subsections, basic equations of the synergetic computer are introduced. These equations can be solved on a serial computer, but can also provide the construction principle of a new type of parallel network, in which the individual nodes or neurons have quite different properties to those of the previous neural computers in the sense of McCulloch and Pitts. The most prominent feature of this approach is the following: the behavior of the network can be treated rigorously in the mathematical sense, so that its performance can be precisely predicted. In particular, there are no so-called spurious states, which are unwanted and in which the system can become trapped. This difficulty, which occurs in both the pattern recognition and learning, and which has been a major problem in traditional neural computers, does not appear in the synergetic computer. In contrast to the bottom-up approach of neural computers where one starts with the properties of individual neurons, and then tries to fix the links between them in such a way that the network performs specific tasks, the approach to the construction of a synergetic computer is top-down. First, the desired properties are identified, and then an algorithm is established which eventually leads to a technical realization.

5.1.1 Cognitive processes and synergetic computers

The simulation or, still better, the understanding of human behavior by means of machines has a long tradition. An example could be clockwork dolls built to mimic human motion such as dancing, etc. Today, the simulation and understanding of human activities, especially of cognitive processes, is undergoing a revolution that began with the advent of the electronic computer. Because it was soon recognized that computers cannot only process numbers but also symbols, simulation of the tasks performed by humans, for instance playing chess, or preparing technical drawings were tackled by means of electronic computers. Cognitive processes were modeled by strings of symbols which were proposed consecutively. Among the early pioneers, Herbert Simon and Allen Newel [74] as well as Marvin Minski [75] may be mentioned. After an initial period of great enthusiasm, a rather critical debate about the success of their concept of Artificial Intelligence can presently be witnessed. Hermann Haken [45] is sure that a similar debate will occur about connection machines, i.e., "neurocomputers", (and synergetic computers) in the future, especially if we are making farfetched claims about their role in *fully* understanding or simulating human intelligence. It was at the time AI was defined that connection machines, in particular the perceptron, came into existence, were then abandoned, and are presently experiencing an enormous revival.

It might be reasonably asked whether connection machines, or in other words neural computers, can, in a single step, conceptually bridge the enormous gap between microscopic events taking place in the real neurons of the brain and the macroscopic phenomena of cognition, or whether the intermediate steps are required. The latter point of view has been clearly expressed by Smolensky [76]. We are also inclined to support this latter view. It is here that the synergetic computer comes in. Being based on a top-down approach, macroscopic events can be studied and attempts made to simulate them by a network which in turn may combine the functions of subunits composed of a greater or lesser number of neurons. In [45], it is demonstrated that such an approach allows making specific predictions that can be compared to psychological findings. The synergetic computer may be further optimized to recognize pattern, and the process is invariant with respect to displacements, rotation, and scaling. It turns out that these invariance properties can be devised in different manners. In the synergetic computer theory a parameter which may be directly related to psychological attention is responsible for the recognition of scenes by a computer. This may shed new light on the way humans perceive complex scenes. The same attention parameter will turn out to be responsible for oscillations that occur in human perception of ambiguous patterns/figures such as in Fig. 5.2. The application of the synergetic computer to the recognition of movement patterns, for instance, the distinctions between different gates of horses, leads to the question whether or not humans perceive these patterns in a similar manner.

Vision has been chosen as an example of cognitive processes for several reasons. First, pattern recognition can be easily performed by the synergetic computer and the results can be easily compared with those of psycho-physical experiments, even in the quantitative manner. At the same time, we believe that vision is a useful paradigm for higher mental processes, such as reasoning, problem solving, creative thinking etc. Interestingly, it is also mirrored in language by the existence of expressions such as "to gain insight" or "to develop a picture of the situation".

In many, if not all cases, as H. Haken believes, and we also agree with him, that the processes of cognition may be thought of as pattern formation by self-organization. This must occur at the abstract level as far as concepts are concerned, and at the material level in cognitive processes related, for example, to firing patterns in neural nets.

We also believe that many, if not all the processes in the surrounding world are driven by the intrinsic laws of self-organization, and thus may be modeled, at least to some degree, by synergetics.

5.1.2 Overview of the mathematical theory of synergetics

As the synergetic computer is based on synergetics theory, a brief reminder of its main concepts is given here.

The system is described by a state vector having components

$$q=(q_1,q_2,...,q_M). \tag{5.1}$$

When treating pattern formation in continuous media, the individual components q_j are not only dependent on time t, but also on space

$$q_j=q_j(x,t) \tag{5.2}$$

where $x=(x,y,z)$.

Examples for the possible meanings of the components are provided by the case of fluid, where the individual components q_j may mean density $\rho(x,t)$, velocity field $v(x,t)$, and temperature field $T(x,t)$.

In chemistry, the symbols may represent concentrations n of different kinds of chemical labeled by indices $j=1,...,M$,

$$q=(n_1,n_2,...,n_j,...,n_M) \tag{5.3}$$

which again depend on space and time.

In biology the components could be the density of specific cells, the firing rates of neurons, etc. In the following we shall use an abbreviation for the time derivative of q namely

$$\frac{dq}{dt}=\dot{q}. \tag{5.4}$$

In all cases considered by the mathematical theory of synergetics, the state vector (5.1) undergoes a time evolution according to equation of the form

$$\dot{q}(x,t)=N[q(x,t),\nabla,\alpha,x]+F(t). \tag{5.5}$$

Here N is a vector which depends on the state variable q at the same or at different space points as those occurring on the left side of (5.5). In continuously extended media diffusion or wave propagation will take place so that the nabla operator $\nabla = (\partial / \partial x, \partial / \partial y, \partial / \partial z)$ appears. The system is subject to external controls, e.g., to the temperature difference in the Bénard instability, to the rate at which chemicals are poured into the system undergoing chemical reactions, etc. The control parameters are denoted by α. In general, N may also depend on special inhomogeneities as indicated by x. The function $F(t)$ represents fluctuating forces which stem from internal or external fluctuations. Their properties are specified later. In a number of cases the fluctuations may be neglected, in others they pay a decisive role. What happens in each case, is indicated below.

Let us consider some examples of equations of the type (5.5). In chemistry as well as in biological population dynamics we deal with numbers or concentrations of molecules (individuals of a species). A typical equation for the rate of change of the concentration n_j is given by the form

$$\dot{n}_j = \alpha n_j - k n_j + k' n_l n_m . \tag{5.6}$$

The first term on the right-hand side corresponds to a so-called autocatalytic multiplication, because the growth rate of n_j is proportional to its own concentration n_j. The next term stems from a spontaneous decay of this type of molecule (or the death of individuals of a species). The last term describes the production of a molecule of type j by the interaction of molecules of type l and m. Equation (5.6) may be written in short in the form

$$\dot{n}_j = R_j(n) \tag{5.7}$$

where the vector n is given by

$$n = (n_1, n_2, ..., n_M). \tag{5.8}$$

In general, not only reactions but also diffusion processes take place. In such a case n_j becomes a function of space and time

$$n_j = n_j(x,t). \tag{5.9}$$

The temporal change of concentration n_j is determined by

$$\dot{n}_j = R_j(n) + D_j \nabla^2 n_j \tag{5.10}$$

in which D_j is the diffusion constant and ∇^2 the Laplace operator. Equations of the type (5.10) are called reaction-diffusion equations.

Quite naturally, it is not possible to solve the equation (5.5) for the general case. However, the basic idea of synergetics is as follows: when a system is driven only weakly by external controls, there will be a time-independent state q_0 which, in the case of homogeneous system, is even space-independent

$$\alpha_0 \Rightarrow q_0. \tag{5.11}$$

From many experiments it is known that the state (5.11) can change qualitatively when the control parameter is changed from α_0 to α. In order to check the stability of the solution (5.11), we make the hypothesis

$$\alpha \Rightarrow q(x,t) = q_0 + w(x,t). \tag{5.12}$$

Inserting (5.12) on the right-hand side of (4.5), neglecting the fluctuating forces for the time being, and expanding the nonlinear function N as a power series in w, the following equation is obtained:

$$N(q_0 + w) = N(q_0) + Lw + \hat{N}(w). \tag{5.13}$$

L on the right-hand side of (5.13) is a matrix

$$L = (L_{ij}) \tag{5.14}$$

which may contain spatial derivatives. The matrix elements are defined by

$$L_{ij} = \frac{\partial N_i}{\partial q_i} \tag{5.15}$$

at $q = q_0$. $\hat{N}(w)$ is a nonlinear function containing the second and/or higher powers of w. Since we are interested, at least for the moment, only in the onset of instability, it may be assumed that w is small so that the nonlinear term in (5.13) can be neglected. Because q_0 was assumed to be a stationary solution, which changes uniquely with the control parameter α,

$$\dot{q}_0 = N(q_0) = 0. \tag{5.16}$$

Consequently, in the linear stability analysis we are left with

$$\dot{w} = Lw. \tag{5.17}$$

The solution of (5.17) can be written in the general form

$$w = e^{\lambda t} v(x) \tag{5.18}$$

provided the eigenvalues λ of L are nondegenerate. Otherwise v may contain powers of t. For the sake of simplicity, we shall focus our attention on the nondegenerate case. The eigenvalues and eigenvectors are distinguished by the index j

$$\lambda_j, v_j(x). \tag{5.19}$$

Now the fully nonlinear equations (5.5) are solved, taking care also of the fluctuating forces. To this end, the hypothesis is made

$$q = q_0 + \sum_j \xi_j(t) v_j(x). \tag{5.20}$$

Inserting it into (5.5), where the decomposition (5.13) is used, the following equation is obtained:

$$\sum_j \dot{\xi}_j(t) v_j(x) = \sum_j \xi_j(t) L v_j(x) + \hat{N}[\sum_j \xi_j(t) v_j(x)] + F(t). \tag{5.21}$$

As can be shown, it is always possible to construct a set of adjoint functions $v_k^+(x)$ with the property

$$\langle v_k^+ v_j \rangle \equiv \int v_k^+(x) v_j(x) dV = \delta_{kj}. \tag{5.22}$$

where δ_{kj} is the Kronecker symbol, $\delta_{kj} = 1$ for $k = j$ and 0 otherwise. (5.21) is multiplied by $v_k^+(x)$ and integrated over the space. Using the property

$$Lv_j(x) = \lambda_j v_j(x) \tag{5.23}$$

and the definitions

$$\int v_k^+(x) F(t,x) dV = F_k(t) \tag{5.24}$$

and

$$\int v_k^+(x) \hat{N}[\sum_j \xi_j(t) v_j(x)] dV = \tilde{N}_k[\xi_j(t)], \tag{5.25}$$

the equations (5.21) can be cast into the form

$$\dot{\xi}_k = \lambda_k \xi_k + \tilde{N}_k(\xi_j) + F_k(t). \tag{5.26}$$

Now, two cases depending on the sign of the real part of the eigenvalues λ_j can be distinguished. If the real part is non-negative, the corresponding configurations $v(x)$ are called *unstable modes*, and in the opposite case, they are called *stable modes*; to distinguish between them, the abbreviations u (unstanble) and s (stable) are introduced. According to this discrimination, the equations (5.26) can be split into two sets, namely

$$\dot{\xi}_u = \lambda_u \xi_u + \tilde{N}_u(\xi_{u'}, \xi_{s'}) + F_u \tag{5.27}$$

and

$$\dot{\xi}_s = \lambda_s \xi_s + \tilde{N}_s(\xi_{u'}, \xi_{s'}) + F_s. \tag{5.28}$$

Note that the indices u and s serve a double purpose: the u and s of ξ_u and ξ_s, respectively, label the variables where $u = 1, ..., n,$ and $s = n+1, ..., K,$ (where K is the number of modes); whereas in ξ_u and ξ_s the index distinguishes between the set of unstable and stable modes, respectively. As the system is operated in a situation where the real part of λ_u is still small, the *slaving principle* of synergetics can be applied. It shall not be derived here because of its very lengthy proof, but an example is given of how it works. For simplicity, the case where λ_u is real is considered, focusing on a special case of (5.28), namely

$$\dot{\xi}_s = \lambda_s \xi_s + \xi_u^2 + ... \tag{5.29}$$

Because λ_u is small (and ξ_u is assumed to be small, too), it can be concluded from (5.27) that ξ_u changes only very slowly in time. According to equation (5.29), ξ_u drives ξ_s, and thus results in $\dot{\xi}_s$ being of the order of

$$\dot{\xi}_s \approx \lambda_u \xi_s. \tag{5.30}$$

As λ_u is much smaller than λ_s, $\dot{\xi}_s$ can be neglected on the left-hand side of (5.29) entirely:

$$\dot{\xi}_s = 0. \tag{5.31}$$

This allows for replacing the differential equation (5.29) by a normal algebraic equation which can be solved immediately

$$\xi_s = -\frac{\xi_u^2}{\lambda_s} - \frac{F_s}{\lambda_s}. \tag{5.32}$$

Actually, a detailed discussion shows that this approximation is even permissible for the fluctuating force F_s since it is taken care of in (5.32). This example shows that ξ_s can be expressed by ξ_u explicitly, and with the same time argument as ξ_u. In other words, ξ_s follows ξ_u instantaneously. Under the assumption that the real parts of λ_u are small, it can be shown quite generally that ξ_s can be expressed explicitly by ξ_u and with the same time argument:

$$\xi_s(t) = f_s[\xi_u(t);t]. \tag{5.33}$$

The additional explicit time-dependence appearing in (5.33) stems from the fluctuating forces; if they are absent, this time-dependence vanishes. By means of (5.33) ξ_s can be eliminated in (5.27) which gives an equation of the form

$$\dot{\xi}_u = \lambda_u \xi_u + \tilde{N}_u(\xi_{u'}) + F_{u,tot.} \tag{5.34}$$

The general case, in which q_0 may be periodic or quasiperiodic in time, and where the λ 's are complex is treated in the literature.

Because of (5.20) and (5.33), the evolving structure or, in other words, the ordering of the system is determined by ξ_u, and thus these variables are called *order parameters*. Equation (5.33) is known as the *slaving principle*.

In order to derive a typical explicit form of (5.34), consider a simple case in which ξ_u obeys the equation

$$\dot{\xi}_u = \lambda_u \xi_u - \xi_s \xi_u \; . \tag{5.35}$$

Using (5.32), we find

$$\dot{\xi}_u = \lambda_u \xi_u - \beta \xi_u^3 + F \tag{5.36}$$

where F is fluctuating force.

In order to grasp the meaning of (5.36), the equation is written in the form

$$\dot{\xi}_u = -\frac{\partial V}{\partial \xi_u} + F, \tag{5.37}$$

i.e., a potential function V is introduced. By adding the acceleration term $m\ddot{\xi}_u$, where m has the meaning of a mass of a particle, to the left-hand side of (5.37), the latter can be interpreted as the equation of the overdamped motion of a particle in a potential V subject to a fluctuating force F . For the case of a single order parameter ξ_k the potential is plotted in Fig. 5.1 for positive and negative values of λ_u . Quite evidently, the potential changes qualitatively.

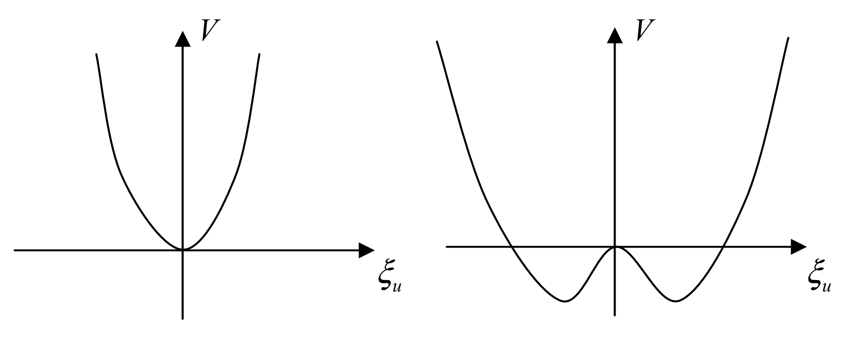


Fig. 5.1 The potential V as function of the order parameter ξ_k for a negative control parameter (*left*) and a positive control parameter (*right*)

It can be immediately seen that for $\lambda_u < 0$ the particle remains close to the minimum and is only displaced slightly by the fluctuating forces F. For the case $\lambda_u > 0$, two minima occur, i.e., ξ_u adopts non-vanishing stable displacements which are subject to small fluctuations. The behavior of the variable ξ_u exhibits features which are well known from systems in thermal equilibrium when they undergo *phase transitions,* e.g., when a magnet goes from its unmagnetized state into a magnetized state or a superconductor from its normal state into the superconducting state. Here the following phenomena are expected: the formerly stable position $\xi_u = 0$ becomes unstable and is replaced by two new stable positions according to Fig. 5.1. The two states are totally equivalent, or, in other words, symmetric with respect to one another. Nonetheless, the system has to decide between these two states and can adopt only one state. Therefore it has to break the symmetry. Such a transition is thus called a *symmetry breaking instability*. Furthermore, when λ_u grows starting from negative values, the potential curve of Fig. 5.1 (left) becomes flatter and flatter. Thus, the particle, which is kicked away from the equilibrium point by the fluctuating force, relaxes back more and more slowly. This, too, is well known from equilibrium phase transitions as *critical slowing down*. Furthermore, because the restoring forces are becoming weaker and weaker, the fluctuations of the "particle", i.e., of ξ_u, become more and more pronounced. Thus *critical fluctuations* occur. Since these phenomena, which are all typical for the conventional phase transitions of systems in thermal equilibrium, now also occur in systems far from equilibrium as treated by synergetics, the change in the behavior of ξ_u is called a *nonequilibrium phase transition*.

As it was shown by the explicit treatment of numerous examples, the number of order parameters is generally much smaller than the number of mode amplitudes ξ_s, which are also called the *enslaved modes*. In a number of cases the right-hand side of (5.34) can be written as a derivative of a potential function V, even if several order parameters are present [cf. (5.37)], where $V = V(\xi_u)$. The potential can be visualized as a landscape having hills and valleys. The bottom of each valley represents a *stable fixed point*, the top of each mountain an *unstable fixed point*. Because the stable fixed point seems to attract the "particle", they are also called *attractors*. Note, however, that the fixed point is a special case of attractor. All points in the landscape from which the particle can roll down into the same attractor form the *basin of attraction*. Points of minimal height on ridges may be described as *saddle points*.

Equations of the type (5.37) turn out to play a decisive role in the approach to pattern recognition.

Once (5.34) or (5.37) have been solved and ξ_s calculated according to (5.33), the desired solution $q(x,t)$ can be constructed by means of (5.20), where the order parameters ξ_u and the amplitudes ξ_s of the enslaved modes can be distinguished now. The following equation is obtained:

$$q = q_0 + \sum_u \xi_u(t) v_u(x) + \sum_s \xi_s(t) v_s(x). \tag{5.38}$$

Close to the transition point the order parameters ξ_u are much bigger than the amplitudes ξ_s, and thus the pattern is described by the first sum in (5.38); this sum is therefore called "the mode skeleton".

In order to understand the pattern recognition theory, consider (5.38) more closely. For the sake of convenience, it is assumed that

$$q_0 = 0. \tag{5.39}$$

When (5.38) is multiplied by v_j^+ and the identity (5.22) used, the following is found

$$\xi_j = \langle v_j^+ q \rangle. \tag{5.40}$$

Since in the following the behavior of the mode-skeleton is of interest, the original equations are changed so that the last sum in (5.38) tends to 0 for $t \longrightarrow \infty$

$$\sum_s \xi_s(t) v_s(x) \to 0. \tag{5.41}$$

For ξ_u equation (5.34) is adopted, i.e.,

$$\dot{\xi}_u = \lambda_u \xi_u + \tilde{N}_u(\xi_{u'}) + F_{u,tot}, \tag{5.42}$$

whereas the equations for ξ_s are replaced by

$$\dot{\xi}_s = -g_s(\xi_{u'}, \xi_s) \tag{5.43}$$

where g_s is a function with the property

$$\xi_s \to 0 \text{ for } t \to \infty. \tag{5.44}$$

Now, (5.42) and (5.43) are multiplied by $v_j(x)$ and summed up over j. Thus the following is obtained:

$$\dot{q} = \sum_u \lambda_u \langle v_u^+ q \rangle v_u + \sum_u v_u \tilde{N}_u(\langle v_u^+, q \rangle) + \sum_s v_s g_s(\langle v_j^+ q \rangle) + \tilde{F}(t). \tag{5.45}$$

Equation (5.45) will be used further as the starting point for the formalism on pattern recognition, see [45] for more details and discussion.

5.1.3 Prototype pattern vectors and test pattern vectors

In order to construct the model, three ingredients are used:

A. The concept of associative memory (or modeling by analogy principle, outlined in section 6.3). When an incomplete set of data is given, the associative memory must be able to complement it.

B. A dynamical process is constructed by which pattern recognition is performed. To this end, a potential landscape is invented in which a fictitious particle, which describes the patterns, moves. An example is provided by the ambivalent patterns on Fig. 5.2. In this case, the pattern that is recognized first is the one for which a certain bias was given, or in other words, the pattern is recognized once it is within its basin of attraction.

C. The system is treated as a synergetic system according to the following idea: partially ordered system, e.g., a fluid in which some of the rolls have been formed, may generate its order parameter, which then competes with the other order parameters of the system. Because of the special preparation of the initial state involving partially ordered subsystems, the order parameter belonging to that specific order wins the competition and, eventually, enslaves the whole system so that it enters a particular ordered state. In pattern recognition the same mechanism is used. Once a set of features is given, they can form their own order parameter which will compete with other order parameters. Eventually the order parameter that had the strongest initial support will win, and will force the system to exhibit the features that were lacking with respect to the special pattern (Fig. 5.2). Thus, there is a complete correspondence between the complementation process during the pattern formation and associative memory during pattern recognition.

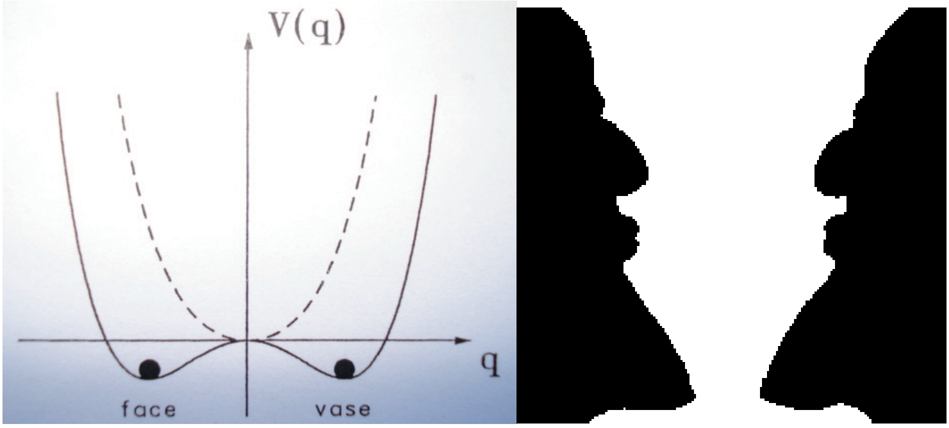


Fig. 5.2 Interpretation of pattern recognition by means of an order parameter moving in a potential landscape with two attractors, vase and face, respectively (left). The right-hand side shows the ambiguous pattern vase/face (picture taken from H. Haken, [45]).

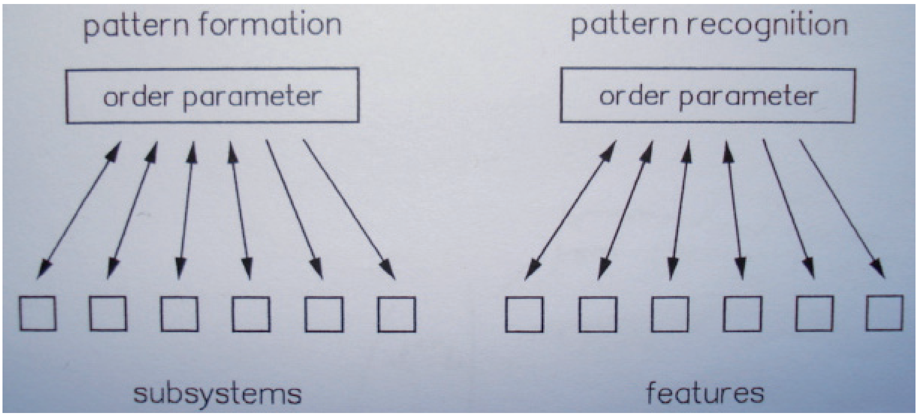


Fig. 5.3. Analogy between pattern formation (left) and pattern recognition (right). (picture taken from H. Haken, [45]).

In pattern formation the subsystems are enslaved by the order parameters; in the case of pattern recognition it is the features that are enslaved by order parameters.

The mathematical form of the process of construction of the test and prototype pattern vectors is outlined below, in section 7.1.1. Next the construction of the dynamics of the system being modeled is dealt with.

5.1.4 Construction of the Dynamics

A test pattern, which is denoted by the vector q, is given. Then a dynamics is constructed which pulls the test pattern via intermediate states $q(t)$ into one of the prototype patterns v_{k_0}, namely the one to which $q(0)$ was closest, i.e., in whose basin of attraction it was lying:

$$q(0) \rightarrow q(t) \rightarrow v_{k_0}. \tag{5.46}$$

The main results of the previous sections immediately provide such a dynamics: the equation of motion for q is constructed in such a way that it resembles (5.37) or, more specifically (5.45), and has the property that it eventually leads to an order parameter equation which allows, e.g., a discrimination of the roll patterns in fluids. Here, of course, the patterns may be far more complicated and the required equation reads

$$\dot{q} = \sum_k \lambda_k v_k (v_k^+ q) - B \sum_{k' \neq k} (v_{k'}^+ q)^2 (v_k^+ q) v_k - C(q^+ q) q + F(t). \tag{5.47}$$

The constant B can be made dependent on k and k' ,

$$B \to B_{kk'}, \tag{5.48}$$

and then has to be taken into the sum. But for the time being this generalization is not needed.

Here is a brief discussion of the meaning of the individual terms on the right-hand side of (5.47). λ_k are called attention parameters. A pattern can be recognized only if the corresponding attention parameters are positive; otherwise it will not be recognized. The expression $v_k \cdot v_k^+$ acts as a matrix. As can be seen, first, v_k^+ is multiplied by a column vector q so that a scalar is generated. Finally, the vector v_k becomes effective which again acts as a column vector. Thus, by the whole process, a column vector is transformed into a new column vector or, in other words, $v_k \cdot v_k^+$ acts as a matrix. The second term serves to discriminate between patterns. Because the first term, at least when λ is positive, will lead to an exponential growth of q , a factor which limits that growth is needed. This is achieved by the third term on the right-hand side. Finally F are the fluctuating forces.

The vector q can be decomposed into the prototype vectors and a residual vector w

$$q = \sum_{k=1}^{M} \xi_k v_k + w \tag{5.49}$$

requiring that

$$(v_k^+ w) = 0 \ \text{ for all } \ k = 1, \ldots, M. \tag{5.50}$$

Now q^+ which appears in $C(q^+ q)q$ in (5.47) is defined, by means of the relations

$$q^+ = \sum_{k=1}^{M} \xi_k v_k^+ + w^+ \tag{5.51}$$

where w^+ obeys the orthogonality relations

$$(w^+ v_k) = 0 \ \text{ for all } \ k = 1, \ldots, M. \tag{5.52}$$

It is obvious that

$$(v_k^+ q) = (q^+ v_k). \tag{5.53}$$

This is followed by inserting (5.49) into the left-hand side of (5.53) to yield

$$(v_k^+ q) = [v_k^+ (\sum_{k'=1}^{M} \xi_{k'} v_{k'} + w)], \tag{5.54}$$

which, by means of the orthogonality relations (7.8, 5.50), can be written as

$$(v_k^+ q) = \xi_k. \tag{5.55}$$

The same result is obtained when inserting (5.51) on the right-hand side of (5.53) and utilizing the orthogonality relations (7.8) and (5.52). Equation (5.53) allows to express the right-hand side of (5.47) either by means of the left-hand side of (5.53) everywhere, or by the right-hand side of (5.53) everywhere, so that only the variables q or q^+ are involved. It can then be readily shown that the formal equations

$$\dot{q} = -\frac{\partial V}{\partial q^+}, \tag{5.56}$$

$$\dot{q}^+ = -\frac{\partial V}{\partial q}, \tag{5.57}$$

hold, in which V plays the role of a potential function explicitly given by

$$V = -\frac{1}{2}\sum_{k=1}^{M} \lambda_k (v_k^+ q)^2 + \frac{1}{4} B \sum_{k' \neq k} (v_{k'}^+ q)^2 (v_k^+ q) + \frac{1}{4} C (q^+ q)^2. \tag{5.58}$$

Thus a rather far-reaching analogy is revealed with (5.37) and (5.45).

Now the order parameter equations belonging to (5.47) are derived. To this end, (5.47) is multiplied from the left by v_k^+. Using the orthogonality relations between v_k^+, v_k, w, w^+, and the definition (5.55), the following equation is obtained:

$$\dot{\xi}_k = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C \left[\sum_{k'=1}^{M} \xi_{k'}^2 + (w^+ w) \right] \xi_k. \tag{5.59}$$

Note that the sum over $k' \neq k$ runs over all values $k' = 1...M$ except for the value k which appears as the coefficient of $\dot{\xi}_k$ on the left-hand side of (5.59). When (5.47) is multiplied by vectors u_l^+ belonging to the space orthogonal to the prototype patterns vectors v_k, and sum up over the individual components, the following equation is obtained:

$$\dot{w} = -C \left[\sum_{k'=1}^{M} \xi_{k'}^2 + (w^+ w) \right] w. \tag{5.60}$$

In the derivation of (5.60), the following relations have been used:

$$w = \sum_{l=M+1}^{N} f_l(t) u_l, \tag{5.61}$$

$$w^+ = \sum_{l=M+1}^{N} f_l(t) u_l^+, \tag{5.62}$$

$$(u_l^+ u_l) = \delta_{ll'}, \tag{5.63}$$

$$(u_l^+ v_k) = (v_k^+ u_l) = 0. \tag{5.64}$$

Because the factor multiplying w on the right-hand side of (5.60) is negative everywhere

$$|w| \to 0 \text{ for } t \to \infty, \tag{5.65}$$

where the norm $|w|$ is defined by

$$|w| = (w^+ w)^{\frac{1}{2}}. \tag{5.66}$$

Thus the dynamics reduces the problem to the prototype vector space

$$\dot{\xi}_k = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C \left(\sum_{k'=1}^{M} \xi_{k'}^2 \right) \xi_k . \tag{5.67}$$

The order parameters obey the initial condition

$$\xi_k(0) = (v_k^+ q(0)) \tag{5.68}$$

which follows directly from (5.55). Quite clearly, the right-hand side of (5.53) can be derived from a potential function

$$\dot{\xi}_k = -\frac{\partial \tilde{V}}{\partial \xi_k}, \tag{5.69}$$

where $\tilde{V}$ is given by

$$\tilde{V} = -\frac{1}{2} \sum_{k=1}^{M} \lambda_k \xi_k^{\,2} + \frac{1}{4} B \sum_{k' \neq k} \xi_{k'}^2 \xi_k^2 + \frac{1}{4} C \left(\sum_{k'=1}^{M} \xi_{k'}^2 \right)^2 . \tag{5.70}$$

Note that the sum over $k' \neq k$ is now a double sum over k' and k each run from $1 ... M$, but with the term $k' = k$ omitted. The role played by the individual terms in (5.58) or in (5.70) can be easily visualized by considering a plot of the potential V in a two-dimensional feature space, where the two axes are spanned by the two prototype vectors v_1 and v_2. Close to the origin, the terms quadratic in q dominate and the first sum in (5.58) has a negative sign provided the attention parameters λ_k are positive. This decrease in the potential ceases when q increases further because then the last term in (5.58) takes over, and eventually, increases much more quickly than the first term decreases. This interplay between the first and last terms generates the valleys. The middle term on the right-hand side of (5.58) finally generates the ridges that define the basins of attraction, and thus enables discrimination between different patterns.

It is also assumed that all attention parameters are equal and positive

$$\lambda_k = \lambda > 0 \tag{5.71}$$

and $\lambda = C$.

The stable fixed points are at $q = v_k$, i.e., at the prototype patterns, and there are no other stable fixed points. The stable fixed points are equally characterized by $\xi_k = 1$, all other ξ's = 0. The only unstable fixed point is at $q = 0$. There are saddle points situated at $\xi_{k_1} = \xi_{k_2} = ... = \xi_{k_m} = 1$, all other $\xi's = 0$. Here $k_1, k_2, ...k_m$ may be any selection out of $1,...,M$. If $|\xi_{k_0}|$ is initially bigger than any other $|\xi|$, the dynamics terminates at the corresponding saddle point, from which only a fluctuation may drive the system into any of the fixed points belonging to $k_1, k_2, ..., k_m$.

5.2 Possible directions for improvements

Some additional modifications to the standard synergetic computer model discussed in the previous section are possible. In this research, useful modifications to the standard model are introduced, tested and implemented in the form of a HVAC software application on the AutoCAD platform, see chapters 7 and 8 for more in-depth discussion.

In this subsection, possible directions for further improvement of the classical synergetic computer model (in addition to those, described in our implementation) are briefly outlined. Implementation details or mathematical expression are not presented here, as this would be a subject for a separate research.

As the traditional way of generating the order parameter, which is dot-product (or pseudo-inverse) based has certain limitations, the first possible class for improvements is changing the order parameters generation mechanism. Essentially, the initial order parameters generation is application of the similarity measure to the comparison between given vectors. There are actually a number of these measures available, for example the mean square error method, correlation coefficient, the absolute difference method, distance method, etc. Such modifications allow for improving the time and accuracy of performance of the synergetic computer or synergetic neural network (SNN).

Another way of optimizing the performance of the SNN is omitting the order parameters' iterations, i.e., introducing the Quick Haken Algorithm (see section 7.2). The analysis of the potential function shows whether attention parameters, the potential function and all its attractors and attracting domains are determined. As a structural feature of synergetic neural network, the domain of attraction of the attractor is completely determined by the initial order parameter and attention parameters, and has the shape according to the rules/equations which are determined directly (especially when the initial value of balanced attention parameters is much larger than the order parameter). The initial value of the order parameter of the evolution of the trajectory has to be determined directly. This saves the time required for the network iteration. Namely, this is the basic idea of Quick Haken Algorithm. The potential function analysis shows/demonstrates that the nature of Quick Haken network parameters and productivity is based on the initial selection of the order parameters, and depends on the input mode in which a prototype pattern corresponds to domain of attraction.

Note that in the case of balanced attention parameters, the order parameter with the largest initial value will be the winner of the competition. In case of unbalanced attention parameters, it is the difference between parameters and the relative size of the number of order parameters of the comparison that determines the outcome of the competition. This also means that it is the relative size of the order parameter rather than absolute size that determines the competition results.

Thus, another class for possible improvements is investigation of the impact of different ways of composition of the attention parameters, especially in the case of unbalanced attention parameters.

In case of SNN, the analysis of the properties of the competition layers of the network is the source for possible optimization and improvements.

In the matching subnet class it can be shown that classical Haken model for the mass pattern recognition is extremely difficult, because it takes a lot of time to deal with extremely difficult vector structures, which requires searching for new methods to solve the massive synergetic neural network data processing. For example, it may be a hierarchical approach to solve the competition problem. In fact, this may lead to a new level of model-based neural network - the neural networks for collaboration.

Biological neural networks have structures with a clear hierarchy of features. First, the brain is divided into two hemispheres, one for creative thinking and the other for logical thinking. Second, the cerebral cortex is divided into many functional areas, such as the visual area, olfactory areas, etc.; in each district, the cells are organized in an orderly fashion together. Cells that react to a particular stimulus or task are organized in a small cylindrical region to work together. From the viewpoint of the traditional artificial neural network, this organization has the following advantages:

1. Robustness - failure of some areas will not lead to the full system failure;
2. The learning tasks of the global learning and local learning divisions are clearer and faster.

Traditional neural network structure also partially reflects the level of typical class structure principles, such as MP model - Perceptron - MLP-BP network. However, being generally robust in case of small samples, regarding large samples they have a lot of constraints. In high-dimensional networks the number of nodes is too large and will result in making learning and work extremely difficult. Therefore, based on the level of learning tasks of biological information processing behavior of the proposed neural network, it will be possible to use primitive neural networks to organize them as a more complex neural network, and to divide the global network for local learning tasks. It is theoretically possible to test this idea in the synergetic neural network, whereas the introduction of multi-order parameter evolution of the local competition allows changing the characteristics of current single global competitive learning that is slow, and has the problem of poor robustness.

In the order parameter evolution for large samples a large number of high-dimensional learning problems exist. The introduction of local competition, to change the traditional co-existence of global competitive neural network is only caused by poor robustness and problems of slow learning. There are two possible ways to address such problems:

- Subdomain competition. While matching the synergetic neural network model of the prototype in the sub-sub clan, first the process of identifying the order parameter in "inter-ethnic" competition has to be conducted, the order parameter that wins the competition represents a model closer to those being identified with the subdomain, then the subdomain competition for the local order parameter follows.
- Introduction of the equation of multi-order parameter competition. Most patterns use a low degree of similarity between the characteristics of the order parameter evolution equations that could be put into a matrix with the properties of sparse matrix equations, and then solved (simultaneous equations' processing).

The above two methods are equivalent to the synergetic neural network that is being decomposed into several subnetwork, where each network is computing tasks to complete a partial, and then higher dimensions in the global competition. Additional experimental work must be carried out for that in future studies.

5.3 Review of the related research

Although the synergetics theory and synergetic computer algorithms have not been realized on the AutoCAD platform before (nor is there any information about previous researches on the implementation of these theories on other CAD platforms), some researches have been conducted on the application of these algorithms for various pattern recognition purposes. In this section, the most relevant of the previous researches are reviewed. It must be noted that there are very few available.

5.3.1 Synergetic stereo matching algorithm for occlusion and reversal position

In [77], the modified synergetic computer algorithm proposed for image comparison and matching used in robot vision applications is described. Matching between left and right images is represented by differential equations with order parameter and attention parameter. In this research, synergetic matching is opposed to conventional matching algorithms such as template matching and dynamic programming matching (DP matching). It is also shown that the proposed synergetic algorithm has higher matching precision compared to conventional methods.

In computer vision, 3-D reconstruction from stereo images is applied to many fields of robot vision and virtual reality. The main issue in processing stereo images is deciding the correspondence between multiple images, i.e., the stereo matching problem [78]. Stereo matching of images including occlusion (correspondence of partially hidden objects) and reversal position (the order of corresponding point differs from the other order) remains difficult, however. Multi-eye stereo using nine cameras [79] is proposed to solve this problem. The authors of this work propose using two cameras, and propose a new stereo matching algorithm based on synergetics. In template matching, a sequential similarity algorithm and block correlation matching [80] are representative. They are simple but their matching precision is low. DP matching is good [81] for simple stereo images due to its simple algorithm and matching precision, but cannot process stereo images including occlusion and reversal position well.

The authors propose 1) a stereo matching algorithm based on synergetics, called a synergetic stereo matching algorithm; 2) a stereo matching algorithm using two-stage DP matching; and show 3) experimental results for measuring matching precision. They also present the results of template matching. In stereo matching problem, an input pattern vector, formed of the edge points of the first image, is assigned to q, and the k-th prototype pattern vector, including the edge points of the other image, to v_k. As usual in the implementation of standard synergetic models, the pixel information (raster graphics) of the underlying geometry is used in this work as well.

The stereo matching algorithm based on synergetics described in this work has high matching precision. In this sense, the method proposed by authors of the work is quite effective. However, this method also has the shortcoming of long computational time. To improve computation efficiency, the authors analyzed order parameter equations and obtained the knowledge on the convergence value of the order parameter. They also presented the stopping condition for the iteration of numerical calculations. This reduced computational time by about 60% without losing matching precision. They are expecting a further reduction of computational time by applying Eular method to numerical calculations.

5.3.2 Application of Synergetic Neural Network in online writeprint identification

The synergetic neural network (SNN) associates synergetics with the artificial neural network, it can rigorously deal with the behavior of a network in the mathematical theory, and has the advantage of fast learning, short pattern recalling time, etc.

In article [82], a pattern recognition method based on the self-adaptive attention parameters presented on the basis of analyzing the key technology of SNN, and the advanced algorithm is employed in the online writeprint identification. The key point of this algorithm is that it can correct initial misidentified patterns through measuring similarity between the prototype pattern and the testing pattern in the evolution of order parameters. Experimental results of this research showed that the advanced SNN has better performance and robustness than the SNN based on balanced attention parameters. Further, the network's self-learning ability and recognition performance are greatly improved by using advanced SNN.

This research concerns the methods of the application of SNN to online writeprint identification. In the preprocessing step, abundant linguistic features are extracted from 2500 samples, and then different features are combined to form the most representative global feature for each author's pattern of writeprint. In the identification step, the method of optimizing the SNN using the dynamic attention parameter is proposed. Writeprint [83] means stylistic features which represent the writing style of authors hidden behind online messages, including character n-grams, punctuation, special symbols, digits, high-frequency words, word and sentence-length distributions, function words, content words, and part-of-speech tag n-grams distribution, etc [84]. These abundant features are combined into a global feature, so the problem of writeprint identification is transformed into the matching problem of different global features. The authors find that the similarity in global features which represent different authors' patterns is great, therefore, reduction of the correlation among different authors' global features and improvement of self-adaptive ability in the identification using SNN have become a critical issues in writeprint identification.

The proposed online writeprint identification system based on improved SNN achieved better recognition performance and robustness than the standard SNN. By using the dynamic method, the system obtained fairly strong self-learning ability as it can adaptively change the value of attention parameters by measuring similarity between the prototype pattern and the identified pattern. This research showed that the values of λ_k, B and C greatly affected the identification performance, but so far, there is no specific guidance to the values of parameters in SNN. Therefore, it is intended to study the values of λ_k, B and C in the future to achieve better performance and robustness in online writeprint identification. For more information on this work, refer, e.g., to [82].

6 AUTONOMOUS DESIGN SYSTEMS (ADS)

The talk below is intentionally structured into separate chapters because of the importance of the notion of ADS. Basically, it would also suit into chapter 4 along with the sections for modeling the creative parts of the engineering design process and for the philosophical analysis of these theories.

In this section, ADS and the related issues are discussed and a short overview of the cybernetic models, which are suitable to combine with synergetic methods, is given. Note that these models are novice and they are still under research. Initially they were intended to be used as standalone frameworks for creative components of Autonomous Design Systems (ADS).

The formal definition of the notion of ADS is given (formulated by the author of this thesis) and the sufficient properties of ADS are listed.

6.1 Definition

Comprehensive research on the subject of self-organization and synergetics allowed coining the notion of Autonomous Design System. Synergetics being a rigorous mathematical discipline, made it possible to apply the top-down approach for the whole research process organization. Interestingly enough, the author was first interested in the philosophical implications of the theory of self-organization, therefore, a hypothesis was raised about the possibility of modeling creativity by means of exploiting self-organization principles. Then the mathematical theory of self-organization, i.e., synergetics was discovered and analyzed. Therefore, the creative part of the engineering design was modeled by applying the synergetics principles, and the novel notion of ADS was introduced. As a result, a working prototype of such a system was created, tested (on the AutoCAD platform) and documented.

It is the mathematical properties of synergetics that allow to realize the whole system, meaning that initially we theoretically comprised the system with all of its needed properties, and the methodical implementation of synergetic tools eventually lead to the realization of the working system. Thus, work was started from the top with the description of the characteristics of the model and finished with a fully functional numerical and programmable model.

ADS is defined as an advanced CAD system, which has AI functionality and particularly the functionality to solve the creative tasks of the engineering design process.

ADS is opposed to the conventional CAD systems, (see, e.g., [85]) which normally automate routine parts of the design process and generally have no AI capabilities.

The introduction of the notion of ADS is an important contribution of this work. ADS is a completely new notion and it has never appeared before in the scientific literature. The introduction and analysis of this term initiates a completely new direction in research activities focusing on the problems of modeling the creative components of the engineering design in general, and

HVAC design in particular. The work related to ADS allows for introducing the world of synergetics and self-organization into the engineering domain, and acknowledges this domain as a complex system having its own intrinsic laws and principles, which must be further researched and formalized.

6.2 Properties of ADS

The basic properties or characteristics of ADS allow for distinguishing this kind of design systems from other, conventional CAD systems. These properties are as follows (as of the moment, probably along with the further evolution and development of ADS, these properties will also be complemented).

- ADS must include *AI technology modules/characteristics*. As ADS is an advanced CAD system, which has AI functionality, it must have the appropriate modules, which use AI. The examples of AI are synergetics, software agents, ANN, SNN, etc.
- ADS has the *ability to deal with the creative parts of the engineering design*. This is the crucial characteristic of ADS. Without this, the system will not classify as ADS, regardless of how intelligent it really is. See, e.g., section 4.4 for a description of the meaning of the creative part of engineering design.
- ADS is capable of *performing some computational sequences* (preferably those representing the creative parts of the design) *without direct supervision and guidance of the operator* (engineer) or, in other words, *functioning autonomously*.
- The *user interface* of ADS must be *simple, easy to learn/use and intuitive*. It has to aid the designer in doing his/her daily work and not to overwhelm with extra information. Probably, its behavior has to be preventive and thoughtful. Software agents' technology seems to be perfectly suited for assuring this property.
- Usually, ADS *are internally* quite *complicated* systems and are often based on existing CAD (BIM) frameworks. Such frameworks are for example AutoCAD, Revit, ArchiCAD.
- Finally, they *must have some useful functionality* that can be applied in engineering (e.g., HVAC) domain and addressed to solve a particular engineering problem. This property is shared with classical CAD systems, and thus, generally not unique to ADS.

6.3 Examples and discussion

The following is a detailed discussion about possible implementation of synergetics and self-organizing systems' properties in various CAD scenarios. As mentioned earlier, introducing the synergetics theory into a CAD system means (or could lead to) essentially constructing ADS. Of course, the resultant system must have the sufficient properties of ADS listed above. Synergetics is used as the main tool for modeling the creative component of engineering design in CAD systems. The following is an explanation of how conventional CAD systems may benefit from synergetics and self-organization and could eventually become successful candidates for ADS.

A successfully constructed synergetic model, which operates according to an internal purpose, is a kind of a self-organizing system. Then an external agent can be introduced that performs reasoning upon the internal characteristics of the system in order to construct an external conclusion or view, thus achieving an external purpose (interpreting the internal information). By doing that a compound self-organizing system (i.e., synergetic model) is created, which has an external goal. Constructing such a system is a relatively straightforward activity since all (traditional) cybernetic and systems engineering models function according to the external purpose. This leads to an assumption that the benefits of self-organizing systems can be used in CAD applications and in ADS.

A. Modeling (reasoning) by analogy

This activity conforms to the evolutionary theory of systems development (including, e.g., culture, human society). It happens that the human brain functions largely in the same way. Moreover, the “learning” processes of the majority of biological species are based on the principle of analogy. The idea is to try to create a model of such a learning system. The model could, in general, function as follows. Some cases/situations are presented to the system from which it can learn, i.e., acquire some information. The system (model) remembers this information, and then, in the future, it will be capable of not only finding the exact learned cases but also analogical cases. To accomplish this, the system must have some reasoning mechanism that allows recognizing analogies in the information presented/surveyed. In order to improve the model (in the sense of self-organization), the “historical component” can be added. The system remembers the case in the historical context, in real time; with the characteristics of the environment such as time, the source of the information, etc. It is the system that takes into consideration the initial conditions of the process of information acquisition. It is important to underline that these initial conditions are not arbitrary as in conventional (classical) cybernetic models. It could also be possible to put the system under conditions of strong non-equilibrium in order to stimulate the emergence of creativity. This way a synergetic model can be obtained which, in addition, functions similarly to the majority of biological systems on Earth, including humans and the functioning of human brain (in creativity context?).

B. Dreams modeling

The idea of modeling human dreams comes from the fact that sometimes the products of creativity arise during sleep, when dreaming. Although the dreaming mechanism is not very well known yet, it is possible to model it at the most primitive level of abstraction. Namely, it is suggested that dreams are composed of previously acquired information, of the interpretations of previous experiences, and of the combinations of this data. The exact mechanism of the combining process is still unknown, but at the simplest level it is asserted to be a random combinatorial activity. In that case, it is possible to model that combining process by means of traditional computing (IT). Assumption: a dream may consist of entities previously known by the system. An algorithm has to be created to combine these entities (possibly fuzzy, random), and then, a mechanism for interpreting these combinations. Some of the combinations may be useful in the system's work. Thus, it can be said that dreams modeling is a kind of combinatorial (random) activity on some known information segments, which may, hopefully, lead to useful combinations of data that could be considered as a product of creativity. Again, it is possible to add to the initial model the properties of a self-organizing system to optimize it and to improve the system's performance. Principles of synergetics can also be used, in combination with other non-classical branches of science, for example with memetics. In [86] such an approach was used to explain and create the alternative theoretical base of a given musical system. Memetics is a theory of mental content based on an analogy with Darwinian evolution. It purports to be an approach to evolutionary models of cultural information transfer. A meme, analogous to a gene, is an idea, belief, pattern of behavior, etc. which is "hosted" in one or more individual minds, and which can reproduce itself from mind to mind. Thus what would otherwise be regarded as one individual influencing another to adopt a belief, is seen memetically as a meme reproducing itself.

A little philosophical digression that is connected with the ideas discussed in this section/part may be useful here (of course it may also reside in sections 4.4 and 4.5 as well).

We suspect that the algorithm of the combinations may be affected by *emotions*, which the system should also have (in pursuit for creativity). Thus, emotions have to be modeled as well. Yet emotions are driven by some different mechanisms (e.g., goals, principles, good/humane considerations), which need to be imitated as well, etc. In the end, a whole system that mimics the creative creature – the human - has to be created. Then the question arises, why the human being (in the creativity context) should be modeled when original creative humans exist, and the possible model ever created would always be worse than the original. The answer, as it seems today, is that in the model the negative features of the human nature can be minimized, and the good ones, on the other hand, can be articulated, further increasing productivity and, what is very important, to direct its flow in a humane, positive direction. The artificial system created could then concentrate solely on the given problem (i.e., creativity in engineering design) and use all available computational resources (compared to ordinary human activity, where only a small part of mental potential is used), and in that fashion it may compete with real human creators, and hypothetically, even beat them in the future.

C. Software agents

Another AI technology that may be further improved using the synergetic approach is autonomous software agents. This type of computational model is suitable for modeling the behaviors of social systems. As self-organization of society is connected with the freedom of individuals, a system of relatively independent (free) software agents can be used to model a self-organizing system here. The software agents technology can be successfully combined with other AI technologies (neural networks, genetic algorithms, e.g., [87], etc.) in order to improve a system even more. In [47], for example, the first attempts were made in this direction, and a suite of independent software agents was developed that run on the AutoCAD platform. Software agents were connected with Artificial Neural Network (ANN) in the form of a separate agent that had ANN functionality and that was capable of being trained in a particular way. See section 7.2.4 for the introduction of using software agents under the HVAC CAD framework. (For an overview of the software agents technology and its hybridization possibilities, see [47].)

Then the characteristics of self-organization, mentioned above, should be included into the resultant system to get a candidate for a successful synergetic model, and consequently for ADS.

For example, in the present work the ADS HVAC system which has been built upon an existing AutoCAD platform is elaborated. This system has the advanced synergetic computer algorithm implemented, which is responsible for modeling the creative behavior of the system. See next sections for detailed information on this particular implementation. It is classified under the ADS class due to its ability to deal with the creative part of the design. Namely, the system automates the non-routine task of visual identification and recognition of outer peripheries of a given building. The identification process is based solely on the visually identifiable vector geometry objects. In such a fashion, it resembles the visual perception of biological species including the human being himself. Thus, it is very hard to say that this is a routine task functionality. Although this is a relatively simple ADS model, especially in the context of the possible features discussed here, this implementation has all the other properties required for successful ADS classification.

The model developed serves primarily as a prototype system and as a base for further developments and ADS features introduction. The possible ways of developing the model are highlighted in this section and in other parts of the dissertation, especially in sections 4.5 and 4.4.

This ADS implementation can be considered a successful experiment, which showed good results. Yet again, this is the first attempt to model the creative part of the engineering design process (in the HVAC domain, in this case) and to create a working prototype of ADS. During the development process of the system and the synergetic computer theory adaptation and implementation on the CAD platform, some minor difficulties arose, and at some points it really seemed that the successful practical realization of the system is seriously endangered. Nevertheless, these difficulties were successfully overcome, and eventually, a fully functional system was built which is in accordance with our initial intentions. For a detailed technical discussion of the adaptation and implementation/development processes, please refer to chapters 7 and 8 respectively.

7 ADAPTATION OF THE STANDARD MODEL FOR HVAC ADS

7.1 Experiments

The adaptation of the standard model of synergetic computer to HVAC ADS is performed in several steps. First, the theoretical base is prepared and the numerical prototype of the system is created in MATLAB. Then, the system is adapted to the general AutoCAD environment and functionality. Finally, the advanced algorithm is elaborated and tested as part of a particular synergetic computer HVAC application on AutoCAD, i.e., HVAC ADS.

The adaptation process of the standard model of synergetic computer to AutoCAD, and more specifically to the HVAC ADS environments, consists of two parts: theoretical and practical ones. The theoretical part involves deduction of the appropriate algorithm or method, and the practical part means testing that algorithm or method in a specific computational environment, i.e., conducting experiments.

This subsection is devoted to the description of both the practical and theoretical parts of the adaptation process. While this section deals specifically with the experiment description, some coverage of the related tests is also included, for instance, in the Results subsection, where, e.g., the optimization of the permutation algorithm is described in 7.2.3. As a result, the final division of covering the experimentation, results, and implementation is not so strict, meaning that some data are introduced at places more appropriate for enhancing the general understanding of the subject and not according to formal rules.

The description of the already working, debugged prototypes is given in the implementation section of this work (see section 8).

The *results* of the adaptation experiments are summed up in the subsections below (see subsection 7.2).

7.1.1 Constraints of vector graphics

The standard Haken model of the synergetic computer is well suited for the raster graphics identification. See the next subsection for a brief overview of the general characteristics of raster and vector graphics. In [45], for example, all the corresponding theory is explained, and some possible examples of the implementation of this theory in the identification of raster graphics images are provided. In such applications, when an image has to be recognized, e.g., a photograph of a face, a grid can be laid over it in order to digitize the whole image. The individual cells or pixels are labeled by numbers $j = 1, 2, \dots N$, then a further number v_j is attributed to each pixel indicating its tone of grey/color number. Next a vector is formed:

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}. \tag{7.1}$$

Because a whole set of different faces has to be stored in the computer, they are distinguished by a label *k*, so that *v* is now replaced by the prototype vectors

$$v_k = \begin{pmatrix} v_{k1} \\ v_{k2} \\ \vdots \\ v_{kN} \end{pmatrix}. \tag{7.2}$$

It is assumed that v_{jk} is real. The label *k* may adopt the values $k = 1, \ldots, M$, where *M* is the number of stored patterns. It is assumed that the number of patterns is smaller than or equal to the number of feature

$$M \le N$$

(for more assumptions, see section 7.2.1).
If not stated otherwise, the vectors v_k are subjected to the condition

$$\sum_k v_k = 0 \tag{7.3}$$

which can always be achieved by forming

$$v_k = \tilde{v}_k - \frac{1}{N} \tilde{v}_{jk} \tag{7.4}$$

where $\tilde{v}_k$ are the “raw vectors” and *N* is the number of features per pattern. Additionally, a transposed vector is needed; this vector is defined by

$$v_k^T = (v_{k1}, v_{k2}, \ldots v_{kN}). \tag{7.5}$$

It is assumed that the following normalization holds

$$(v_k^T v_k) \equiv \sum_{j=1}^{N} v_{kj}^2 = 1. \tag{7.6}$$

This can always be achieved by dividing the “raw vector” $\tilde{v}_k$ by $(\tilde{v}_k^T \tilde{v}_k)^{1/2}$. Because the vectors v_k are not necessarily orthogonal to each other, adjoint vectors (occasionally also called *pseudo-inverse*) are needed

$$v_k^+ = (v_{k1}^+, v_{k2}^+, \ldots v_{kN}^+), \tag{7.7}$$

these vectors obey the orthonormality relations

$$(v_k^+ v_{k'}) = \delta_{kk'}. \tag{7.8}$$

The adjoint vectors v_k^+ are represented as a superposition of the prototype vectors $v_{k'}$

$$v_k^+ = \sum_{k'} a_{kk'} v_{k'}^T, \tag{7.9}$$

where $a_{kk'}$ are constants and $v_{k'}^T$ is the transpose of $v_{k'}$.

In the beginning, this ready model seems quite obviously suitable for our goal as well. The problem, however, is that AutoCAD software is based on vector graphics. Thus, a way to work around this constraint has to be found, or, as an alternative option, the recognition model can be adapted to the vector graphics environment. For instance, one theoretical possibility to apply the raster graphics model to AutoCAD is to use the fact that AutoCAD seems to use some sort of raster graphics, e.g., in the preview image window in the Open Drawing dialog. This must mean that AutoCAD software internally converts the vector graphics data from the drawing database to a bitmap image to display. It also means that the appropriate functions and functionality are probably exposed in the ObjectARX library, and it is possible to convert the vector graphics data to raster graphics data. Be it noted that we actually do not investigated these opportunities in depth because this approach is not computationally efficient, on the contrary, it is complicated and unwieldy. Therefore, this approach is decided to use as plan B, and first, finding a way of *adapting* the mathematical model of the synergetic computer to the vector graphics object recognition task is opted for. Thus, theoretical analysis has been performed and several ways to address the problem proposed. It is decided to start from the simplest one, i.e., from the assumption that a vector graphics object can be coded through vectors representing its geometry by the distances between the object's characteristic points (as in 7.2.2). Thus, eventually, every graphical object can be represented as a vector of certain length. This, of course, is a quite primitive representation, but sufficient for our final purpose, i.e., for the room geometry identification. This is due to the fact that the majority of a building's rooms have usually simple geometry, and in most cases, this is either square, rectangular or a combination of these.

Thus, this model is theoretically tested, then described in MATLAB, and tested numerically. As the results are good, work proceeds with testing the synergetic model and the implementation of its basic functionality in AutoCAD as described further in section 7.2.2.

7.1.1.1 Raster versus vector graphics

A *raster graphic* or image is made up of pixels. Pixels are small squares of information. In order to understand what a raster graphic is, a large grid that is made up of many squares (pixels) all of the same size can be imagined (see Fig. 7.1). If a different color is put in each square, then, backing away from the grid, the individual squares blend together to make up a picture. This is a raster graphic or bitmap graphic. Raster graphics are perfectly suitable for rich, full-color images such as photographs. Raster graphics are rendered images on a pixel-by-pixel basis and they are useful when handling shading and gradients.

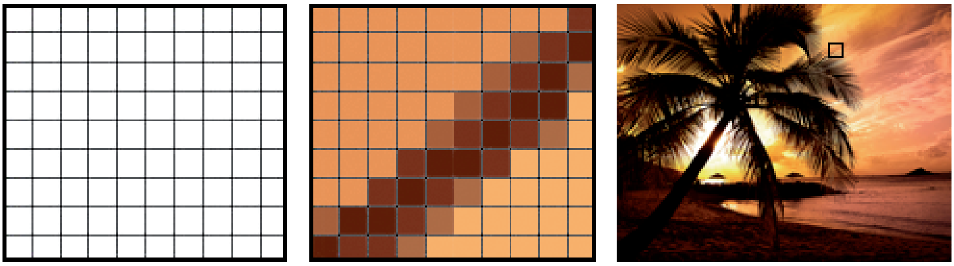


Fig. 7.1 An example of a raster graphics image

Advantages of raster graphics:

- Raster graphics are most appropriate when creating rich and detailed images. Every pixel in a raster image can be a different color, therefore, a complex image with any kind of color changes and variations can be created.
- Almost any program can work with a simple raster file. The most recognized application that handles raster graphics is Adobe Photoshop, but there are also several other image editing software options to choose from.

Disadvantages of raster graphics:

- Raster images cannot be scaled up in size very well. If a raster image is enlarged, it will look grainy and distorted. This is because raster images are created with a finite number of pixels. While increasing the size of a raster image, as there are no longer enough pixels to fill in this larger space, gaps are created between the pixels in the image. The photo editing software used will try to fill these gaps the best it can, however, the resulting image is often blurry.

- Raster graphics files are often quite large. Raster graphics files contain all the information for every single pixel of the image that is worked with. Each of these pixels has an X and Y coordinate as well as color information associated with it, therefore, raster graphics files tend to be very large. Besides, as these raster images hold so much data, they may be slower to edit.
- Raster graphics are not very suitable for embroidery. Because raster images are based on square pixels, embroidery may look like it has jagged edges. If an embroidery of an image with smoother edges is desired, it is best to use vector graphics instead of raster graphics.

Vector graphics are based on mathematical formulas. A vector graphic is made up of a series of small points that combine together to make lines and images (see Fig. 7.2). The most recognized applications which handle vector based graphics are Adobe Illustrator, Macromedia Freehand and Corel Draw form the image processing side and AutoCAD from the field of technical drawing. Vector graphics are generally used for line art, illustrations, engineering drawings (CAD) and embroidery.

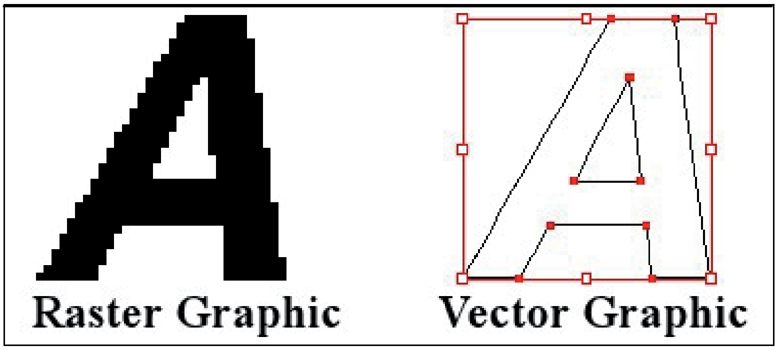


Fig. 7.2 Raster graphics versus vector graphics object representation

Advantages of vector graphics:

- Vector files are small because they contain a lot less data than raster files.
- Vector graphics are more flexible than raster graphics because they can be easily scaled up and down without any loss to the quality of the image.

- Vector graphics have smoother lines when compared to square, pixel-based raster graphics, therefore, they are better with straight lines and sweeping curves than raster graphics.

Disadvantages of vector graphics:

- If there are small errors or faults in a vector graphic, these will be seen when the vector image is enlarged significantly.
- Vector graphics are generally filled with a solid color or a gradient. They cannot display the lush color depth of a raster graphic.

7.1.2 Order parameter generation

In our basic system's prototype testing in MATLAB and in AutoCAD, the dot product based order parameter (OP) generation is used:

$$\xi_k = (v_k^+ q),$$

where q is the state vector of a test (input) pattern with the initial value q_0, ξ_k is the order parameter, v_k is the prototype pattern vector, and v_k^+ is the adjoint vector of v_k.

This OP generation algorithm conforms to the standard synergetic computer model. In the MATLAB and AutoCAD implementations three test/prototype patterns are used for recognition. The order of the vector elements that correspond to the pattern always stays unchanged.

The experiments, both in MATLAB and in AutoCAD showed satisfactory results. That is, the maximum OP generated always corresponded to the right winning pattern. The small number of test/prototype vectors have not allowed to pinpoint the weaknesses of the standard dot product based OP generation method. As the results obtained so far have been good, this OP generation method was adapted for the next task, i.e., for room geometry identification as a part of HVAC ADS.

In this task, due to the AutoCAD database representation and our goal constraints, the possible room's characteristic vector values have to be permuted, i.e., the order of the vector elements is constantly changing. For example, in the case of just one vector consisting of three elements there are $C_k^r = \dfrac{k!}{(k-r)!} = \dfrac{3!}{(3-3)!} = 6$ permutations. In this equation, *C* is the number of permutations, *k* is the total number of the elements in the permutation set, *r* is the number of elements in the selection set participating in a given permutation. Thus, *C* may be referred to as the number of *r* permutations out of *k*. Actually, in the room geometry recognition scenario, a total number of 22 AutoCAD graphic objects or entities and a total number of 44 different points has to be processed, see Fig. 7.3. The test/prototype vector's length is chosen equal to 3. This gives $C_k^r = \frac{k!}{(k-r)!} = \frac{44!}{(44-3)!} = 79{,}464$ permutations.

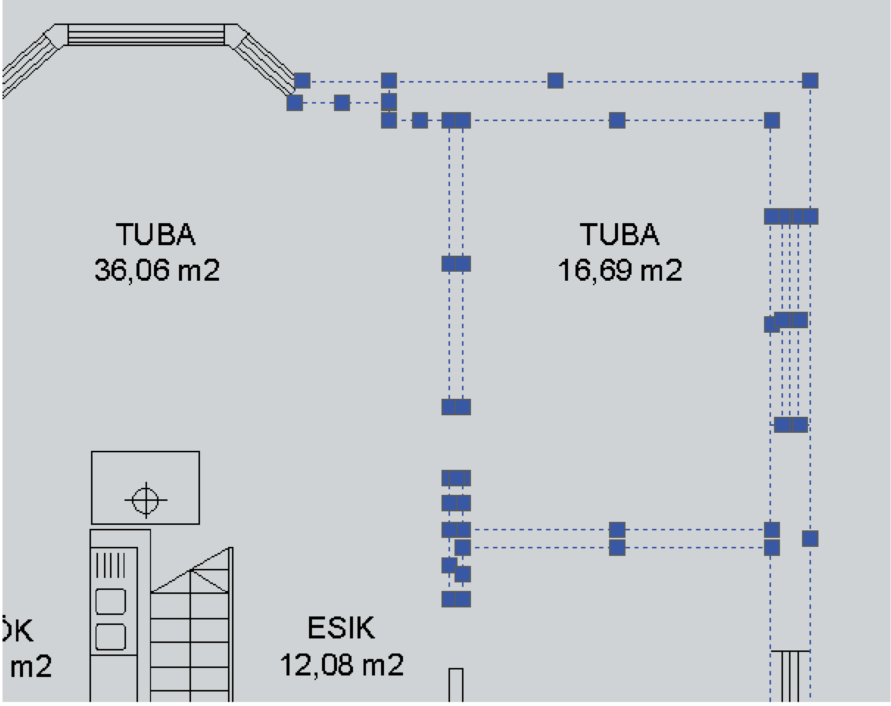


Fig. 7.3 Permutation objects/points in room geometry recognition task

First, the effect of the dot product OP order of the vector elements is not noticed, so the conventional dot product based OP generation method is used for the task. It is no surprise that among 79,464 different order parameters several are found, whose absolute value exceeds our prototype vector's indicative OP value. This repetitively results in wrong pattern identification. By gradually reducing the number of elements/points participating in permutations, the cause for the system's erroneous behavior is eventually discovered. For example, reducing the number of participating points to 18, means the existence of at least 4 different permutations of OP values of greater or comparable magnitude than in the real prototype pattern. The next challenge is to correct the behavior of the system.

As our approach in the attention parameters domain is based on the standard model, too (i.e., the balanced attention parameter λ_k), the first possible resolution seems to be the introduction of the unbalanced attention parameter technique. The balanced attention parameter means that all λ_k are the same and equal to, e.g., 1, i.e.,

$$\lambda_k = \lambda > 0,$$
$$\lambda = C = 1.$$

This means essentially that the attention parameter does not influence the magnitude of the order parameter, and consequently, the results of the identification process. When the attention parameters λ_k that correspond to the given test/prototype pattern are not equal, they are called unbalanced attention parameters. In this case, the final value of the winning order parameter is determined by $\xi_k = \sqrt{\lambda_k}$, while in the case of balanced attention parameters it is $\xi_k = \lambda_k = 1$. The main challenge here is to find a suitable attention parameters selection mechanism in order to determine the values of λ_k. This leads to searching the possible similarity measures between comparable test and prototype vectors. For example, the most primitive way to determine the degree of similarity between vectors may be as follows:

$$v_k - q_0 = X,$$

where v_k is the prototype vector and q_0 is the test vector. Then, based on the magnitude of the value of X, the corresponding λ_k value is selected. The principle of selection of the value of the attention parameter is inverse proportionality, i.e., the smaller is X, the bigger λ_k is selected. Using unbalanced attention parameters may mean that initially the smaller order parameter ξ_k eventually wins the competition. This is because in the case of an unbalanced attention parameter, the behavior of the system is determined by the relationship between the attention parameters and the order parameter. It is also possible that unbalanced attention parameters are of dynamic and/or adaptive nature. This means that λ_k are changing their values according to the given computational rule. This research has led us to the discovery of the cosine similarity measure between two vectors. While this measure may be used for determining the values of unbalanced attention parameters (through the introduction of an additional selection procedure), we find that this measure works well for determining the winning order parameter (OP) directly as well, and thus, the additional numerical computations related to λ_k selection may be omitted. Eventually, these tests are abandoned in favor of a simpler and more efficient approach, which is described in more detail below.

The next possibility is to introduce the additional competition between the order parameters belonging to the set of its maximum values. In this approach, it is very important to decide the size of the set, because depending on the number of possible permutations, the appropriate size of this set that includes the indicative OP value is varying. For instance, in our particular problem, if the number of participating points is 18, then k = 18 in $C_k^r = \frac{k!}{(k-r)!}$, and it is sufficient to select the size of the set of maximum OP values equal to n = 4. If, however, the number of participating points k = 38 , as many as n = 10 maximum OP values are needed, meaning that choosing, e.g., n = 8, the OP values that correspond to the correct prototype vector (pattern) are simply not included in the set of additional competition. Tests have shown that performing the additional competition is not computationally efficient (the selection of maximum OP value out of 79,464 different order parameters is a very processor intensive task), and as the size of the additional competition set grows, the system quickly becomes unwieldy.

For this additional competition, a new method of OP generation, compared to the classical Haken model has been found. Namely, OP generation based on *cosine similarity* (or correlation coefficient) measure between two vectors is introduced:

$$\xi_k = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\sqrt{\sum_{i=1}^{n}(B_i)^2}}.$$

Here ξ_k is the order parameter, and A and B are the vectors being compared. This formula is tested on a set of maximum OP values and is found to work well for our particular case.

Due to the computational complexity of the additional competition process, the initial competition, or more specifically the generation of the initial order parameters, is substituted by this new method. The results of this are very impressive: the right patterns are correctly identified within all of the tested permutations, including the complete permutation set (equals to 79,464 permutations). The overall performance numbers of the resultant process are better, too, for more details, see next subsection (7.1.3).

This is the most prominent *modification* of the classical synergetic computer algorithm that expands the functionality of the synergetic computer to the domain of vector graphics with all of its different applications in, e.g., CAD, AutoCAD and ADS.

Indeed, this quite simple modification has led to positive results in our experiments, and means that our hypothesis about the possibility of using the synergetic computer in ADS, and particularly, in modeling the creative tasks of the engineering design process has been confirmed. Actually, if this new OP generation method had not been found, we would eventually have ended up with a negative result, and confined ourselves to the description of the experiments and assertion that further research on the subject is needed to investigate this hypothesis in more detail.

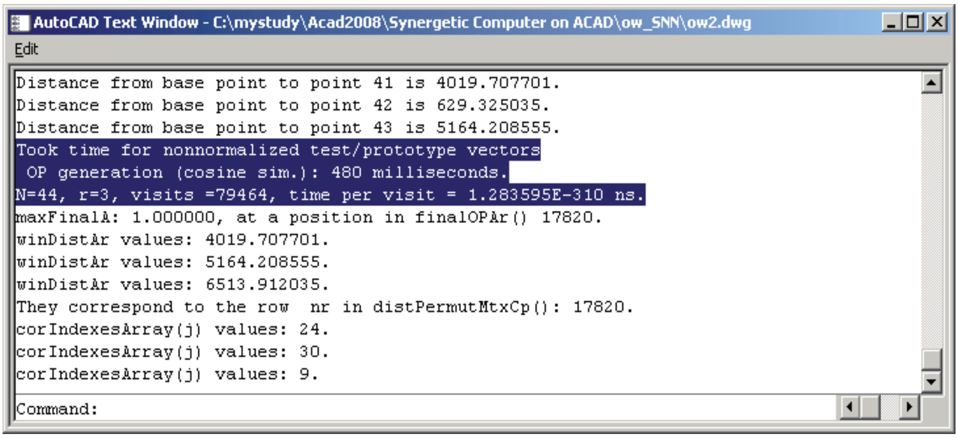
7.1.3 Some performance issues

In this subsection, the performance issues that occurred during the implementation process of our systems are discussed. The software used in these experiments is listed and covered in more detail in section 8.

7.1.3.1 Normalized versus non-normalized values of test/prototype vectors

Tests have shown that order parameter (OP) generation performance is approximately the same for both normalized as well as non-normalized OP. We get 480 ms (milliseconds) in execution time for non-normalized values and 470 ms for normalized values. Note that we use the Debug configuration and all performance numbers are valid for that configuration (in Release configuration the performance is significantly better, e.g., for non-normalized values the execution time is only 20 ms; we have intentionally opted for larger numbers to articulate the difference between them). However, the cost for the generation of normalization itself is 70 ms. This means that the total processor time for normalized values is 540 ms, which is greater than for non-normalized values. Thus, for our case (N=44, r=3, 79,464 permutations), for particular distances, OP generation based on non-normalized vector values is selected as the more efficient computational algorithm. AutoCAD text window for non-normalized vector values is shown on Fig. 7.4.

The computational environment's parameters are as follows:
OS Name - Microsoft Windows XP Professional
Version - 5.1.2600 Service Pack 3 Build 2600
System Manufacturer - IBM
Processor - x86 Family 6 Model 9 Stepping 5 GenuineIntel ~1495 Mhz
Total Physical Memory - 512,00 MB.
AutoCAD Architecture 2008 SP1, ObjectARX 2007, Boost R 1.49.0, Eigen 3.1.0, built with MSVC 8.0 compiler.



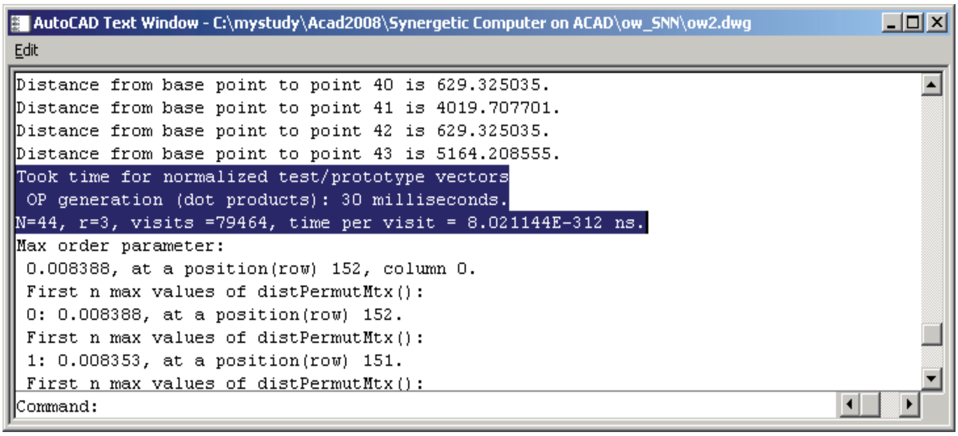
```
Distance from base point to point 41 is 4019.707701.
Distance from base point to point 42 is 629.325035.
Distance from base point to point 43 is 5164.208555.
Took time for nonnormalized test/prototype vectors
 OP generation (cosine sim.): 480 milliseconds.
N=44, r=3, visits =79464, time per visit = 1.283595E-310 ns.
maxFinalA: 1.000000, at a position in finalOPAr() 17820.
winDistAr values: 4019.707701.
winDistAr values: 5164.208555.
winDistAr values: 6513.912035.
They correspond to the row  nr in distPermutMtxCp(): 17820.
corIndexesArray(j) values: 24.
corIndexesArray(j) values: 30.
corIndexesArray(j) values: 9.

Command:
```

Fig. 7.4 Program output messages: performance of order parameter (OP) generation in the case of non-normalized vectors values

7.1.3.2 Order parameter (OP) generation based on dot products versus cosine similarity algorithms

OP generation based on dot products algorithm (1) takes only 30 ms, see Fig. 7.5. This algorithm utilizes normalized vectors/matrices, thus, another 70 ms has to be added for the normalization procedure to obtain fair comparison conditions. The OP generation process based on cosine similarity (also known as the correlation coefficient) algorithm (2) takes 480 ms. These results are quite obvious, since (1) consists essentially of a single operation ($A \cdot B$), and in our implementation the matrix and the column vector products are used to obtain the vector of dot products, using efficient Eigen (linear algebra) library (Fig. 7.6). Although in the Boost package exist even more precise timers than *boost::chrono::system_clock*, this timer is chosen intentionally, because we are not interested in the exact duration of the event, but the difference between the events' execution time. The function of performance measurements is rather to determine which of the approaches should be preferred to the other from the computation time point of view than to measure the absolutely precise duration of this.



```
Distance from base point to point 40 is 629.325035.
Distance from base point to point 41 is 4019.707701.
Distance from base point to point 42 is 629.325035.
Distance from base point to point 43 is 5164.208555.
Took time for normalized test/prototype vectors
 OP generation (dot products): 30 milliseconds.
N=44, r=3, visits =79464, time per visit = 8.021144E-312 ns.
Max order parameter:
 0.008388, at a position(row) 152, column 0.
 First n max values of distPermutMtx():
 0: 0.008388, at a position(row) 152.
 First n max values of distPermutMtx():
 1: 0.008353, at a position(row) 151.
 First n max values of distPermutMtx():
Command:
```

Fig. 7.5 Program output messages: performance of order parameter (OP) generation in the case of normalized vectors values and dot product OP

```
//performance measuring
using namespace boost::chrono;
boost::chrono::system_clock::time_point t1 =
boost::chrono::system_clock::now();
//multiply to get the vector of dot prods
distPermutMtx*=q0;
//distPermutMtx now should hold dot products
   //for (int i=0; i<8; i++)
   //acutPrintf(_T("\ndistPermutMtx values
(dot.prod.): %f."), distPermutMtx(i, 0));
system_clock::duration d  = system_clock::now() - t1;
acutPrintf(_T("\nTook time for normalized
test/prototype vectors\n OP generation (dot products):
%u milliseconds.\nN=42, r=3, visits =79464, time per
visit = %g ns. ")
          ,duration_cast<milliseconds>(d).count(),
duration_cast<nanoseconds>(d).count()/79464);
```

Fig. 7.6 Code snippet showing the use of dot product OP generation by matrix/vector multiplication

On the other hand, approach (2) requires at least five computational operations to obtain the OP based on cosine similarity:

$$\xi_k = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}}.$$

Eventually, since, to fulfill our goal (as seen above, approach (1) does not give satisfactory recognition results in our case) approach (1) needs to be combined with approach (2), the processor times for the approaches have to be summed up. Thus, approach (2) proves to be computationally more efficient (30+70+480 ms versus 480 ms).

7.2 Results of experimentations and ways of further optimization

In this section, the results of the experiments of adaptation of the synergetic computer algorithm for the basic AutoCAD functionality, and for the one particular Autonomous Design System (ADS) functionality on the AutoCAD platform are presented. The results of the analysis of the use of synergetic computer algorithms are also presented here.

Some results of the author's previous experimentation related to the current research are also highlighted, and ways of further improvement of the researched systems in the sense of ADS (in the HVAC field) are discussed.

The system has been built capable of recognizing the building's rooms' outer peripheries in a manner close to human visual perception. While the algorithm implemented is based largely on H. Haken's models, significant differences to the basic characteristics of the classical model are introduced and successfully tested in one of the most popular CAD environments. For example, in our approach, the number of meaningful elements of the prototype pattern vectors is varied and the number of features per pattern is allowed to be smaller than the number of patterns (M≥N). In addition, the order parameter generation algorithm is completely different from the standard model and highly optimized for the AutoCAD environment.

7.2.1 Summary of the standard model of the synergetic computer theory

In this section, a short overview of the mathematical background of the synergetic computer concept is presented. This is done in order to facilitate the explanation of the synergetic computer algorithm's adaptation results presented in the following subsections. For more in-depth discussion of the standard model, see section 5 and [45].

We have analyzed the benefits and drawbacks of the synergetic computer approach and these are summed up in this section as well. This analysis forms the additional contribution of the research.

The basic dynamic equation of the synergetic computer or synergetic neural network is as follows:

$$\begin{aligned}\dot{q} = \sum_k \lambda_k v_k (v_k^+ q) - B \sum_{k' \neq k} (v_{k'}^+ q)^2 (v_k^+ q) v_k \\ -C(q^+ q) q + F(t),\end{aligned} \tag{7.10}$$

where q is the state vector of a test (input) pattern with the initial value q_0, λ_k is the attention parameter, v_k is the prototype pattern vector, v_k^+ is the adjoint vector of v_k, which obeys the orthonormality relation

$$(v_k^+ v_{k'}) = \delta_{kk'}. \tag{7.11}$$

B, C are positive constants and $F(t)$ describes fluctuating forces, which may drive the system out of its equilibrium state. Expression $v_k \cdot v_k^+$ acts as a matrix. This matrix has occurred in a number of other publications and is called the learning matrix. The term

$$\xi_k = (v_k^+ q) \tag{7.12}$$

is called the order parameter. The equation (7.10) describes the dynamics, which pulls the test pattern $q(t)$ into one of the prototype patterns v_{k_0}, namely the one closest to $q(0)$. This means the pattern is being recognized by the system.

The corresponding dynamic equation for the order parameters reads:

$$\dot{\xi}_k = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C \sum_{k'=1}^{M} \xi_{k'}^2 \xi_k. \tag{7.13}$$

The order parameters obey the initial condition $\xi_k(0) = (v_k^+ q(0))$ by which the initial values of order parameters in the evolution series are determined.

The equations (7.10) and (7.13) can be derived from corresponding potential function equations (7.14) and (7.15). That is

$$V = -\frac{1}{2}\sum_{k=1}^{M}\lambda_k (v_k^+ q)^2 + \frac{1}{4}B\sum_{k\neq k'}(v_k^+ q)^2 (v_{k'}^+ q)^2$$
$$+\frac{1}{4}C(q^+ q), \qquad (7.14)$$

and

$$\dot{\xi}_k = -\frac{\partial \tilde{V}}{\partial \xi_k},$$
$$\tilde{V} = -\frac{1}{2}\sum_{k=1}^{M}\lambda_k \xi_k^2 + \frac{1}{4}B\sum_{k'\neq k}\xi_{k'}^2 \xi_k^2$$
$$+\frac{1}{4}C(\sum_{k'=1}^{M}\xi_{k'}^2)^2. \qquad (7.15)$$

The potential function is used to represent the potential field in the space of k order parameters in which the fictitious particle, representing the dynamics of the test pattern or the corresponding order parameter, moves. An example of the potential V is shown on Fig. 7.7.

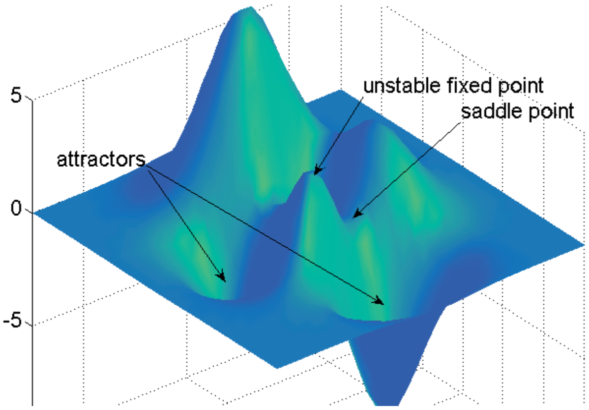


Fig. 7.7 Example of potential function

In this plot, the attractors are clearly visible. The attractors are the stable fixed points, which are represented by the bottom of each valley. The top of each mountain is an unstable fixed point. All points in the landscape from which the particle can roll down to the same attractors form the basin of attraction. Points of minimal height on ridges are saddle points.

The stable fixed points are at $q = v_k$, i.e., at the prototype patterns, and there are no other stable fixed points. The stable fixed points are equally characterized by $\xi_k = 1$, all other ξ's = 0.

The classical Haken model is built up upon a number of assumptions. The most important of them are as follows:

- all attention parameters are equal and positive (i.e., balanced attention parameters)

$$\begin{aligned} &\lambda_k = \lambda > 0, \\ &\lambda = C \end{aligned} \tag{7.16}$$

- the number of patterns is smaller than or equal to the number of features

$$M \le N \tag{7.17}$$

- vectors v_k are subject to the condition

$$\sum_k v_k = 0 \tag{7.18}$$

- the following normalization holds

$$(v_k^T v_k) \equiv \sum_{j=1}^{N} v_{kj}^2 = 1 \tag{7.19}$$

- the number of features per pattern should be the same for all prototype and test vectors (equality of vectors' meaningful dimensions). That is, for each

$$\begin{aligned} &v_1(k), v_2(l) \ldots v_n(m); \\ &k = l = \ldots = m, \end{aligned} \tag{7.20}$$

where *k, l,...,m* are vectors' meaningful dimensions.

As vectors v_k are not necessarily orthogonal to each other, the adjoint vectors need to be constructed, which may be formed as superpositions of the transposed vectors v_k^T:

$$v_k^+ = \sum_{k'} a_{kk'} v_{k'}^T. \tag{7.21}$$

The coefficients $a_{kk'}$ must be determined to satisfy the orthogonality condition (7.11). This may be done by multiplying (7.21) by $v_{k'}$ and interpreting $a_{kk'}$ and scalar products $(v_{k'}^T v_k)$ as elements of the corresponding matrices A and W

$$\begin{aligned} A &= (a_{kk'}) \\ W &= [(v_{k'}^T v_{k'})]. \end{aligned}$$

Equation (7.21) can then be written in the form

$$I = AW \tag{7.22}$$

and can be solved formally by $A = W^{-1}$.

7.2.1.1 Synergetic Neural Network

The synergetic computer may be realized by artificial neural networks, which act in a fully parallel manner. The resulting system is then called the Synergetic Neural Network or SNN. SNN may be realized, e.g., as a one- or three-layer network. By using the order parameter concept, the network can be considerably simplified. As order parameters are defined by (7.12), and satisfy

$$\dot{\xi}_k = \xi_k(\lambda - D + B\xi_k^2), \tag{7.23}$$

where

$$D = (B + C)\sum_{k'} \xi_{k'}^2, \tag{7.24}$$

then, for a three-layer network, representation of order parameters as neurons can be used in the network's second layer. The input layer is represented by input (test) pattern vectors $q_j(0)$, and if SNN has to act as an associative memory, the third layer should consist of

$$q_j(t) = \sum_k \xi_k(t) v_{kj}, \tag{7.25}$$

where q_j is the activity of the cell j at the output layer, ξ_k is the final state of order parameter cell layer with $\xi_k = 1$ for $k = k_0$ (i.e., the pattern has been recognized), and $\xi_k = 0$ otherwise. The network may be further simplified by introducing a common reservoir D as in (7.23). This way the number of connections may be further reduced.

7.2.1.2 Benefits of the synergetic computer (SNN) approach

The classical model has a number of advantages over traditional neural computers (networks). That is, compared to, e.g., Hopfield Neural Network (HNN), it has the following advantages.

1. The model training time is short.
2. The space complexity is low: for SNN it is np, where n is the number of features, p is the number of patterns and $p \ll n$, while for HNN it is n^2.
3. The processor time complexity for the recognition process is also low. For SNN it requires p^2 multiplications and p additions, and for HNN n^2 multiplications and n additions.
4. There are no so-called *pseudo-states*. This is the most important property. It may be proved that besides the prototype vectors there are no other attractors.

In HNN, the system has the following potential functions. In case of a discrete HNN:

$$V = -\frac{1}{2}\sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} w_{ij} v_i v_j + \sum_{i=1}^{N} \theta_i v_i$$

In case of a continuous HNN:

.

$$V = -\frac{1}{2}\sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} v_i I_i + \sum_{i=1}^{N} \frac{1}{\tau}\int_0^{v_1} g^{-1}(t)dt.$$

This potential function cannot guarantee that all of the attractors are actually the desired ones. In the construction of a HNN, no matter how carefully learning and adjusting the weights w_{ij} and threshold value θ_i is performed, it is still difficult to avoid/control the generation of a pseudo-state. In the Haken model, considering the system's dynamic behavior, the precise control of the potential function (energy function) in the energy potential field and not the type of connection of neurons nor the non-linear mappings of them, allow thereby to eliminate the pseudo-states.

5. The association effect: all of the prototype patterns can be clearly and equally identified.

7.2.1.3 Limitations of the synergetic computer

The limitations of the synergetic computer model, as it often happens, are extensions of its advantages. Namely, the most prominent disadvantage worth noticing is that the system always selects the winning pattern from among the patterns presented, even if they are not actually the right ones. This occurs due to the fact that the biggest order parameter formed initially as a dot product from among all available vectors will always win the competition. Therefore, if the vector set does not contain the right prototype pattern, it will not be recognized by the system. It is true, however, that the system will recognize the test pattern (vector) that is the closest to the prototype pattern (vector).

Some problems with correct identification may occur when the vectors have different orders of elements (features). This is because the inner (dot) product value's magnitude depends on the order of elements in the sequence (vector). Such identification difficulties may occur in case of testing different element combinations, e.g., in the points/distances permutation example (see onward in the next sections).

7.2.2 Synergetic computer algorithm on AutoCAD

In this section, the results of the experiments for adaptation of the synergetic computer algorithm and its basic functionality to the AutoCAD environment are presented.

The prototype patterns for the case of general AutoCAD testing are chosen from among AutoCAD (Acad) polygon entities (more specifically, these constitute polyline objects in the Acad database), namely, the triangle, square and hexagon. As Acad is vector graphics software, we had to invent the way of representing our prototype vectors properly. Vector elements are coded as a relative measure between entities' endpoints, i.e., the distance between the polygon's vertices, as shown on Fig. 7.8.

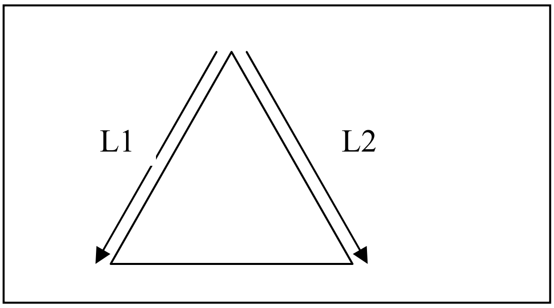


Fig. 7.8 Representation of prototype vectors' elements in the AutoCAD graphics system.

Thus, in the case of a triangle the raw prototype vector is as simple as $v_k = (L_1, L_2)$. We now have three state vectors to recognize (shown as raw vectors):

$$\begin{aligned} v_0 &= (L_{01}, L_{02}, L_{03}) \\ v_1 &= (L_{11}, L_{12}) \qquad (7.26) \\ v_2 &= (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}). \end{aligned}$$

From (7.23) we can deduce a discrete equation for the order parameter evolution

$$\begin{aligned} &\xi_k(n+1) - \xi_k(n) \\ &= \gamma(\lambda_k - D + B\xi_k^2(n))\xi_k(n), \end{aligned} \qquad (7.27)$$

where γ is the iteration speed and term D is in accordance with (7.24). The corresponding evolution for the case of three order parameters is shown on Fig. 7.9.

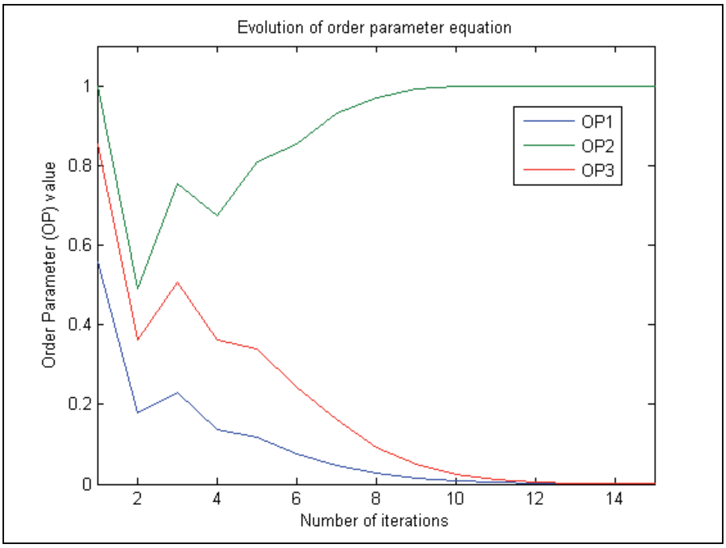


Fig. 7.9 Evolution of order parameter equations

From Fig. 7.9 it can be seen that the system converges after twelve steps, i.e., the winning order parameter becomes $\xi_1 = 1$ and the other order parameters $\xi_2 = \xi_0 = 0$. Thus, the prototype pattern that corresponds to ξ_1 will be recognized. Note that due to the fact that the biggest initial order parameter $\xi_k(0)$ will always win the competition (in the case of balanced λ_k), the iteration (7.27) may be omitted and the resultant system remarkably simplified. The learning process will then be restricted to satisfying (7.19) and solving (7.22). If, however, we are dealing with normalized vectors, finding of adjoint vectors means just transposing and the whole dynamics is reduced to forming the inner products $(v_k^T q_0)$, which further reduces the complexity of numerical computations.

7.2.2.1 Differences from standard model

The model differs from the classical Haken representations in the following respects: (7.17), (7.18) and (7.20). More specifically, in our Acad model, the number of patterns is allowed to be bigger or equal to the number of features $M \geq N$. Additionally, numerical simulations have shown that the model works well in situations where $M \gg N$.

Condition (7.19) has been omitted in our model, as tests have proved it to be redundant.

The size of the prototype vectors is different in our implementation, thus (7.20) is not satisfied. However, the model still performs well. Here, of course, it is the number of meaningful dimensions that is important. For a system to be solvable, trailing zeros should be added to vectors of different size:

$$\begin{aligned}
v_0 &= (L_{01}, L_{02}, L_{03}, 0, 0) \\
v_1 &= (L_{11}, L_{12}, 0, 0, 0) \\
v_2 &= (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}).
\end{aligned}$$

All these modifications, while simplifying the system and allowing for a greater flexibility, do not degrade the model's performance nor obscure the general properties of the synergetic computer or SNN.

7.2.3 Possible realization in ADS

The implementation of the synergetic computer for the recognition of simple AutoCAD entities is certainly not an aim of its own. Instead, these principles are applied to useful recognition scenarios, e.g., as a component of the ADS system. Although there are plenty of different application possibilities that could be elaborated, we have researched one from the HVAC (Heating Ventilation Air Conditioning) engineering domain. More specifically, the process of data acquisition of a building's outer peripheries (e.g., outer walls) from the AutoCAD environment is being automated. To this end, the Heatloss original software can be used. This software has been developed in the author's earlier research as a framework for software agents testing and implementation on the AutoCAD platform.

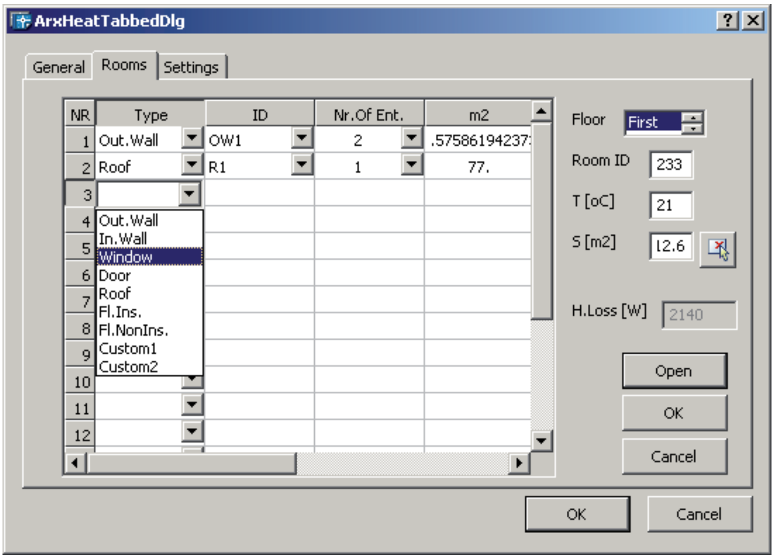
The process of the selection of outer walls of the building clearly falls into the engineer's creative activity class, if we consider this process as a dynamic synergetic system. It is quite easy for the human designer to visually identify the outer walls from other geometry on the graphics screen. For the computer, however, it is not an easy task, if all graphical data is treated equally in the sense of human visual perception. It is our task to treat the underlying vector entities as patterns, so that the principles of the self-organization theory can be applied and the synergetic tools described above be used. Note that the same task may be solved by the "traditional" cybernetic approach methods, e.g., by introducing some additional metadata to the graphical objects in order to enable straightforward computational identification. Such parametrical approach is nowadays very common in information systems design; in the CAD domain it is used, e.g., in BIM (Building Information Modeling) applications. Yet using this approach does not fall into the ADS domain of expertise (for more discussion, see sections 4 and 6), and therefore not the subject of our interest.

In the following subsections, the HeatLoss software's related functionality is briefly explained and its possible improvements by exploiting the properties/advantages of the synergetic computer are described. The issues of optimal permutation selection for test patterns are also discussed.

7.2.3.1 Additional functionality for HeatLoss software

The HeatLoss software is an ObjectARX module that automates the calculation of a building's heat losses. Below is a brief explanation of the related GUI.

The Rooms tab is the main working UI of the program (see Fig. 7.10). It has a grid control, which is similar to a spreadsheet by its functionality. The grid control used in this program is a very powerful custom control. It has a rich set of different features; most of them are currently not used in HeatLoss, but are planned to be included in future releases. In the next version, for example, adding a drag-and-drop capability to the grid is planned.



NR	Type	ID	Nr.Of Ent.	m2
1	Out.Wall	OW1	2	.57586194237
2	Roof	R1	1	77.
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Floor First
Room ID 233
T [oC] 21
S [m2] 12.6
H.Loss [W] 2140

Fig. 7.10 HeatLoss application's room peripheries window (Rooms tab)

The user selects the type of periphery, its ID, and the program immediately directs him to the AutoCAD drawing screen (DWG) where he picks the characteristic points. Next, the program computes the area of the periphery and the heat loss of the room, based on the data in the Settings tab and the room's inner temperature and displays this UI to the user. The process is repeated. It is, of course, possible to modify data in the grid and in the tab after the initial calculation is done. The user may change, for instance, the number of entities, the area of the periphery, the U value, the inner temperature, etc, and the program updates the heat loss value accordingly.

We now add to this program an additional functionality, i.e., the ability for the system to select the outer periphery (outer wall) on its own, using the synergetic computer approach. It has been shown earlier that such an ability transforms the conventional CAD system into an ADS system.

This can be achieved as follows. When the user selects the ID of the type of the periphery chosen, the system, instead of the prompting the user to manually select the periphery's (e.g., the wall's) length, performs the identification process based on the prototype patterns stored. Upon identification, the system acquires technical parameters of the room identified and calculates the heat loss of this room.

In such a fashion, a CAD system is developed that drastically improves the quality of the HVAC process of computer aided heat loss calculation. This is achievable by a high degree automation of the engineering tasks. This means that the engineer needs only to specify the height of the corresponding periphery, all the remaining work is done by the computer. The system then automatically identifies the rooms, peripheries of the room and its geometry, and last, performs the calculation and depicts the results of the calculation on the drawing. Thus, one particular part of engineering design in the HVAC domain is completely automated. It is very important that all this is done based on synergetics principles and the process of identification is compliable with biological properties of visual perception and identification. The input data for this process constitute solely vector graphics and no additional parametric data is used. Thus, the outcome is a HVAC ADS system that is capable of mimicing the performance of a human operator and modeling the non-routine, or in other words, the creative part of the engineering design process.

It must be noted that the software system developed and discussed in more detail in our previous work [47] is already quite intelligent. Although its intelligence characteristics do not fall into the category of modeling the creative parts of the engineering design process, and are therefore not directly classifiable as ADS components, they may constitute the supporting framework for ADS, so using this functionality may further increase the level of autonomy and automation of the resultant CAD system. This refers to the software agents and ANN functionality of the HeatLoss application. Indeed, this was the first attempt to increase the level of automation of a CAD system by providing the background error checking capability and remembering user preferences, i.e., introducing the dynamic learning ability into the process of heat loss calculation. At those times (back in 2006) the term Autonomous Design Systems (ADS) had not yet been introduced, the trend, however, towards the realization of such functionality was clearly seen. A brief overview of this functionality may be useful under the general discussion. In order to minimize the interference of the logic of the current presentation, this review is given in a standalone subsection, see 7.2.4.

One way to compare a room's characteristic vectors is to use all possible combinations (more specifically, permutations) of the distances represented (see Fig. 8.6 in section 8) and to form corresponding test vectors. The basic principle is the same as described in section 7.2.2 or 7.2.4.

In Fig. 8.6 three leaders L1, L2, L3 drawn from the base point BP are shown. This combination corresponds to the prototype pattern vector being recognized. However, in order to include these points (distances) in comparison in AutoCAD, all line/polyline type entities in the vicinity of BP have to be selected. Thus, a number of different distances from BP to the corresponding entities' start/end points is obtained. Next these distances have to be permuted to get the right vector element combination (e.g., L1, L2, L3). For example, in the case of 42 distances (i.e., 42 possible elements of the test vector) and the prototype vector consisting of 3 elements, there are as many as $C_k^r = \frac{k!}{(k-r)!} = \frac{42!}{(42-3)!} = 68,880$ different permutations.

Therefore, in order to preserve numerical computation efficiency, the choice of permutation computation algorithm is very important. In the next subsection this issue is discussed in more detail.

7.2.3.2 Algorithms to generate permutations

As the selection of the permutation algorithm directly influences the performance of the application, a brief description of the permutations generation process/algorithms is given below.

In computing, generating permutations of a given sequence of values may be required. The methods best adapted to do this depend on whether only randomly chosen permutations are needed, or all permutations, and in the latter case if a specific ordering is required. Another question is whether possible equality among entries in the given sequence is to be taken into account; if so, only distinct multiset permutations of the sequence should be generated.

An obvious way to generate permutations of n is to generate values for the Lehmer code (possibly using the factorial number system representation of integers up to n!), and convert those into the corresponding permutations. However, the latter step, while straightforward, is hard to implement efficiently, because it requires n operations each of selection from a sequence and deletion from it, at an arbitrary position; of the obvious representations of the sequence as an array or a linked list, both require (for different reasons) about $n^2/4$ operations to perform the conversion. With n likely to be rather small (especially if generation of all permutations is needed) that is not too difficult, but it turns out that both for random and for systematic generation there are simple alternatives that do considerably better. For this reason, it does not seem useful, although certainly possible, to employ a special data structure that would allow performing the conversion from Lehmer code to permutation in $O(n \log n)$ time.

Random generation of permutations. For generating random permutations of a given sequence of *n* values, it makes no difference whether the intention is to apply a randomly selected permutation of *n* to the sequence, or to choose a random element from the set of distinct (multiset) permutations of the sequence. This is because, even though in case of repeated values there can be many distinct permutations of n that result in the same permuted sequence, the number of such permutations is the same for each possible result. Unlike for systematic generation, which becomes unfeasible for large n due to the growth of the number *n!,* there is no reason to assume that n will be small for random generation.

The basic idea when generating a random permutation is to generate at random one of the *n!* sequences of integers $d_1, d_2, \ldots, d_n$ satisfying $0 \leq d_i < i$ (since d_1 is always zero, it may be omitted), and to convert it to a permutation through a bijective correspondence. For the latter correspondence the (reverse) sequence can be interpreted as a Lehmer code, and this gives a generation method first published in 1938 by Ronald A. Fisher and Frank Yates [88]. While at the time computer implementation was not an issue, this method suffers from the difficulty sketched above to convert from Lehmer code to permutation efficiently. This can be remedied by using a different bijective correspondence: after using *di* to select an element among *i* remaining elements of the sequence (for decreasing values of *i*), rather than removing the element and compacting the sequence by shifting down further elements one place, the element is swapped with the final remaining element. Thus, the elements remaining for selection form a consecutive range at each point in time, even though they may not occur in the same order as they did in the original sequence. The mapping from sequence of integers to permutations is somewhat complicated, but it can be seen to produce each permutation in exactly one way, by an immediate induction. When the selected element happens to be the final remaining element, the swap operation can be omitted. This does not occur sufficiently often to warrant testing for the condition, but the final element must be included among the candidates of the selection, to guarantee that all permutations can be generated.

The resulting algorithm for generating a random permutation of $a[0], a[1], ..., a[n - 1]$ can be described as follows in pseudocode:

```
for $i$ from $n$ downto 2
do   $di \leftarrow$ random element of $\{ 0, ..., i - 1 \}$
swap $a[di]$ and $a[i - 1]$
```

This can be combined with the initialization of the array $a[i] = i$ as follows:

```
for $i$ from 0 to $n-1$
do   $di+1 \leftarrow$ random element of $\{ 0, ..., i \}$
$a[i] \leftarrow a[di+1]$
$a[di+1] \leftarrow i$
```

If $di+1 = i$, the first assignment will copy an uninitialized value, but the second will overwrite it with the correct value *i*.

Generation in lexicographic order. There are many ways to systematically generate all permutations of a given sequence. One classical algorithm, which is both simple and flexible, is based on finding the next permutation in lexicographic ordering, if it exists. It can handle repeated values, for which case it generates the distinct multiset permutations each once. Even for ordinary permutations it is significantly more efficient than generating values for the Lehmer code in lexicographic order (possibly using the factorial number system) and converting those to permutations. To use it, sorting the sequence is started in (weakly) increasing order (which gives its lexicographically minimal permutation), and then this is repeated advancing to the next permutation as long as one is found. The method goes back to Narayana Pandita in the 14th century India, and has been frequently rediscovered ever since. [89]

The following algorithm generates the next permutation lexicographically after a given permutation. It changes the given permutation in-place.

1. Find the largest index k such that $a[k] < a[k + 1]$. If no such index exists, the permutation is the last permutation.
2. Find the largest index l such that $a[k] < a[l]$. Since $k + 1$ is such an index, l is well defined and satisfies $k < l$.
3. Swap $a[k]$ with $a[l]$.
4. Reverse the sequence from $a[k + 1]$ up to and including the final element $a[n]$.

After step 1, it is known that all of the elements strictly after position k form a weakly decreasing sequence, so no permutation of these elements will make it advance in lexicographic order; to advance $a[k]$ must be increased. Step 2 finds the smallest value $a[l]$ to replace $a[k]$ by, and swapping them in step 3 leaves the sequence after position k in weakly decreasing order. Reversing this sequence in step 4 then produces its lexicographically minimal permutation, and the lexicographic successor of the initial state for the whole sequence.

Generation with minimal changes. An alternative to the above algorithm, the Steinhaus–Johnson–Trotter algorithm, generates an ordering on all the permutations of a given sequence with the property that any two consecutive permutations in its output differ by swapping two adjacent values. This ordering on the permutations was known to 17^{th} century English bell ringers, among whom it was known as "plain changes". One advantage of this method is that the small amount of change from one permutation to the next allows the method to be implemented in constant time per permutation. The same can also easily generate the subset of even permutations, again in constant time per permutation, by skipping every other output permutation [89].

7.2.3.3 Pattern's feature selection optimization

As computational speed is quite important in engineering applications [90], we do not want to degrade software performance in the permutations module either. This is particularly important for CAD and ADS systems. That is why ObjectARX (C++) technology has been chosen for our systems' implementation. In comparison with other AutoCAD development technologies (VBA, .NET, AutoLisp/DCL see, e.g., [91]), ObjectARX is the most powerful IDE creating the fastest and most compact ARX (DLL) modules available. The processing speed of the permutation algorithm depends on the functions and programming language constructs chosen for one particular application. A review of some widely used C++ permutation algorithms/functions is presented here. We then compare them and select the most effective (optimal) routine for our system implementation, i.e., we perform the pattern selection process optimization for our ADS module.Howard Hinnant [92] has performed the tests amongst most widely used permutation (combination) algorithms. Below is a brief review of the results. There are 3 different approaches, solutions A, B, and C.

Solution A

The standard library has *std::next_permutation* and it is possible to trivially build a *next_k_permutation* from it and a *next_combination* from that (Fig. 7.11):

```
template<class RandIt, class Compare>
bool next_k_permutation(RandIt first, RandIt mid, RandIt last,
Compare comp)
{
    std::sort(mid, last, std::tr1::bind(comp,
std::tr1::placeholders::_2
                                             ,
std::tr1::placeholders::_1));
    return std::next_permutation(first, last, comp);
}
```

Fig. 7.11 Example code for solution A

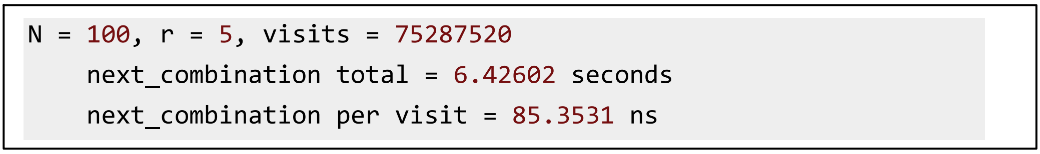
The performance results of this solution are as follows (Fig. 7.12):

```
N = 100, r = 5, visits = 75287520
    next_combination total = 4519.84 seconds
    next_combination per visit = 60034.3 ns
```

Fig. 7.12 Performance printout of solution A

Solution B

This solution was developed by Hervé Brönnimann (called N2639) and can be found in [93]. This proposal adds eight algorithms (*std::next_partial_permutation, next_combination, next_mapping, next_repeat_combination_counts, their counterparts std::prev_partial_permutation, std::prev_combination, std::prev_-mapping, std::prev_repeat_combination_counts,* with their overloads) to the header <algorithm>, for enumerating permutations and combinations, with and without repetitions. They mirror and extend *std::next_permutation* and *std::prev_permutation.* For sizes known at compile time, these algorithms can generally be simulated by a number of nested loops. The performance results of this solution are shown in Fig. 7.13.

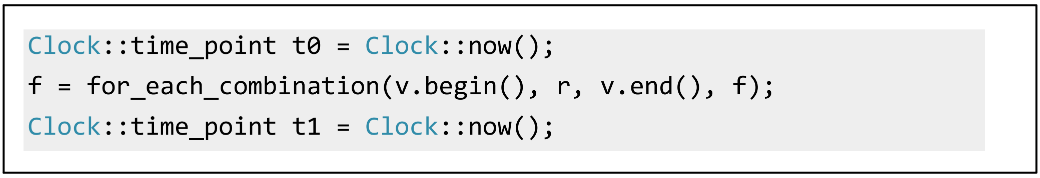


```
N = 100, r = 5, visits = 75287520
    next_combination total = 6.42602 seconds
    next_combination per visit = 85.3531 ns
```

Fig. 7.13 Performance printout of solution B

Solution C

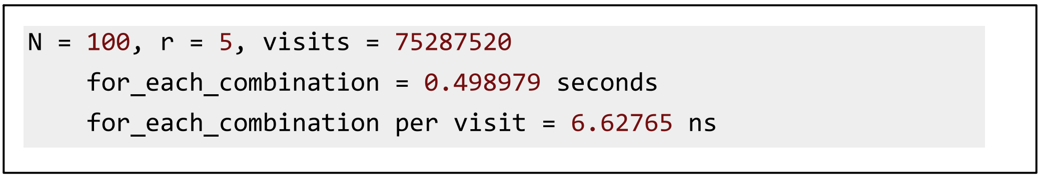
Finally, there is solution C (see [94]). This solution has a different signature/style and is called *for_each_combination (for_each_permutation),* and is used much like *std::for_each*. The driver code between the timer calls is as follows (Fig. 7.14):



```
Clock::time_point t0 = Clock::now();
f = for_each_combination(v.begin(), r, v.end(), f);
Clock::time_point t1 = Clock::now();
```

Fig. 7.14 Driver code between the timer calls of solution C

The performance results of this solution are shown in Fig. 7.15.



```
N = 100, r = 5, visits = 75287520
    for_each_combination = 0.498979 seconds
    for_each_combination per visit = 6.62765 ns
```

Fig. 7.15 Performance printout of solution C

Solution C is 12.9 times faster than solution B, and over 9000 times faster than solution A.

We consider this a relatively small problem in case of only 75 million visits. As the number of visits increases into billions, the discrepancy in the performance between these algorithms continues to grow. Solution A is already unwieldy. Solution B eventually becomes unwieldy. Solution C is the highest performing algorithm to visit all combinations/permutations the author is aware of.

Thus we have to choose the approach of *solution C* for our systems' coding.

7.2.4 HeatLoss software, its existing functionality of software agents and ANN

In this section, a brief overview of the software system developed in the author’s previous work is presented. The rationale of this is to remind the reader of one possible underlying framework for ADS implementation and the fact that the search for ADS characteristics actually began already back in the year 2006. In addition, this part of the paper describes the mechanisms of the HVAC CAD heat loss calculation.

According to the objectives stated in that research, a software application based on agents technology was developed. Particularly, Jade 2.5 platform was used for our agent system and the AutoCAD platform for getting and setting input/output data.

From the HVAC discipline the process of heat loss calculation of a building (room) was selected as a case for automation and agents technology testing and implementation.

As an output, a software program based on EVS 829:2003 was developed. This program was tested and intensively used in real-life design scenarios during the period of 2006 – 2010, and proved itself as a versatile and intuitive tool that could help a HVAC engineer in his daily work. It could be used with or without the agents suite, which was programmatically a standalone module and could be used independently from the heat loss calculation routine.

7.2.4.1 Architecture

Basically, the practical implementation consists of two modules: the ObjectARX program, named HeatLoss, and agents infrastructure (HeatLossAgents), which is an independent process, written entirely in C#. Thus it runs under .NET framework.

The HeatLoss program is designed to run under the AutoCAD platform. It is written using ObjectARX technology/libraries (unmanaged C++).

These two modules, although they may be run independently from each other, are designed to work together (see Figure 7.16).

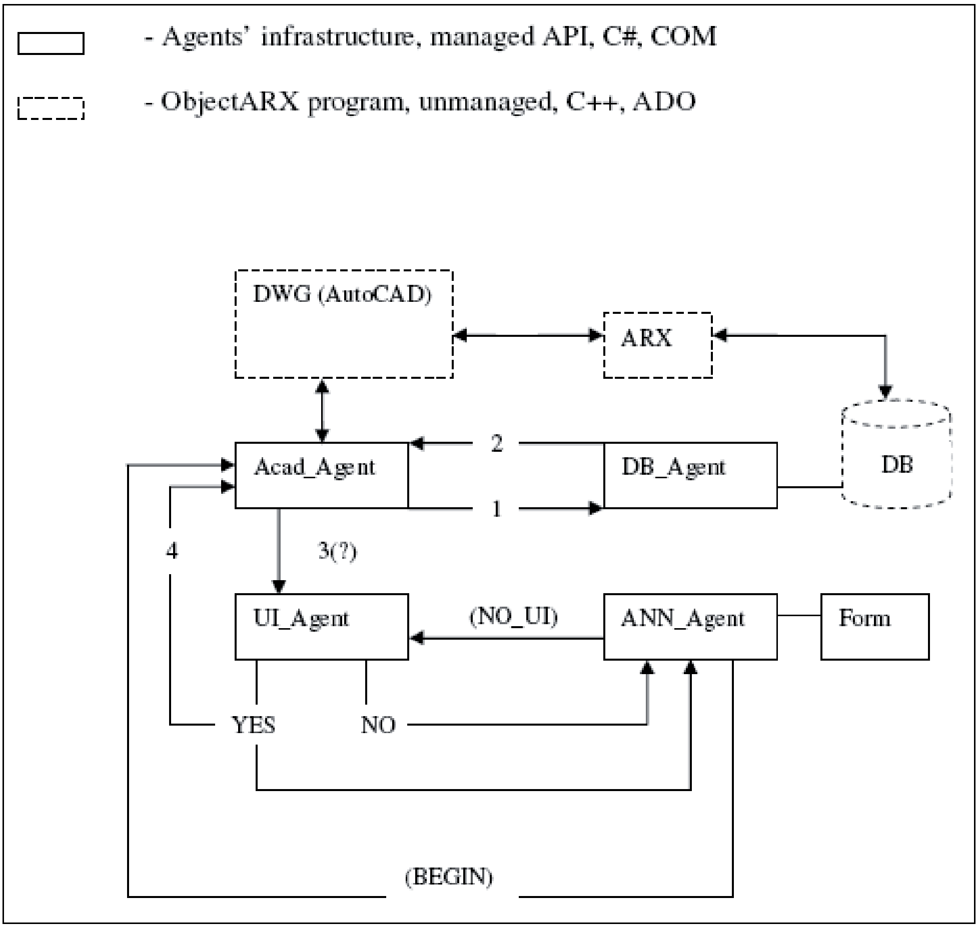
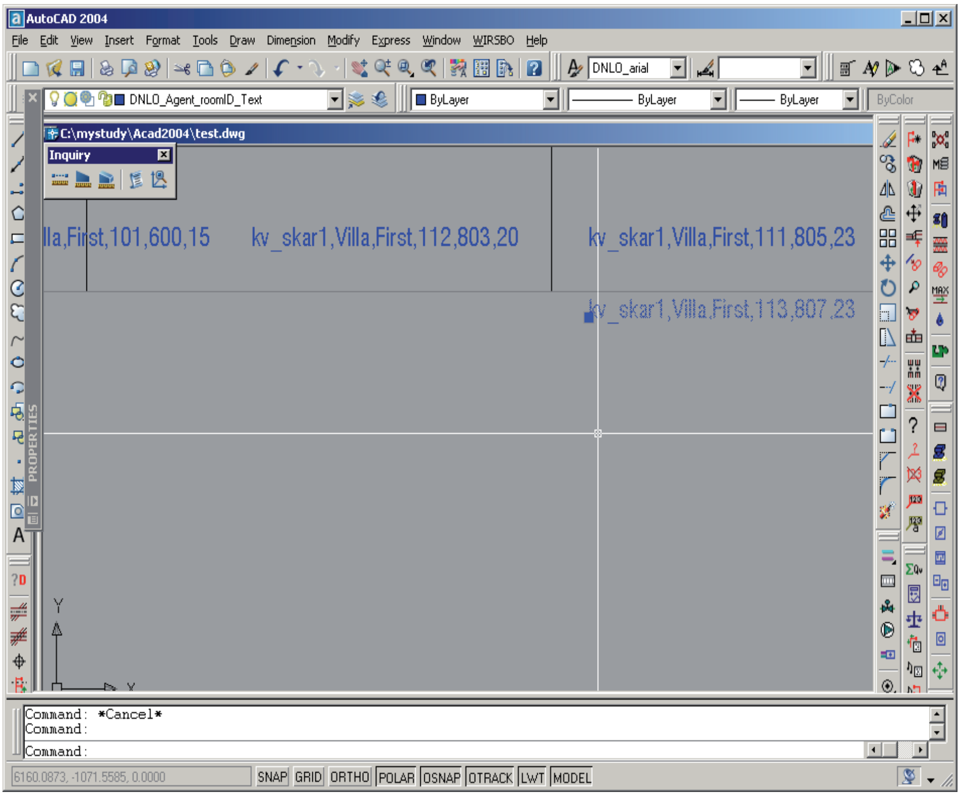


Fig. 7.16 Program architecture (ARX + Agents suite)

The main idea behind these two software applications is to facilitate routine heat loss calculation done by a human designer. The HeatLoss application calculates heat losses and writes the appropriate data to the program database. HeatLossAgents, while working in parallel, compares this data with that already in the program database (DB), and suggests possible changes (if it detects errors) to the designer in real time.

HeatLoss and HeatLossAgents communicate with each other indirectly through the AutoCAD database (and graphical editor) and program database, which is marked as DB in Figure 7.16

Program database (DB, HeatLossData.mdb) is an MS Access relational database and consists of four (4) tables. These tables are Projects, RoomBoundaries, Rooms, and Settings. Currently two of them are used – Rooms and Settings (the remaining tables along with additional program features are intended to be included in the next program releases, the current one is v2.1). The Settings table data are used entirely by HeatLoss, from here it reads/writes program settings data. This data includes dimensions and values of U [W/m2*K] coefficients of the building peripheries, such as doors, walls, windows, etc, the Project ID value, outdoor temperature, and type of the building (e.g., villa, industrial, residential etc). The Rooms table is used both by HeatLoss and HeatLossAgents. This table holds data about an individual room's properties and consists of the following fields: ID, Room ID, Project ID, Building type, Area of the room [m2], Heat loss of the room [W], and Floor Type of the room (may be First, Last or Intermediate).



```
lla,First,101,600,15
kv_skar1,Villa,First,112,803,20
kv_skar1,Villa,First,111,805,23
kv_skar1,Villa,First,113,807,23
```

```
Command:  *Cancel*
Command:

Command:
```

```
6160.0873, -1071.5585, 0.0000
```

Fig. 7.17 AutoCAD drawing window with text entities created by HeatLoss

HeatLoss application writes (adds) this data into DB after a given room is processed. HeatLossAgents, by contrast, reads/writes data from DB, operates on it and computes some new outputs from it.

Initially, HeatLoss creates a record in the AutoCAD drawing database (DWG) and adds a text entity to the drawing screen (Fig. 7.17). After that it adds a record with appropriate data to DB.

HeatLossAgents operates both on these text entities, extracting the data needed, and DB data, calculating and comparing it with the data extracted from DWG. That is how agents get the data flow from the HeatLoss ObjectARX application.

The module intercommunication is done through databases – one through DB and another through the AutoCAD DWG. No direct communication is accomplished between the two program modules. This means one program saves data to a database and the other reads this data.

7.2.4.2 HeatLoss (ARX)

HeatLoss is an ObjectARX application that runs under AutoCAD. The main objective of that application module is to calculate heat losses of the building or room and supply output data (AutoCAD text object) to HeatLossAgents.

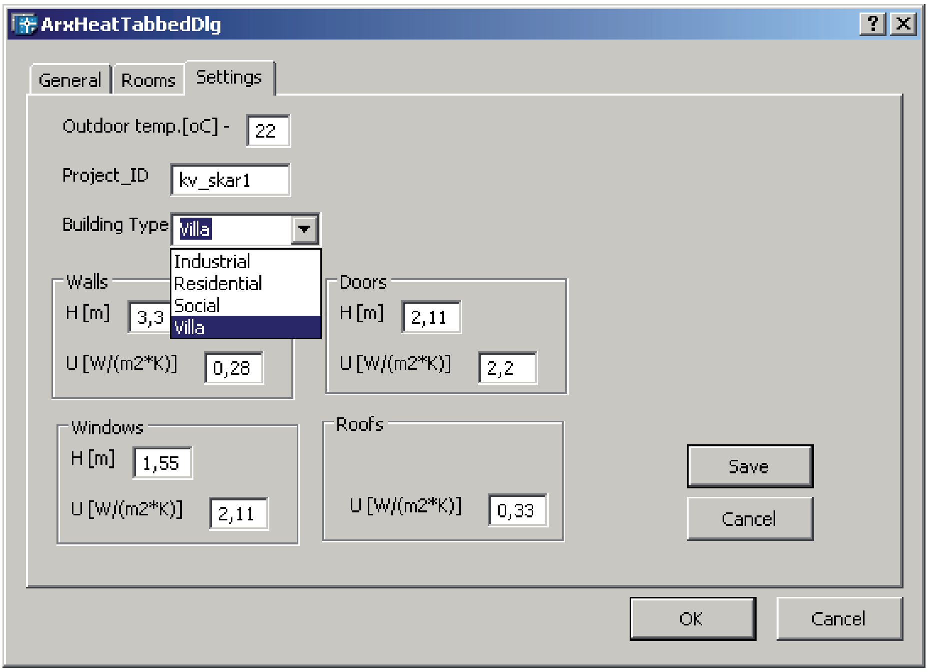
The program calculates heat losses based on the room's geometry and user inputs, both of them supplied through the AutoCAD drawing screen. Unlike the agents suite, HeatLoss is a regular computer program that has no built-in intelligence or communication capabilities. It may be used without the accompanying agents suite. In that case it becomes a CAD helper application that automates the engineer's daily tasks without the use of agents technology.

Nevertheless, it was created as a powerful, intuitive and laconic application. At the time of its creation we had no information about analogous software on the market.

The main rationale behind the program is that the user selects dimensions of the room's peripheries in the drawing. It is similar to the process of picking points in AutoCAD. There is no need to type the dimensions of the peripheries separately. Such dimension acquiring mechanism was selected intentionally in order to guarantee program compatibility with virtually any architectural input data (it does not depend on the layer's name, type of the object representing the periphery, etc).

In the current version (v2.1), the program has two functional tabs: Rooms and Settings. The Rooms tab is described in section 7.2.3.1 (see also Fig. 7.10).

When all peripheries of the room are processed, heat loss data needs to be added to DB and DWG. Pushing the "OK" button does this. The format of the record is as follows: Project ID, Type of the building, Type of the floor, Room NR, Heat Loss, and Area of the room. All this information is needed for HeatLossAgents to operate properly. The Settings tab is needed to restore the program's default values (Fig. 7.18), i.e., those that do not change frequently. It holds data for the periphery's height, U coefficient and project ID, type of the building, and outdoor temperature. The "Save" button saves this data to DB.



Outdoor temp.[oC] - 22
Project_ID kv_skar1
Building Type Villa

Walls
H [m] 3,3
U [W/(m2*K)] 0,28

Doors
H [m] 2,11
U [W/(m2*K)] 2,2

Windows
H [m] 1,55
U [W/(m2*K)] 2,11

Roofs
U [W/(m2*K)] 0,33

Fig. 7.18 HeatLoss Settings tab

Special attention is paid to the simplicity of using the program, e.g., to compute the heat loss of a non-insulated floor (calculated by zones according to EVS 829:2003), the user must only select the room's outer/inner points and all the geometry calculation is done by the program. Such approach to application functionality took a lot of time and resources. The function alone that computes the areas of zones of non-insulated floor, for example, includes about 500 lines of code.

Tools and Technologies Used

The following tools and technologies were used for the application module:
ADO (COM), C++, ObjectARX, MFC, Microsoft Visual C++ v7.0, CGridCtrl v2.25

Facts

Physical memory usage (without AutoCAD) - ~3000K
Size of the file (Release mode) – 420KB
Total files used (Debug, Release) – 110
Total number of lines of code (only source files, without Grid control, without header files) – 4500
Runs under AutoCAD 2004 or higher

7.2.4.3 HeatLossAgents

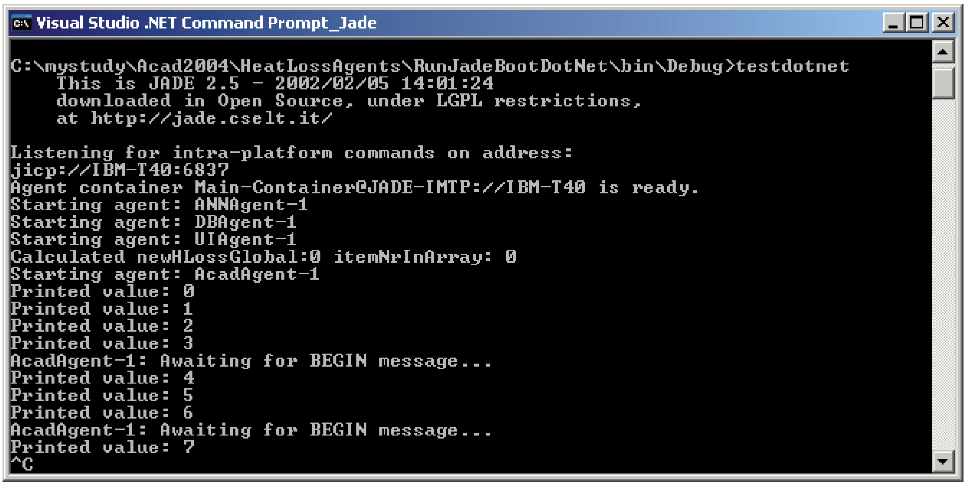
Developing the HeatLossAgents agents suite was one of the main contributions of that work [47]. The goal was to test and analyze the use of software agents in the HVAC design. HeatLossAgents addresses this goal by aiding the human designer in routine heat loss calculation.

The application was developed using JADE 2.5 framework's LEAP version ported to .NET platform. Agents were written mainly in C#.

The agent system is programmed to check correctness of the calculation of heat losses.

The main principle behind this is that agents constantly monitor outputs of HeatLoss and compare this with the calculated data based on DB records. Specifically, the agent extracts from DB the value of the specific heat loss (W/m2) that corresponds to the given building type and floor type, and compares this data with the respective data from DWG (e.g., the drawing currently processed using HeatLoss). When it finds a difference that is greater than some threshold value (e.g., 10%), it informs the user. When the user chooses to correct the value, agents update the data both in DWG and DB.

The agent system consists of four autonomous agents: Acad_Agent, DB_Agent, UI_Agent and ANN_Agent. Each agent has its own task and is capable of communicating with other agents (see Figure 7.16).



```
C:\mystudy\Acad2004\HeatLossAgents\RunJadeBootDotNet\bin\Debug>testdotnet
    This is JADE 2.5 - 2002/02/05 14:01:24
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.cselt.it/

Listening for intra-platform commands on address:
jicp://IBM-T40:6837
Agent container Main-Container@JADE-IMTP://IBM-T40 is ready.
Starting agent: ANNAgent-1
Starting agent: DBAgent-1
Starting agent: UIAgent-1
Calculated newHLossGlobal:0 itemNrInArray: 0
Starting agent: AcadAgent-1
Printed value: 0
Printed value: 1
Printed value: 2
Printed value: 3
AcadAgent-1: Awaiting for BEGIN message...
Printed value: 4
Printed value: 5
Printed value: 6
AcadAgent-1: Awaiting for BEGIN message...
Printed value: 7
^C
```

Fig. 7.19 Agents system awaiting BEGIN message

Acad_Agent

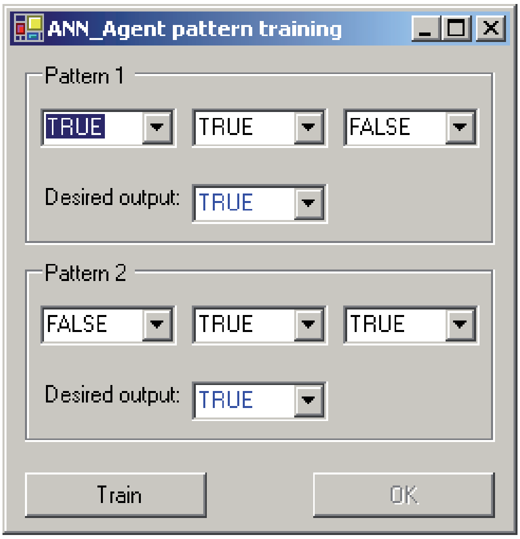
When the system is started, Acad_Agent is responsible for starting the AutoCAD session. It opens a drawing and waits for a BEGIN message from ANN_Agent (see Figure 7.19). When the BEGIN message arrives, Acad_Agent starts monitoring DWG. It processes each heat loss record in DWG, and sends this data to DB_Agent, which replies with specific heat loss value. Acad_Agent then compares thess data, and if any difference is found, informs UI_Agent about it. UI_Agent consults with the user and if he/she decides to (not) correct the error, UI_Agent inform Acad_Agent about the decision. Acad_Agent and DB_Agent then decide? whether to update the data or not. Acad_Agent is responsible for AutoCAD operations, data comparison and communication with UI_Agent and DB_Agent.

DB_Agent

DB_Agent is responsible for program database operations. It can read/write the specific data from DB and calculate different characteristics. DB_Agent communicates only with Acad_Agent.

UI_Agent

UI_Agent's mission is to be a buffer between the user and the other agents. It can communicate with the human designer and translate his/her commands back to peer agents. It has the user interface to accomplish this. It communicates with the user, Acad_Agent and ANN_Agent.



Pattern 1

TRUE TRUE FALSE

Desired output: TRUE

Pattern 2

FALSE TRUE TRUE

Desired output: TRUE

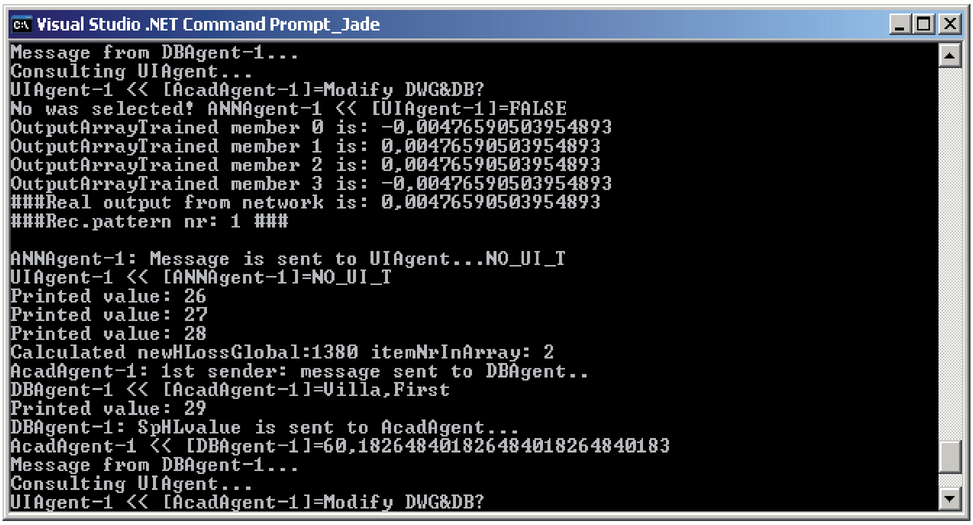
Fig. 7.20 ANN_Agent's user interface form

ANN_Agent

This agent was introduced to add even more intelligence into the agent system. ANN_Agent incorporates the Artificial Neural Network (ANN) functionality. In our case, the ANN can be trained to recognize four (4) different patterns with four different outputs. In this application, patterns are answers that the user gives to UI_Agent. The answer is either "Yes" or "No". "Yes" means that the error found is to be corrected, "No" – that it is not to be corrected. When ANN_Agent starts, it displays the user form where the user can train different patterns and choose some arbitrary outputs (see Figure 7.20). When the network has been trained, ANN_Agent sends a BEGIN message to Acad_Agent, and the agent system starts functioning. When the system finds errors, UI_Agent asks the user about the desired action, ANN_Agent receives pattern elements from UI_Agent and tries to recognize them. If it recognizes the pattern, it sends a NO_UI message to UI_Agent. After that UI_Agent does not ask the user anymore, but behaves according to the desired output value of the recognized pattern.

Such a mechanism models intelligent prediction services, which further simplify the work of a human designer. Note that patterns and their outputs are adjusted during program runtime, i.e., there is no programmed behavior, and no logic exists between pattern and its output.

The reason for training only two patterns, while the system is capable of recognizing four, are the selected ANN characteristics, which allow to recognize inverse patterns in addition to the trained ones. This means that if we trained the network for a TRUE, TRUE, FALSE pattern with TRUE output, the system automatically recognizes the FALSE, FALSE, TRUE pattern, and assigns to it the output FALSE. Agents' communication output is shown in Fig. 7.21.



```
Message from DBAgent-1...
Consulting UIAgent...
UIAgent-1 << [AcadAgent-1]=Modify DWG&DB?
No was selected! ANNAgent-1 << [UIAgent-1]=FALSE
OutputArrayTrained member 0 is: -0,00476590503954893
OutputArrayTrained member 1 is: 0,00476590503954893
OutputArrayTrained member 2 is: 0,00476590503954893
OutputArrayTrained member 3 is: -0,00476590503954893
###Real output from network is: 0,00476590503954893
###Rec.pattern nr: 1 ###

ANNAgent-1: Message is sent to UIAgent...NO_UI_T
UIAgent-1 << [ANNAgent-1]=NO_UI_T
Printed value: 26
Printed value: 27
Printed value: 28
Calculated newHLossGlobal:1380 itemNrInArray: 2
AcadAgent-1: 1st sender: message sent to DBAgent..
DBAgent-1 << [AcadAgent-1]=Villa,First
Printed value: 29
DBAgent-1: SpHLvalue is sent to AcadAgent...
AcadAgent-1 << [DBAgent-1]=60,1826484018264840182648401 83
Message from DBAgent-1...
Consulting UIAgent...
UIAgent-1 << [AcadAgent-1]=Modify DWG&DB?
```

Fig. 7.21 Agents' communication log

Tools and Technologies Used

The following tools and technologies were used for that application module: JADE 2.5, LEAP, ADO.NET, COM (ActiveX), C#, J#, managed C++, Microsoft Visual C# .NET v7.0, NeuroBox 2.5, Java

Facts

Physical memory usage (without AutoCAD) - ~27700K
Size of the file (EXE) – 20KB
Sizes of other files (DLL, different agents) – 12KB, 8KB, 6KB

Libraries (also loaded in memory):
AutoCAD.dll – 684KB
JadeLeapDotNet.dll – 652KB
NeuroBox.dll – 36KB
Total files used (Debug) – 131
Total number of lines of code (only agents' source files):
Acad_Agent – 595
DB_Agent – 233
ANN_Agent – 170
UI_Agent - 144

7.2.5 Synergetic computer algorithm on AutoCAD as part of HVAC ADS

In this section, the results of the implementation of the synergetic computer core functionality algorithm and its optimization for the vector graphics CAD environment and one particular HVAC problem are discussed. This algorithm is based on the results obtained during the adaptation of the standard model to the AutoCAD environment discussed in the preceding subsections, as well as on additional modifications that were done while implementing the model as part of ADS.

In order to extend the functionality of the synergetic computer on the AutoCAD platform, a number of modifications to the standard Haken model were committed. These are considered in more detail in the following subsections. This optimization together with the system implementation may be regarded as an important contribution to the classical synergetic computer algorithm. Again, all these modifications, while simplifying the system and allowing for greater flexibility, do not degrade the model's performance or obscure the general properties of the synergetic computer. It is worth noting that the recognition rate of our model has so far been 100%. This could be explained by the fact that only noiseless patterns have been used as test vectors.

The implementation of SNN in ADS, as well as the treatment of noisy patterns, is a subject for further research.

7.2.5.1 Quick Haken Algorithm

Normally, in the classical model, order parameters obey the initial condition $\xi_k(0) = (v_k^+ q(0))$ and change according to the dynamic equation of order parameters (for the explanations of the equation's parameters, here and forth, refer e.g., to sections 5 or 7.2.1):

$$\dot{\xi}_k = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C \sum_{k'=1}^{M} \xi_{k'}^2 \xi_k .$$

The corresponding evolution, e.g., for the case of three order parameters is shown in Fig. 7.9.

The system converges after twelve steps, i.e., the winning order parameter becomes $\xi_1 = 1$ and others $\xi_2 = \xi_0 = 0$. Thus, the prototype pattern that corresponds to ξ_1 will be recognized. It is possible, however, to prove that the biggest initial order parameter $\xi_k(0)$ will always win the competition (in case of balanced λ_k), and this iteration may be omitted and the resultant system remarkably simplified. This proof shall not be shown here, but rather the final result of the work is presented, tested and implemented in digital models, first in MATLAB, and then on the AutoCAD platform.

Such simplified computational process is called the *Quick Haken Algorithm*, since the overall learning time and the complexity of numerical computations is significantly reduced. This enhanced algorithm is used in our software applications.

7.2.5.2 Cosine similarity OP generation

The standard model of synergetic computer assumes that the order parameters (OP) are found as a dot product (DP) between prototype and test pattern vectors:

$$\xi_k = (v_k^+ q).$$

This approach is justified in pattern recognition scenarios where the process variables satisfy the conditions of the classical synergetic computer model (see section 5). The experiment showed that DP based order parameter generation does not work in our system and the winning prototype patterns are *identified erroneously*. The reason for this is that DP value depends on the order of the vector elements and, since this order varies in the permutation series, it is virtually possible that maximum OP is achievable at different element combinations from those equal to the prototype vectors. In order to correct the behavior of the system, we tried to introduce additional competition among the selected maximum OP value patterns, but this approach, with the number of patterns increasing, eventually led to an unwieldy system. Thus, a new OP generation principle was introduced instead.

We have found that OP generation based on the *cosine similarity* (CS, also known as a correlation coefficient) measure works well in our situation, and it is sufficient to arrange a single competition set to determine the correct winning pattern. CS OP generation equation, given two vectors A, B, is as simple as:

$$\xi_k = \cos\theta = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}}. \tag{7.28}$$

If compared vectors are identical the CS OP value obtains its maximum, and the resulting similarity ranges from −1, meaning exactly opposite, to 1, meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

7.2.5.3 Test/prototype vectors' dimensions

The classical Haken representations state that the number of features per pattern should be the same for all prototype and test vectors (equality of vectors' meaningful dimensions). That is, for each

$$\begin{aligned} &v_1(k), v_2(l) \ldots v_n(m); \\ &k = l = \ldots = m, \end{aligned} \tag{7.29}$$

where *k, l,…,m* are vectors' meaningful dimensions.

The size of the prototype vectors is different in our implementation, thus (7.29) is not satisfied. However, the model still performs well. Here, of course, it is the number of meaningful dimensions that is important. For a system to be solvable, trailing zeros should be added to the vectors of different size:

$$\begin{aligned} &v_0 = (L_{01}, L_{02}, L_{03}, 0, 0) \\ &v_1 = (L_{11}, L_{12}, 0, 0, 0) \\ &v_2 = (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}). \end{aligned} \tag{7.30}$$

7.2.5.4 The number of patterns and features

In our CAD model the number of patterns is allowed to be bigger than or equal to the number of features $M \geq N$. Additionally, numerical simulations have shown that the model works well in a situation where $M \gg N$. The standard model assumes that $M \leq N$.

7.2.5.5 Vectors zero-mean and normalization conditions

In the standard model, vectors v_k are subject to the condition $\sum_k v_k = 0$. Our experiments showed that the model performs well even without satisfying this condition, so this was omitted in our implementation.

In the standard model the following normalization holds:

$$(v_k^T v_k) \equiv \sum_{j=1}^{N} v_{kj}^2 = 1.$$

The system was tested both with and without this normalization and it was found that the OP generation based on normalized vectors is 10 ms shorter than non-normalized OP generation. However, the normalization operation alone takes 70 ms, so eventually the non-normalized operations become faster by 60 ms. Obviously, this option was chosen as a more efficient computational method.

7.2.6 Contribution to the HVAC domain

As the HVAC related discussion and the importance of the research related to the field of heating and ventilation is distributed across this rather capacious paper, a concise summary of the benefits of the particular system in the HVAC domain is provided below. This discussion is intended to give an insight into the HVAC related results of this research and will hopefully help to grasp the meaning of the importance of the current work for this field.

The possible contributions of this research to the field of heating and ventilation engineering are discussed in this section.

7.2.6.1 Automation of routine engineering tasks/work

The most notable contribution lies in simplifying the design engineer's routine activities by providing a way of automating design tasks.

The software system discussed in this paper can be applied in the HVAC design scenarios on a daily basis and can considerably improve the quality and performance of buildings' heat loss calculations. Currently, the most spread real-life practice for calculating heat losses is still the usage of MS Excel type spreadsheets and manual typing of the building's input data into the software UI. Although such a practice is faster than completely manual calculations, it still requires a lot of time and effort to successfully complete the task. In addition, human errors are very likely to arise during such an activity. In recent years, with the introduction of the BIM paradigm and other parametric design tools, the option of calculating the heat losses of rooms within the digital model of the whole building has appeared. This, of course, is a good perspective and possibility, but it has the major drawback of requiring the composition of a complete 3D model of the whole building, while the majority of the engineers in this field do not use BIM technology at all. In contrast, the ADS system proposed in our research is of universal nature (as are synergetics and the self-organization paradigm). Its general properties and functions are equally applicable in plain 2D drawings, in the 3D world, and in BIM models. This is actually as it should be, as the true ADS system models the creative activity of the engineer/designer, and for a creative human being there is no difference how the geometry of the building/room is visually recognized, whether in a 3D model or on paper.

The system proposed in this thesis is able to automatically perform the calculation of heat loss of a given building. The only input required from the operator (in the case of a 2D drawing) is the height of the corresponding periphery, i.e., the outer wall, window, door, etc. The system autonomously processes the geometry of the drawing, identifying and selecting the appropriate entities on the graphics screen, and based on the data collected, performs the calculation of heat loss. It is important to add that this system is able to significantly reduce the time of overall operation and the number of possible errors during the calculation process. Combining the features of ADS structures with other useful properties of cybernetic models (see, e.g., section 7.2.4) allows improving and optimizing the performance of the underlying systems even further. This ultimately results in conserving resources and improving efficiency.

The actually implemented part of the system described in the current work is the part responsible for the identification and selection of the geometry of the building drawing/plan (see section 8). However, it is shown that implementing the rest of the discussed functionality is a rather simple mechanical effort; therefore it was not realized in this AutoCAD implementation. Instead, theoretical argumentation and the corresponding reasoning and discussion of issues related to the modeling of engineering creativity have been focused on. Thus, the required theoretical and practical basis has been created, assuring successful future implementations of ADS in HVAC.

It must be noted that the core functionality of the ADS system discussed, being of universal nature, is also universally applicable and not only in the HVAC field but in other branches of engineering and in other engineering design processes. As the system is based on synergetics, the limits of its application possibilities are limited only by the ingenuity of man.

7.2.6.2 Modeling of the creative part of the HVAC design

This part is related to the discussion in the previous subsection, however the focus is laid on modeling the creative part of the engineering design, thus providing even higher level of automation.

The general discussion, which is equally applicable to the HVAC domain, is presented in sections 4 and 6. In this subsection, issues specific to the heating and ventilation field of expertise, in particular to the process of heat loss calculation, are pinpointed.

Identification of the outer peripheries of the building in the process of heat loss calculation, based *solely on visual perception,* belongs to the creative part of engineering design. This is due to the fact that this type of activity (in such a universal manner) is very hard to model using the conventional cybernetic approach. The common practice for visual identification of outer walls, windows, etc is performed by (the eyes of) the operator. The case of parametric “identification”, common in BIM models/software, is not considered here as this does not belong to creative activities. Then, the information acquired is transmitted to the calculation module of the CAD application or, in the simplest case, used for manual calculations. The approach provided in this thesis helps to automate the *acquisition* process *of graphical information* from the computer-represented data. These data are then used in the corresponding calculation modules of the ADS systems discussed in the previous subsection. This is the part of the implementation that is called modeling the creative component of engineering design. Similar to the discussion mentioned above, this helps to facilitate engineering design, reduces design time errors, optimizes overall process efficiency and, consequently, preserves resources. Again, this ADS module is universally applicable and may be used in other engineering tasks. The case of outer peripheries identification is just one particular example among a wide range of possible applications in the HVAC design. Other possible examples are, e.g. automatic identification of mechanical parts of the HVAC equipment, collision identification of ventilation ducts, and HVAC and plumbing pipes.

8 IMPLEMENTATION

In this section, the results of the prototype testing in the form of software applications are presented. First, an overview of different software technologies and libraries used in this implementation is given. As the AutoCAD platform has been chosen as the base framework for our models, a short overview of its available development technologies is presented as well. Then a more in-depth discussion of the software applications built in the framework of this project, the explanation of their functionality, user interfaces, etc is presented.

8.1 Tools and technologies used

The following software development and prototyping tools and technologies have been used: AutoCAD Architecture 2008 as the main framework for the model, VC++ 8.0, ObjectARX 2007, Eigen 3.1.0, MATLAB 7.0, and Boost 1.49.0.

Although newer versions of AutoCAD exist now, the older one was chosen intentionally in order to demonstrate that our contemporary approach also works on older, more basic platforms. If needed, building an application that runs on AutoCAD 2012/2013 is as simple as just recompiling the project with VC++ 10.0 and ObjectARX 2013 settings.

In order to articulate the algorithms' performance differences, processor clock times have been measured in the debug version of the application. Naturally, release configuration builds are much faster and more compact than debug configurations, and for real life programs the former should be used. Moreover, to get an idea of the computational overhead of our systems, they are run in a quite moderate computational environment: processor - x86 Family 6 Model 9 Stepping 5 GenuineIntel ~1495 Mhz, and Total Physical Memory - 512 MB.

8.2 Developing under AutoCAD

AutoCAD software provides a flexible development platform for specialized design and drafting applications. Its open architecture enables customizing AutoCAD to suit the designer's unique purposes. Examples include Autodesk's industry-specific design software and thousands of add-on applications from members of the Autodesk Developer Network.

8.2.1 Programming tools

The following is a short overview of different programming technologies that the AutoCAD platform currently provides. This overview also attempts to answer the question why the author of this work has selected *ObjectARX technology for interaction with AutoCAD*.

8.2.1.1 ObjectARX

The ObjectARX programming environment provides an object-oriented C++ application programming interface (API) for developers to use, customize, and extend AutoCAD software and AutoCAD-based products like AutoCAD Architecture, AutoCAD Mechanical, and AutoCAD Civil 3D software (currently, however, ObjectARX may refer also to the development with C#, and VB .NET API's).

ObjectARX libraries provide a versatile set of tools for application developers to take advantage of the open architecture of AutoCAD software and provide direct access to the AutoCAD database structures, graphics system, and native command definition. In addition, these libraries are designed to work in conjunction with Visual LISP, ActiveX Automation, and COM so that developers can choose the programming tools best suited to their needs and experience.

ObjectARX technology helps to develop fast, efficient, and compact applications. It enables power users to customize AutoCAD software and frees CAD designers from repetitive tasks. Smaller files, faster drawing operations, and smooth interoperability make an application built with ObjectARX the best choice for a design software solution. That is the official Autodesk's position on ObjectARX technology. It is clear that this programming technology has remained the most powerful and complete solution for AutoCAD platform customization until now.

Developers can use ObjectARX to accomplish the following tasks:

- Access the AutoCAD database
- Interact with the AutoCAD editor
- Create user interfaces using the Microsoft Foundation Classes (MFC) or Win32 native support
- Support the multiple document interface (MDI)
- Create custom classes
- Build complex applications
- Interact with other programming environments

An ObjectARX application is a dynamic link library (DLL) that shares the address space of AutoCAD and makes direct function calls to AutoCAD. Developers can add new classes to the ObjectARX program environment and export them for use by other programs. It is also possible to extend the ObjectARX protocol by adding functions to existing AutoCAD classes at run time.

The ObjectARX SDK is published by Autodesk and freely available under license from Autodesk

8.2.1.2 .NET

.NET is a quite new AutoCAD API, compared to, e.g., ObjectARX or VisualLisp. Starting from AutoCAD 2005, it has been possible to use it among other customization tools to develop AutoCAD applications. .NET has been developed for customizing and extending AutoCAD and AutoCAD-based products with direct access to AutoCAD database structures, native command definition, and more using any .NET supporting language (VB.NET, C#, J# etc). The main advantage is getting the power of ObjectARX® with the ease of learning and use of Microsoft® Visual Basic® (VB (e.g., WinForms). The disadvantages are the still limited functionality (if compared to ARX), and execution speed of program modules being lower than that of ARX analogies (e.g., the loading time overhead of the Layers properties dialog box in AutoCAD 2005, which was written on .NET).

The fact that still roughly about 70% of AutoCAD software itself is developed using ObjectARX (C++) technology speaks in favour of ObjectARX. This holds true for the latest releases, regarding the earlier ones, e.g., AutoCAD 2005 was 99% unmanaged C++ and AutoCAD 2004 and earlier were 100% pure unmanaged C++.

8.2.1.3 ActiveX (COM Automation)

Using the ActiveX (COM Automation) interface in AutoCAD software, applications with a variety of programming technologies can be built, including Microsoft Visual C++, Microsoft Visual Basic for Applications (VBA), Microsoft Visual Basic (VB), Delphi™, and Java™.

Generally, it is possible to use any language if AutoCAD type libraries are exposed to COM and there are COM wrappers for a particular programming language.

Disadvantages are slow execution time and the fact that there are no COM wrappers for all programming languages.

8.2.1.4 Microsoft Visual Basic for Applications

Microsoft made the decision to stop offering VBA distribution licenses to new customers as of July 1, 2007, and they have expressed that there are no plans to provide VBA product enhancements in the future. As a result of this, though Autodesk will continue to “unofficially” support VBA in its most recent releases of the products, software developers are strongly encouraged to base all future Microsoft Windows based development for AutoCAD based products on the Microsoft .NET Framework (VB .NET, C#, managed C++, etc).

As first introduced with AutoCAD 2010, AutoCAD is not distributing VBA as part of the most recent AutoCAD installation disk. Rather, customers will need to download VBA library separately.

The combination of the powerful ActiveX Automation object model in AutoCAD and Microsoft Visual Basic for Applications (VBA) presents a compelling framework for customizing the AutoCAD software program. With ActiveX Controls and other applications that host VBA (such as Microsoft Office), there is no limit to the objects developers can work with when developing custom solutions for the AutoCAD software.

The main advantage is that developing with VBA is simple and reduces time to completion. In addition VBA allows to:

- visually build a user interface simply by dragging tools from the ActiveX Toolbox that contains ActiveX Controls;
- access extensive code examples to take advantage of the AutoCAD ActiveX object model ;
- embed and run multiple routines in a single AutoCAD drawing.

VBA has the same editor and programming interface that Microsoft Office applications use, so sharing drawing information throughout the company is fast and easy. Besides, there are more than three million trained VBA programmers who can help their enterprise extend the functionality of the AutoCAD software.

The disadvantages are limited functionality and execution speed overhead. Nevertheless, it is claimed that with VBA it is possible to solve about 90% of the programming problems that arise. This satisfies the majority of users and software developers. Yet serious mission- and performance-critical industry applications are still developed using ObjectARX technology.

8.2.1.5 Visual LISP

Visual LISP technology is a tool for code creation in the AutoCAD software application. It is a full-featured, interpretive programming language that power users and developers can use to call AutoCAD commands, system variables, and dialog boxes.

Visual LISP offers a complete development environment, including

- reduced development time using the integrated development environment (IDE), which makes it easier and faster for users and developers alike to create, debug, and deliver AutoLISP-based applications;
- access to ActiveX objects and event reactors;
- source code protection against theft and alteration;
- operating system file-operation functions;
- LISP function extensions for list processing.

Visual LISP is designed especially for engineers and AutoCAD users to allow simple and fast platform customization and task automation.

The Visual LISP environment itself is written in C++, i.e., it is an ObjectARX application.

8.2.1.6 ObjectDBX

ObjectDBX™ is a C++ software library that allows developers to access, read, and write AutoCAD DWG and DXF™ files. ObjectDBX for AutoCAD provides the highest degree of compatibility possible with AutoCAD DWG files; this includes read and write support for the AutoCAD 2012 as well as for earlier releases (e.g., 2000i, 2002, 2005 and 2008 releases) and the new features now available with AutoCAD 2012. ObjectDBX is used internally by Autodesk to provide DWG support in non-AutoCAD-based products such as Autodesk Vault, Autodesk Revit and Autodesk Inventor.

Software applications that link with the ObjectDBX libraries are called host applications and do not require the presence of an AutoCAD application. ObjectDBX does not impose a user interface and provides the framework to support in-memory file operations and custom objects.

8.3 MATLAB

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

MATLAB version 7.0 is used in our system models.

8.4 Eigen

Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms. We have used Eigen 3.1.0-alpha2 (release with support for Intel MKL, released on February 6, 2012) for matrix and vector calculations in our system implementation. Below is a condensed list of its main characteristics.

Eigen is versatile. It supports all matrix sizes, from small fixed-size matrices to arbitrarily large dense matrices, and even sparse matrices. It supports all standard numeric types, including *std::complex*, integers, and is easily extensible to custom numeric types. It supports various matrix decompositions and geometry features. Its ecosystem of unsupported modules provides many specialized features such as non-linear optimization, matrix functions, a polynomial solver, FFT, and much more.

Eigen is fast. Expression templates allow to intelligently remove temporaries and enable lazy evaluation, when that is appropriate. Explicit vectorization is performed for SSE 2/3/4, ARM NEON, and AltiVec instruction sets, with graceful fallback to non-vectorized code. Fixed-size matrices are fully optimized: dynamic memory allocation is avoided, and the loops are unrolled when that makes sense. For large matrices, special attention is paid to cache-friendliness.

Eigen is reliable. Algorithms are carefully selected for reliability. Reliability trade-offs are clearly documented and extremely safe decompositions are available. Eigen is thoroughly tested through its own test suite (over 500 executables), the standard BLAS test suite, and parts of the LAPACK test suite.

Eigen is elegant. The API is extremely clean and expressive while feeling natural to C++ programmers, due to expression templates. Implementing an algorithm on top of Eigen is similar to copying pseudocode.

Eigen has good compiler support as it was tested against many compilers to guarantee reliability and work around any compiler bugs. Eigen is also standard C++98 and maintains very reasonable compilation times.

Eigen is Free Software. Starting from the 3.1.1 version, it is licensed under the MPL2, which is a simple weak copyleft license. Earlier versions were licensed under the LGPL3+.

8.5 Boost

Boost provides free peer-reviewed portable C++ source libraries. We have used version 1.49.0 released on February 24th, 2012 21:20 GMT. It is used mainly for objects permutation routine in our application and for some minor functional enhancements of the system code. A brief overview of its properties is as follows.

Boost emphasizes libraries that work well with the C++ Standard Library. Boost libraries are intended to be widely useful, and usable across a broad spectrum of applications. The Boost license encourages both commercial and non-commercial use.

It is aimed to establish "existing practice" and provide reference implementations so that Boost libraries are suitable for eventual standardization. Ten Boost libraries are included in the C++ Standards Committee's Library Technical Report (TR1) and in the new C++11 Standard. C++11 also includes several more Boost libraries in addition to those from TR1. More Boost libraries are proposed for TR2.

Boost works on almost any modern operating system, including UNIX and Windows variants. Popular Linux and Unix distributions such as Fedora, Debian, and NetBSD also include pre-built Boost packages.

The main reason for using Boost is productivity. Use of high-quality libraries like Boost speeds initial development, results in fewer bugs, reduces reinvention-of-the-wheel, and cuts long-term maintenance costs. Besides, since Boost libraries tend to become de facto or de jure standards, many programmers are already familiar with them. Ten of the Boost libraries are included in the C++ Standard Library's TR1, and are thus slated for later full standardization. More Boost libraries are in the pipeline for TR2. Using Boost libraries gives an organization a head start in adopting new technologies. Many organizations already use programs implemented with Boost, like Adobe Acrobat Reader 7.0

Boost is distributed under Boost Software License which grants permission to copy, use and modify the software for any use (commercial and non-commercial) without a fee.

8.6 Software application developed

According to the objectives stated in this paper, the software application based on the synergetic computer theory was elaborated, tested and successfully implemented on the AutoCAD platform.

8.6.1 Overview

The development of the working prototype of the system capable of recognizing the outer peripheries of a building's rooms was organized into three major steps. These were as follows.

- The development and realization of the basic synergetic computer algorithm suitable for vector graphics objects identification in MATLAB.
- (Based on the previous step) The development and realization of the basic synergetic computer algorithm suitable for vector graphics objects identification in the AutoCAD environment.
- (Based on the previous step) The development and realization of the advanced synergetic computer algorithm as a part of HVAC ADS on the AutoCAD platform.

As the first step was intended as basic numerical testing and prototyping of our models, it is not described in the subsequent sections on the implementation part of this thesis. Instead, the second and third steps are analyzed in more detail below.

8.6.2 Architecture

The resultant system's architecture/framework functionality is described by Figures 8.1 and 8.2.

In Fig. 8.1 the synergetic computer's basic functionality realization on the AutoCAD platform is presented. This system's order parameter generation is based on dot product (DP) measures, and thus similar to the standard model of synergetic computer.

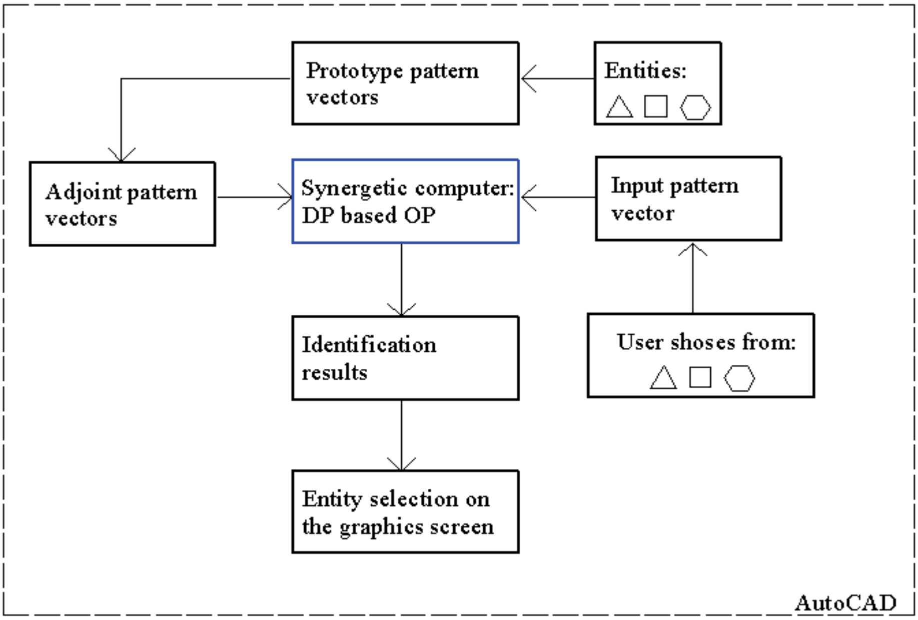


Fig. 8.1 The framework of the synergetic computer's basic functionality system

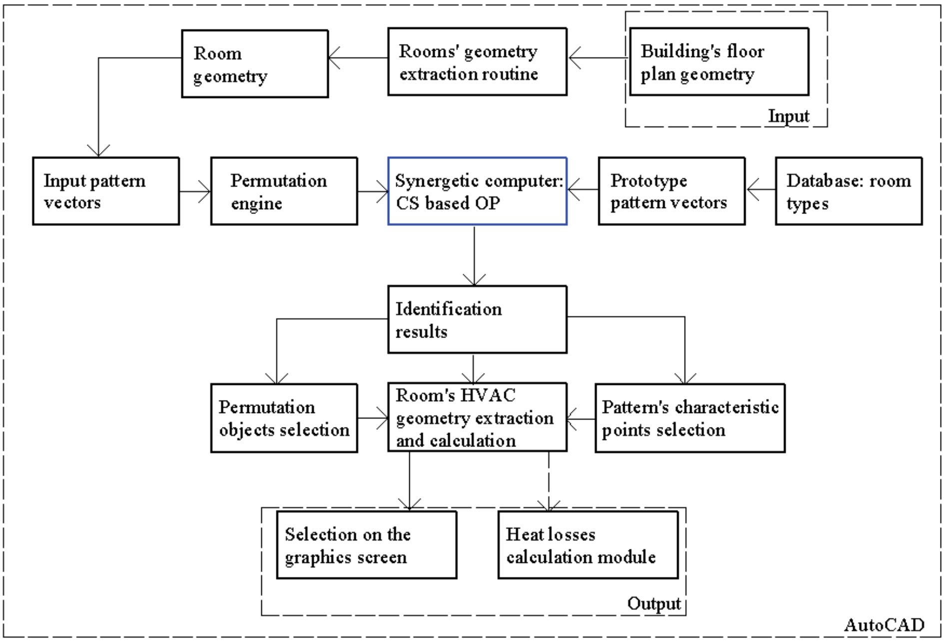


Fig. 8.2 The framework/architecture of the synergetic computer system with advanced functionality (as part of HVAC ADS)

Fig. 8.2 shows functional relations between the components of the synergetic computer system that is used as part of HVAC ADS. This system utilizes the cosine similarity (CS) based order parameter (OP) generation. It also uses permutation of possible characteristic vectors of a given room's geometry to finally select the right pattern. Dashed arrow shows possible data transmission direction to the building's heat loss calculation software component.

8.6.3 Characteristics

In the following subsections, the functionality of the developed programs is discussed. In this part, the already functional software's final versions are discussed. The debug process details, difficulties in the adaptation process of the synergetic computer algorithm to the AutoCAD environment, the performance comparison of different approaches, etc are discussed in the section devoted to developing the suitable algorithm of the synergetic computer for ADS (see section 7).

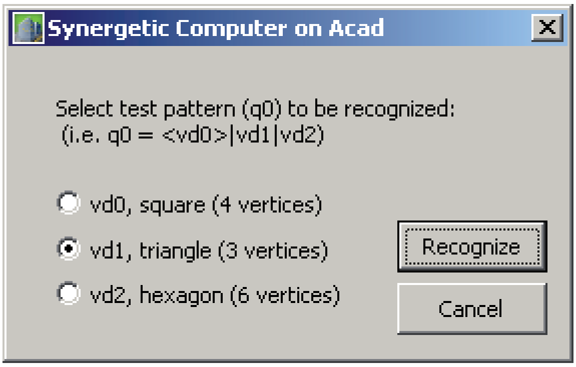
8.6.3.1 Synergetic computer's basic functionality

Implementation of the synergetic computer's core functionality on the AutoCAD platform is an intermediate step towards the realization of the synergetic computer and synergetic neural network (SNN) in the real-life design process as the main tools in ADS modeling.

To this end, the objective was established to create an AutoCAD application that can recognize a number of simple geometric structures. At first the MATLAB prototype was created in order to test the basic functionality of the model, and then the algorithm was implemented in the AutoCAD environment. For the sake of simplicity of presentation, only three different patterns were implemented in the AutoCAD environment. Actually, the number of patterns tested (in MATLAB) was bigger, and noisy patterns were elaborated as well.

The prototype patterns for our case were chosen from among the AutoCAD (Acad) polygon entities (more specifically, these constitute polyline objects in Acad database), namely, the triangle, square and hexagon.

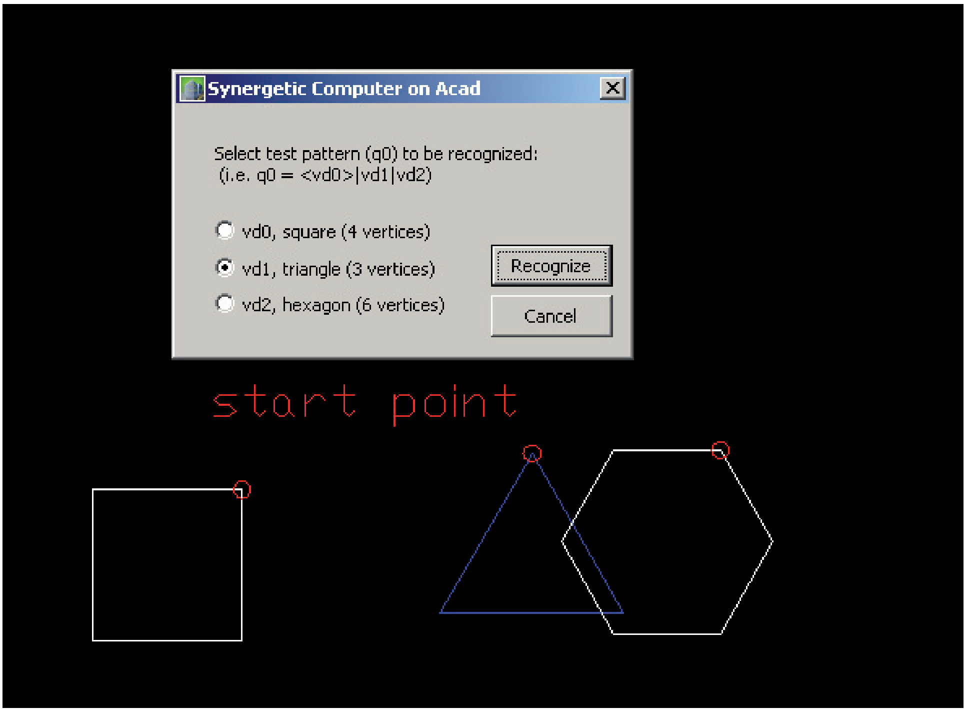
The user interface of the resultant system is shown in Fig. 8.3.



Select test pattern (q0) to be recognized:
(i.e. q0 = <vd0>|vd1|vd2)

○ vd0, square (4 vertices)
● vd1, triangle (3 vertices)
○ vd2, hexagon (6 vertices)

Fig. 8.3 GUI of SNN application



Select test pattern (q0) to be recognized:
(i.e. q0 = <vd0>|vd1|vd2)

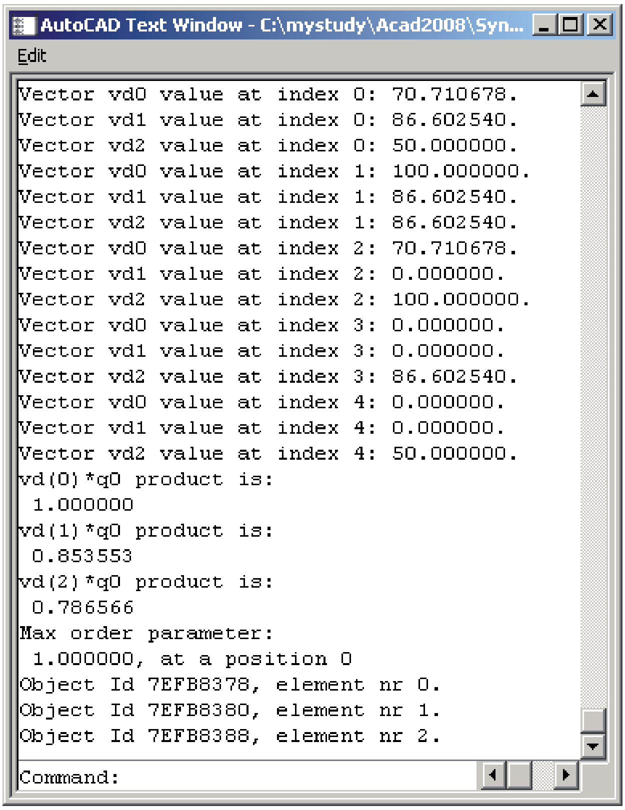
○ vd0, square (4 vertices)
● vd1, triangle (3 vertices)
○ vd2, hexagon (6 vertices)

Fig. 8.4 AutoCAD graphics screen, the identified pattern is selected

The user may choose the pattern desired by selecting the respective radio button, then, after pressing the action button, the system performs the recognition process and selects the identified polygon by changing its color property (to blue), see Fig. 8.4.

The location and position of test patterns has no effect on the recognition results. The polygons may be rotated, or, even overlapped with each other, the system can still successfully identify the figures.

In Fig. 8.5 the Acad text window with the implemented system's output messages is shown.



```
Vector vd0 value at index 0: 70.710678.
Vector vd1 value at index 0: 86.602540.
Vector vd2 value at index 0: 50.000000.
Vector vd0 value at index 1: 100.000000.
Vector vd1 value at index 1: 86.602540.
Vector vd2 value at index 1: 86.602540.
Vector vd0 value at index 2: 70.710678.
Vector vd1 value at index 2: 0.000000.
Vector vd2 value at index 2: 100.000000.
Vector vd0 value at index 3: 0.000000.
Vector vd1 value at index 3: 0.000000.
Vector vd2 value at index 3: 86.602540.
Vector vd0 value at index 4: 0.000000.
Vector vd1 value at index 4: 0.000000.
Vector vd2 value at index 4: 50.000000.
vd(0)*q0 product is:
 1.000000
vd(1)*q0 product is:
 0.853553
vd(2)*q0 product is:
 0.786566
Max order parameter:
 1.000000, at a position 0
Object Id 7EFB8378, element nr 0.
Object Id 7EFB8380, element nr 1.
Object Id 7EFB8388, element nr 2.

Command:
```

Fig. 8.5 AutoCAD text window, program control messages

Facts

Total memory usage (without AutoCAD, debug) - ~464K
Size of the file (ARX, debug) – 224KB
Size of the file (ARX, release) – 127KB
Total number of files used (Debug) – 36
Total number of lines of code (source files): 731
Runs under AutoCAD 2008 or higher

8.6.3.2 Synergetic computer as part of HVAC ADS

The prototype system capable of finding and selecting the room's outer walls, windows and floor area was built as an ObjectARX application. In this subsection, the working principles, outputs and user interface of the resultant program are discussed.

The prototype patterns for our case may be composed of three values, i.e., of the distances between the base point, and the characteristic points of the room's geometry, as shown in Fig. 8.6

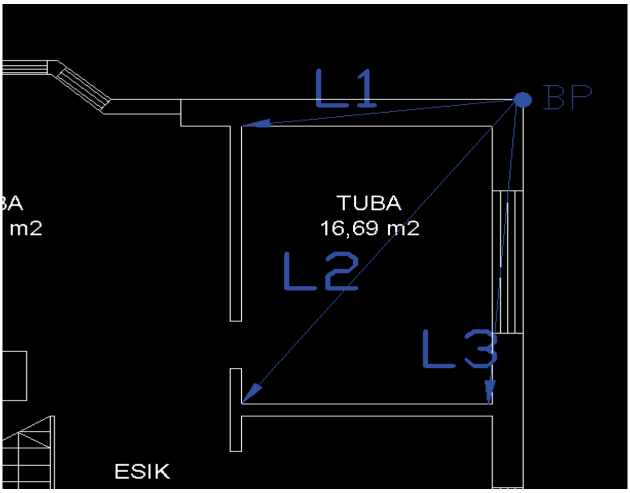


Fig. 8.6 Representation of the room's geometry as prototype vector elements in the AutoCAD graphics system

The system was tested with three different prototype patterns with different vector lengths (shown as raw vectors):

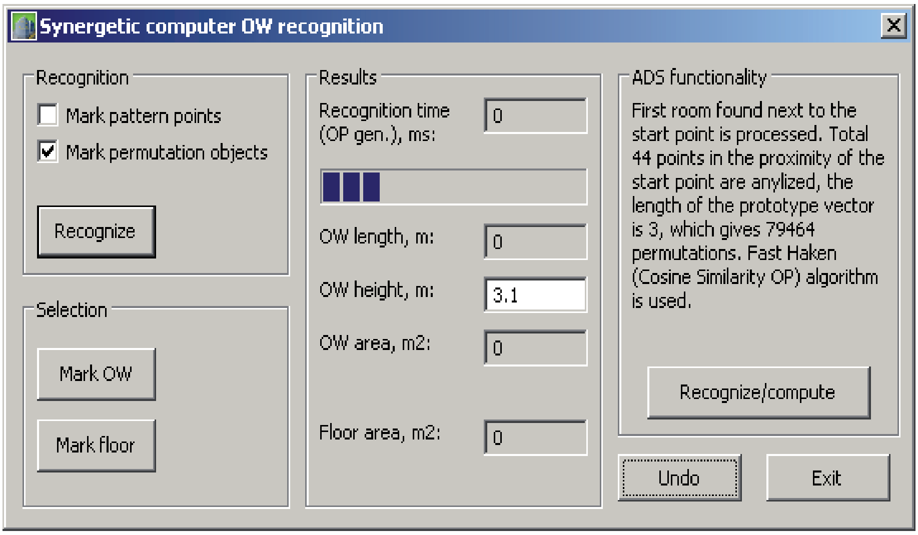
$$v_0 = (L_{01}, L_{02}, L_{03})$$

$$v_1 = (L_{11}, L_{12})$$

$$v_2 = (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}).$$

Actually, the number of prototype patterns may be arbitrary, as this affects only the computational speed of the application. In fact, in real-life applications the number of prototype patterns equals to the number of room types stored. In this prototype build, for the sake of presentation simplicity, only the first room found is analyzed. Other rooms are treatable exactly in the same manner.

For convenient testing and visualization of its results, the GUI of the underlying system was created. The user interface of the resultant system is shown in Figs. 8.7 and 8.8. In the later build the window selection functionality was added.


Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☑ Mark permutation objects

Results

Recognition time (OP gen.), ms: 0

OW length, m: 0

OW height, m: 3.1

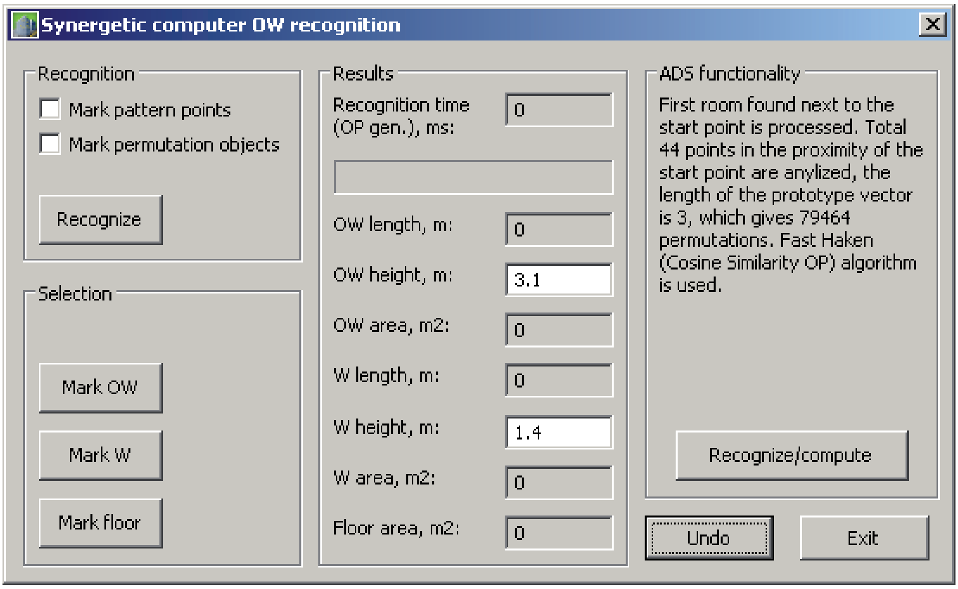
OW area, m2: 0

Floor area, m2: 0

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.7 GUI of the application, build #1


Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☐ Mark permutation objects

Results

Recognition time (OP gen.), ms: 0

OW length, m: 0

OW height, m: 3.1

OW area, m2: 0

W length, m: 0

W height, m: 1.4

W area, m2: 0

Floor area, m2: 0

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

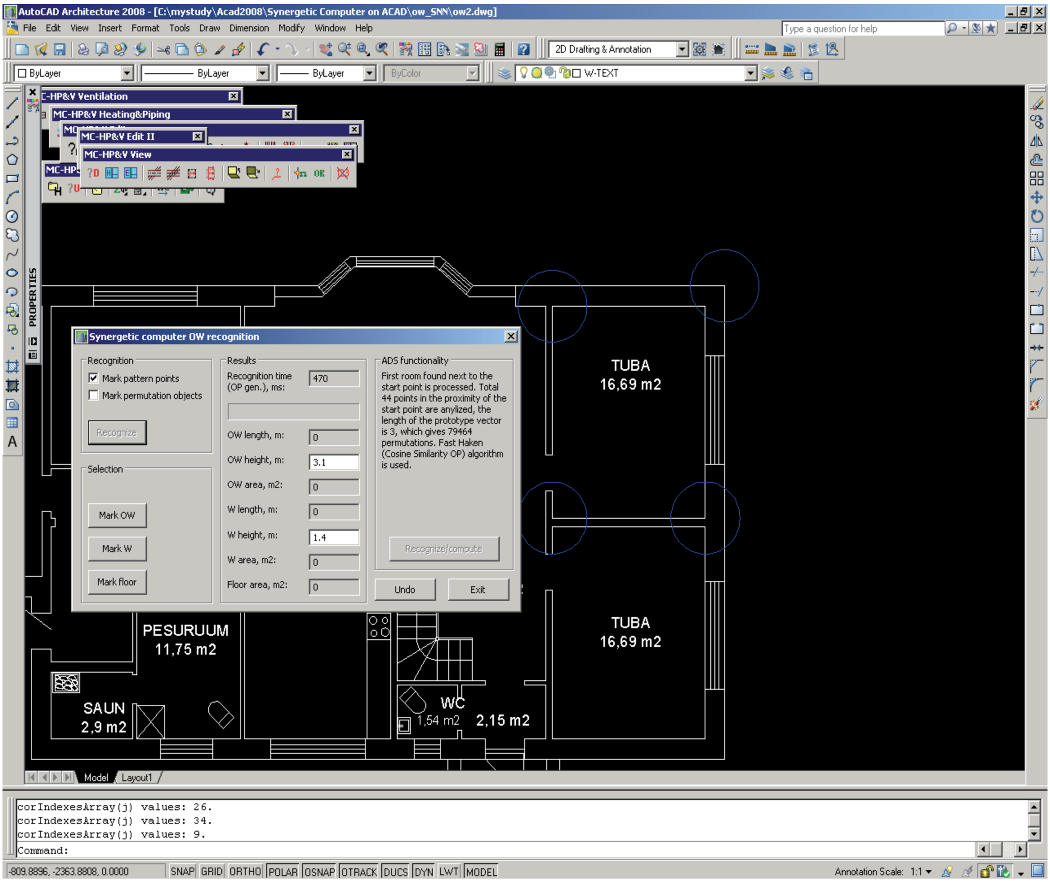
Fig. 8.8 GUI of the application, build #2

The dialog has four major groups/parts: Recognition, Selection, Results and ADS functionality.

In the Recognition group/part, by pressing the Recognize button, depending on the check box selected, the system performs pattern recognition and identifies the pattern’s characteristic points/permutation objects on the graphics screen (Figs. 8.9, 8.10). There are a total of 79,464 permutations (test patterns, i.e.,

$C_k^r = \frac{k!}{(k-r)!} = \frac{44!}{(44-3)!} = 79,464$) in this room identification process, from which the winning pattern is selected.

As the winning pattern is recognized, it is possible to extract the room's geometry, and then use this in automated engineering calculations, e.g., in the room's heat loss calculation. For visual identification of the room's geometry, there are three buttons in the Selection section of the dialog box. These are Mark OW (Outer Walls), Mark W (Window) and Mark floor buttons. The output of the action of Mark OW button is shown in Fig. 8.11. In the middle section of the dialog box, the results of the calculations are shown: the speed of the recognition process, length/area of the outer wall, and the floor area of the room.



AutoCAD Architecture 2008 - [C:\mystudy\Acad2008\Synergetic Computer on ACAD\ow_SNN\ow2.dwg]

Synergetic computer OW recognition

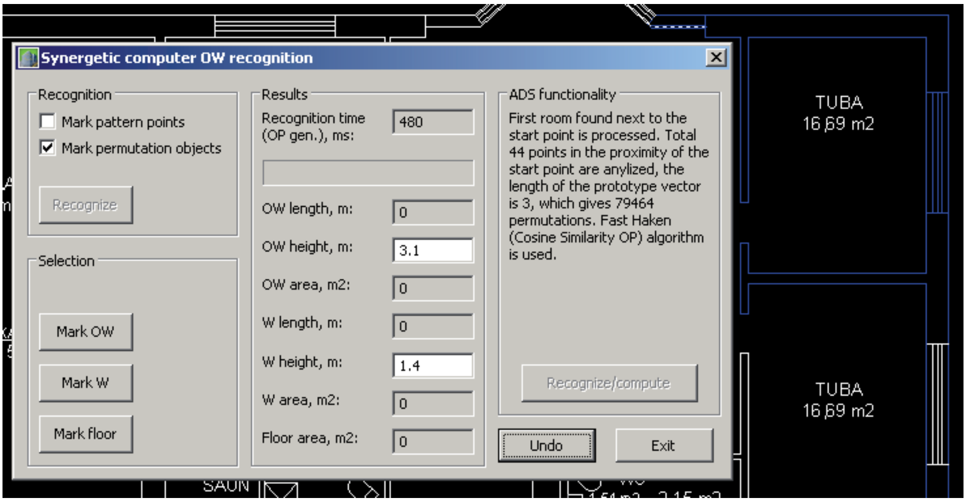
Recognition
☑ Mark pattern points
☐ Mark permutation objects

Results
Recognition time (OP gen.), ms: 470
OW length, m: 0
OW height, m: 3.1
OW area, m2: 0
W length, m: 0
W height, m: 1.4
W area, m2: 0
Floor area, m2: 0

ADS functionality
First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

```
corIndexesArray(j) values: 26.
corIndexesArray(j) values: 34.
corIndexesArray(j) values: 9.
Command:
```

Fig. 8.9 AutoCAD graphics screen – "Mark pattern points" option selected



☐ Mark pattern points

☑ Mark permutation objects

Recognition time (OP gen.), ms: 480

OW length, m: 0

OW height, m: 3.1

OW area, m2: 0

W length, m: 0

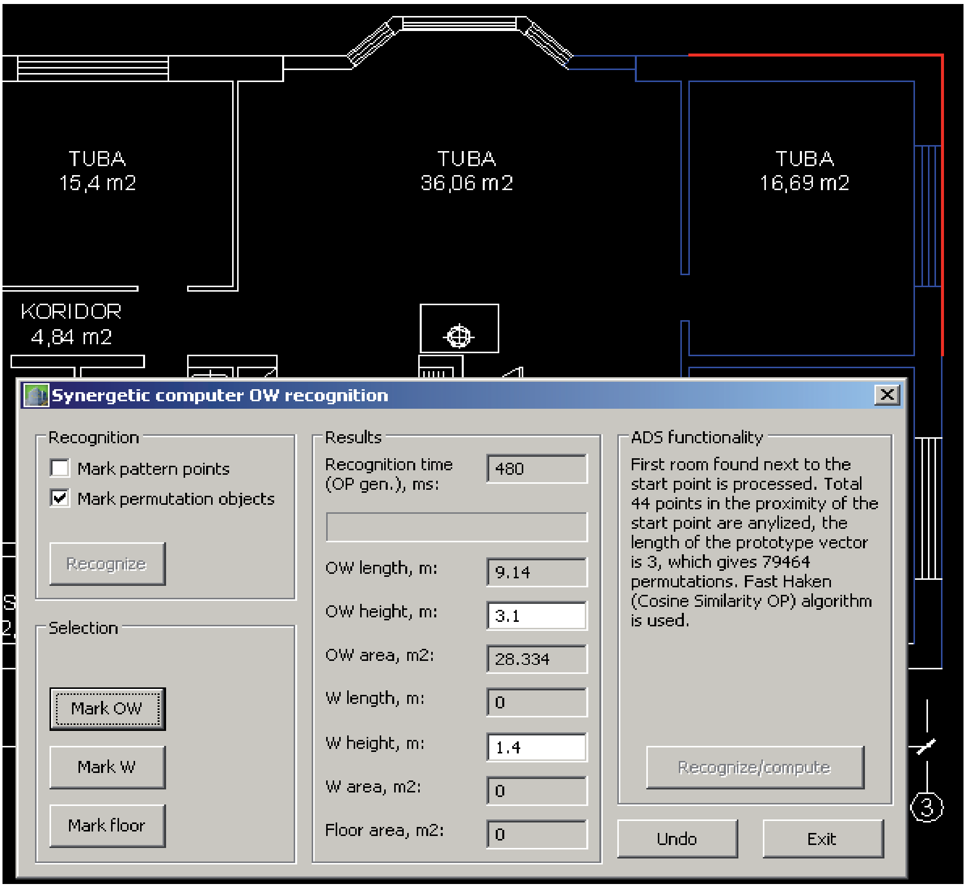
W height, m: 1.4

W area, m2: 0

Floor area, m2: 0

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.10 AutoCAD graphics screen – “Mark permutation objects” option selected



☐ Mark pattern points

☑ Mark permutation objects

Recognition time (OP gen.), ms: 480

OW length, m: 9.14

OW height, m: 3.1

OW area, m2: 28.334

W length, m: 0

W height, m: 1.4

W area, m2: 0

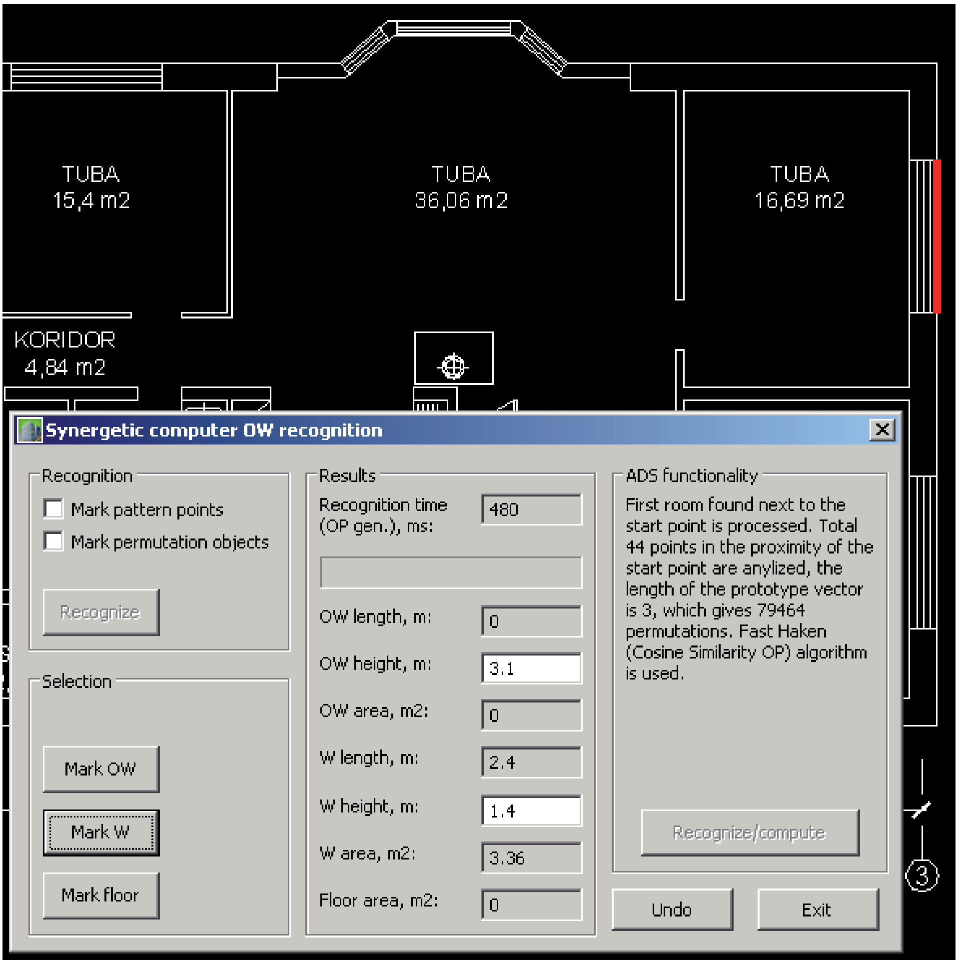
Floor area, m2: 0

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.11 AutoCAD graphics screen – “Mark OW” button pressed

The results of window and floor selection are shown in Figs. 8.12 and 8.13 respectively.

In the middle section of the dialog box, the results of the calculations are shown: the speed of the recognition process, length/area of the outer wall, length/area of the outer wall's window, and the floor area of the room.



Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☐ Mark permutation objects

Selection

Results

Recognition time (OP gen.), ms: 480

OW length, m: 0

OW height, m: 3.1

OW area, m2: 0

W length, m: 2.4

W height, m: 1.4

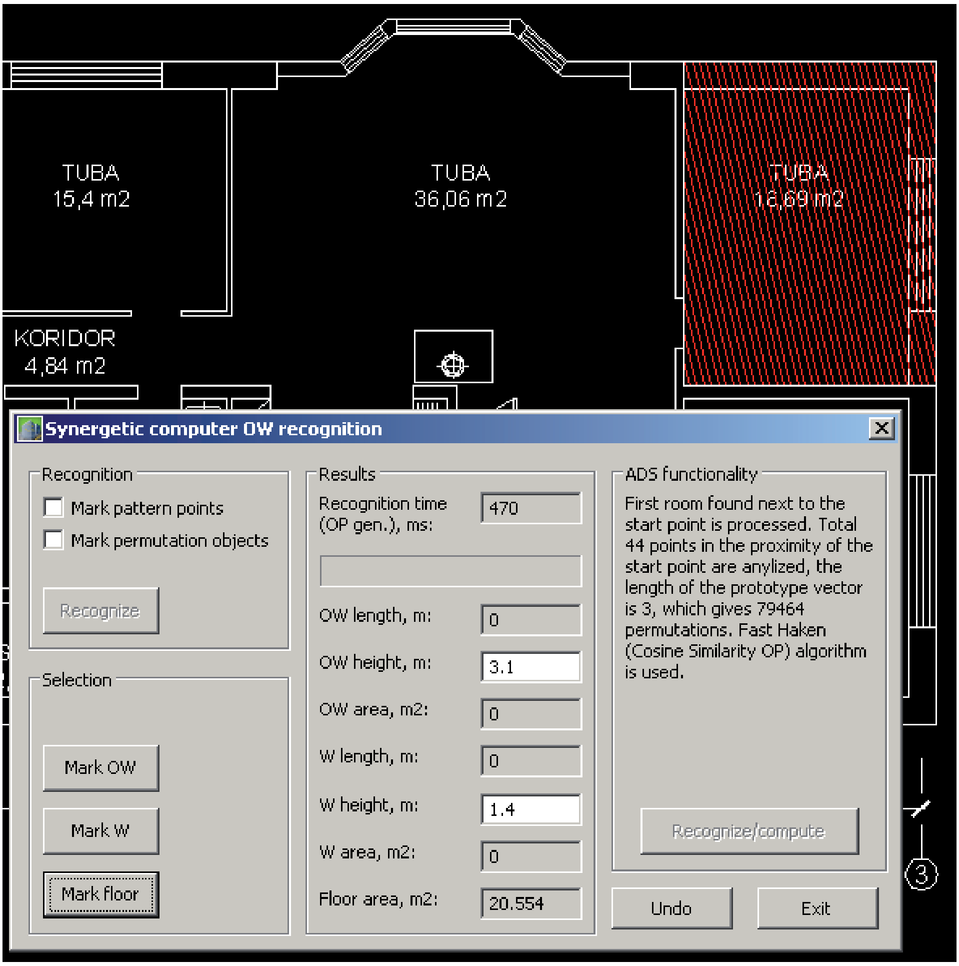
W area, m2: 3.36

Floor area, m2: 0

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.12 AutoCAD graphics screen – "Mark W" button pressed



Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☐ Mark permutation objects

Selection

Results

Recognition time (OP gen.), ms: 470

OW length, m: 0

OW height, m: 3.1

OW area, m2: 0

W length, m: 0

W height, m: 1.4

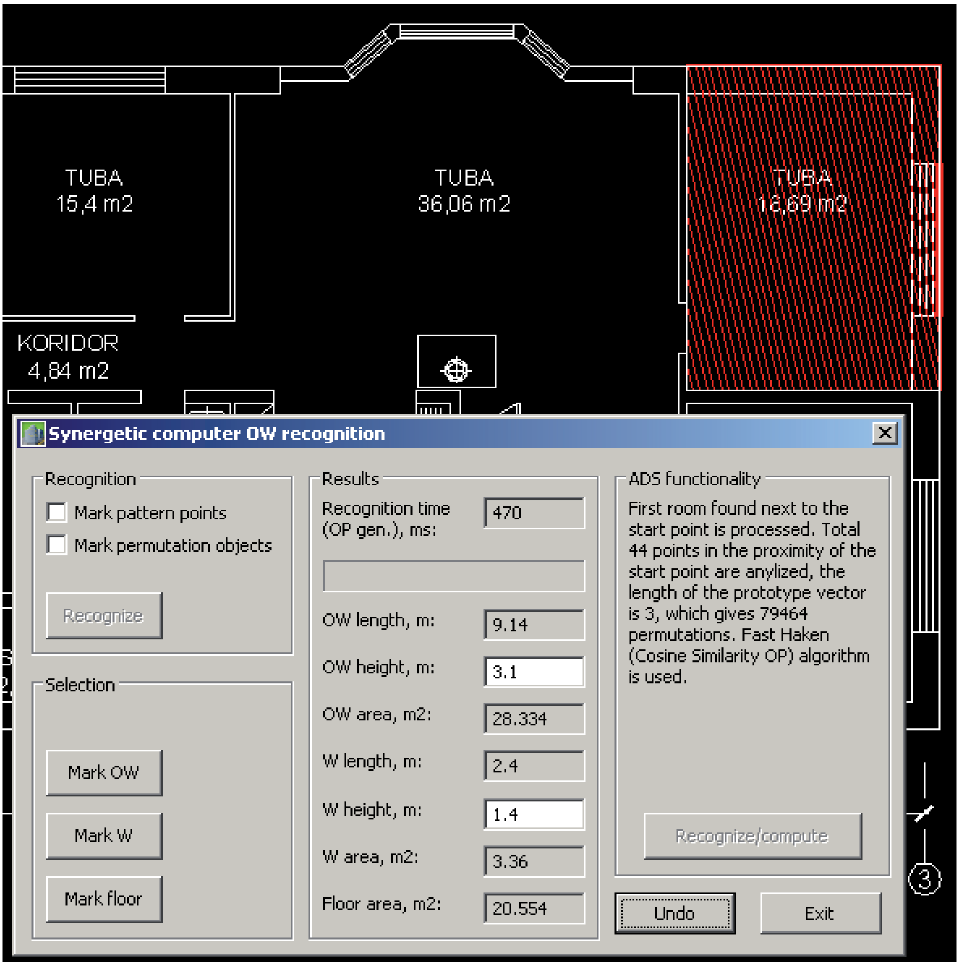
W area, m2: 0

Floor area, m2: 20.554

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.13 AutoCAD graphics screen – “Mark floor” button pressed



☐ Mark pattern points
☐ Mark permutation objects

Recognition time (OP gen.), ms: 470
OW length, m: 9.14
OW height, m: 3.1
OW area, m2: 28.334
W length, m: 2.4
W height, m: 1.4
W area, m2: 3.36
Floor area, m2: 20.554

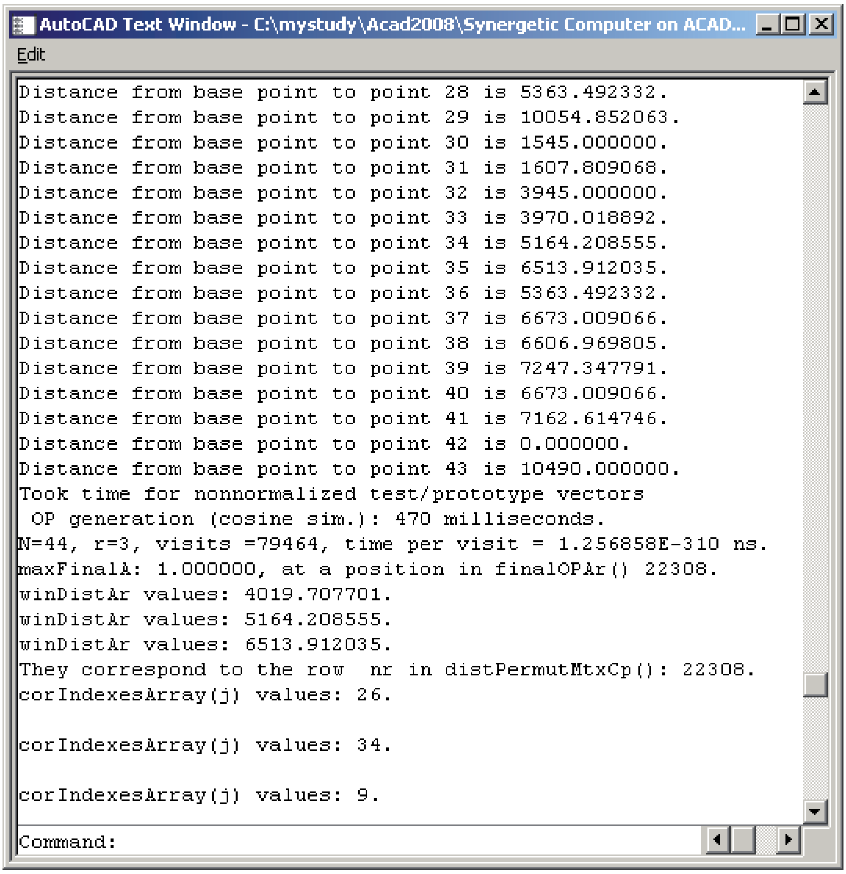
First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 8.14 AutoCAD graphics screen – “Recognize/compute” button pressed

ADS functionality group has only one button, intended for ADS essence/performance visualization. That is, by pressing this button the system performs automatic recognition and computation of the room’s geometry (see Fig. 8.14) and selects all the meaningful objects of the room on the graphics screen.

As the user controls are implemented on the modal dialog, Undo button is used for convenient undoing while keeping the focus on the window.

An excerpt of the program debug/control messages is shown in Fig. 8.15.



```
Distance from base point to point 28 is 5363.492332.
Distance from base point to point 29 is 10054.852063.
Distance from base point to point 30 is 1545.000000.
Distance from base point to point 31 is 1607.809068.
Distance from base point to point 32 is 3945.000000.
Distance from base point to point 33 is 3970.018892.
Distance from base point to point 34 is 5164.208555.
Distance from base point to point 35 is 6513.912035.
Distance from base point to point 36 is 5363.492332.
Distance from base point to point 37 is 6673.009066.
Distance from base point to point 38 is 6606.969805.
Distance from base point to point 39 is 7247.347791.
Distance from base point to point 40 is 6673.009066.
Distance from base point to point 41 is 7162.614746.
Distance from base point to point 42 is 0.000000.
Distance from base point to point 43 is 10490.000000.
Took time for nonnormalized test/prototype vectors
 OP generation (cosine sim.): 470 milliseconds.
N=44, r=3, visits =79464, time per visit = 1.256858E-310 ns.
maxFinalA: 1.000000, at a position in finalOPAr() 22308.
winDistAr values: 4019.707701.
winDistAr values: 5164.208555.
winDistAr values: 6513.912035.
They correspond to the row  nr in distPermutMtxCp(): 22308.
corIndexesArray(j) values: 26.

corIndexesArray(j) values: 34.

corIndexesArray(j) values: 9.
Command:
```

Fig. 8.15 AutoCAD text window with program control messages (excerpt)

Facts
Total memory usage (without AutoCAD, debug) - ~5596K
Size of the file (ARX, debug) – 284KB
Size of the file (ARX, release) – 204KB
Total number of files used (Debug) – 40
Total number of lines of code (only source files, without header and library files): 1801
Runs under AutoCAD 2008 or higher

9 RESULTS

According to the goal and objectives of this thesis, the results of this work can be grouped into two categories: theoretical and practical, and they are summarized below. For a more specific discussion, please refer to Chapter 1, and Chapters 4, 6, 7 and 8, which constitute the direct contribution of this research along with sections 5.2 and 5.3, which form the additional, analytical contribution of this work. Chapters 2 and 3, and section 5.1, although being thoroughly compiled and analyzed, are not considered as the author's direct contribution, but serve primarily as aiding, informative modules.

9.1 Theoretical results

In the theoretical part of the work, a thorough analysis of the use of synergetics and synergetic computer technology in the HVAC design was performed. Particularly, the problem of modeling the creative part of the engineering design process was analyzed and synergetics was proposed as the main tool for that purpose. The research performed led to the introduction of the novel notion of the Autonomous Design System (ADS) and to defining its necessary properties. The standard model of the synergetic computer was analyzed and modified accordingly, to fit our purposes for modeling the creative components of HVAC ADS. Finally, the general theory of synergetics and self-organization was analyzed from the philosophical point of view.

Synergetics provides mathematical tools for coping with the self-organization phenomenon. These tools are based on the combination of the differential equations theory and stochastic modeling. The principles of synergetics can be used in a great variety of scientific disciplines ranging from theoretical physics to social sciences. A hypothesis was raised that the creative component of engineering design obeys the rules of self-organization (or even more - it is a self-organizing process in itself), and thus synergetics can be used for modeling it. The problem of modeling the creative part of design was analyzed from the general point of view, and different ways for modeling that component were proposed. As a result, a corresponding computational model was proposed and, eventually, successfully realized in the form of a computer program.

The novel notion of ADS was proposed and its characteristics were defined and explained. ADS is considered as an advanced CAD system that has the Artificial Intelligence (AI) functionality, and particularly the functionality to deal with creative components of the engineering design process. A couple of cybernetic models, which can be further optimized by the methods of synergetics, were proposed during the research conducted. As an example, the theoretical consideration of ADS in one particular application domain, i.e., the HVAC field was explained.

General analysis of the process of engineering design was performed. Basic properties, characteristics, and problems of the engineering design process were highlighted, both from the general as well as conceptual points of view. A brief overview of the AI field of expertise was also given in this work, gradually preparing the reader for the more advanced topics considered in this paper.

General analysis of the synergetic computer theory and synergetics (as well as self-organization) was performed from the general and philosophical points of view. This paper provides an introduction to the concepts of synergetics, self-organization, and the problem of modeling the creative part of the engineering design process. A brief philosophical analysis of the related issues is also presented. A contemporary analysis of the corresponding literature is given and major researchers in this field are considered as well.

The mathematical theory of the standard model of the synergetic computer was thoroughly analyzed, modified and suited for our particular purposes. The adaptation of the standard model of synergetic computer to HVAC ADS was performed in several steps. First, the theoretical base was prepared and the numerical prototype of the system was created in Matlab. Then, the system was adapted to the general AutoCAD environment and functionality. Finally, the advanced algorithm was elaborated and tested as part of a particular synergetic computer HVAC application on AutoCAD, i.e., HVAC ADS. The adaptation process of the standard model of synergetic computer to AutoCAD, and more specifically to the HVAC ADS environments, consisted of two parts: theoretical and practical ones. In the theoretical part, the appropriate algorithm or method was deduced, and in the practical part, that algorithm or method was tested in a specific computational environment, i.e., experiments were performed. For the first time ever, the synergetic computer algorithm was adapted to and successfully implemented on the AutoCAD platform and in the HVAC field.

At the beginning of this work the following hypotheses were raised and the following research questions formulated. *Hypothesis*: synergetics (synergetic computer theory) can be successfully used to model the creative component of the engineering design process. *Sub hypothesis:* synergetic algorithms (and the self-organization paradigm in general) reflect the creativity of the human being. Thus, creativity is derived from self-organization. The *research questions*: Can synergetics help to model the creative part of the engineering design process? Can synergetic algorithms help to develop the Autonomous Design System (ADS) software? Can the synergetic computer theory be used for the outer walls recognition mechanism in the HVAC software on the AutoCAD platform? The hypotheses raised were actually proved (through numerical testing on the CAD platform). This was one of the definite results of this work. After the research had been conducted, the research questions could also be answered positively.

An additional theoretical result of this research was developing the conceptual framework for modeling the creative component of the engineering design process in the AutoCAD environment using the synergetics-based approach. The conceptual framework and research methodology were established during the course of this work. These proved to be consistent and successful, and can thus be used in other related research tasks/activities.

As an additional contribution of the research, an application for the patent for the method of synergetic recognition of vector graphics objects in the AutoCAD (or another computational) environment was filed a couple of months ago, and the probability of obtaining the respective patent is considered to be very high.

9.2 Practical results

According to the objectives stated in this paper, the software application based on the synergetic computer theory was elaborated, tested and successfully implemented on the AutoCAD platform. This is the first documented usage of a synergetic computer algorithm on AutoCAD and in HVAC CAD/ADS.

The major practical outcome of this research was developing a software framework based on a synergetic computer implementation which can be run on the AutoCAD platform. This constituted development of a fully functional component of ADS, i.e., the component capable of a building's outer walls geometry recognition. The recognition algorithm uses the advanced synergetic computer model which is optimized for one particular design task and relies solely on vector graphics information, i.e., no additional parametric data is needed. This component can be used in real-life software applications (both on traditional ones and on BIM platforms, particularly in the HVAC domain) that use the visual recognition tasks in their work.

Actually, two separate application modules were created on the AutoCAD platform. The first one was implementation of the synergetic computer's basic functionality, the second was the application of the synergetic computer's advanced algorithm for a particular HVAC ADS design (sub)task.

Thus, implementation of the synergetic computer's core functionality on the AutoCAD platform was an intermediate step towards the realization of the synergetic computer and synergetic neural network (SNN) as the main tools in ADS modeling in the real-life design process. To this end, the objective was established to create an AutoCAD application that could recognize a number of simple geometric structures. First, a MATLAB prototype was created in order to test the basic functionality of the model, and then the algorithm was implemented in the AutoCAD environment. The prototype patterns for our case were chosen from among AutoCAD (Acad) polygon entities (more specifically, these constitute polyline objects in the Acad database), namely, the triangle, square and hexagon. The system was capable of recognizing these simple geometric figures.

Next, based on the previous results, a prototype system capable of finding and selecting the room's outer walls, windows and floor area in the form of an ObjectARX application that runs on the AutoCAD platform was built. As a result, a computer program that utilizes the strengths of the synergetic computer and helps a designer of HVAC systems in his/her daily work was created. Particularly, this software module (as part of HVAC ADS) automatically finds the building's/room's outer peripheries and extracts its geometry for, e.g., further heat loss calculation. In this sense, it acts completely autonomously, the only input data needed is some (vector) graphical representation of the building’s geometry. The system then automatically identifies the rooms’ geometry within a given building plan and extracts the data for further processing. After minor additional work, this prototype program can be successfully used in real-life design scenarios in daily engineering work in the HVAC domain.

10 SUMMARY, FINAL CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

10.1 Summary

The context of this research includes the analysis of possible applications of synergetics (the synergetic computer) in the Heating Ventilation and Air Conditioning (HVAC) field. Specifically, this research addresses issues within the context of the HVAC design process and CAD.

During the course of this work the hypotheses raised at the beginning of the research were proved and the research questions were answered in a quite positive manner. The main goal of this work was to investigate the problem of modeling the creative parts of the engineering design in the HVAC design process and to try to use the principles of synergetics and self-organization paradigm while addressing the problem.

The problem of modeling the creative part of the engineering design process has been considered from both general and philosophical points of view. The standard synergetic computer theory was thoroughly analyzed and its strengths and limitations were highlighted. This analysis made it possible to propose potential ways for the improvement of this theory and to develop the advanced synergetic computer based recognition algorithm that is capable of recognizing vector graphics geometry and that can be used in HVAC ADS applications on the AutoCAD platform. The adaptation of the standard synergetic computer theory to vector graphics recognition scenarios in HVAC ADS is one of the direct contributions of this research.

The results of the analysis of the general theory of the engineering design process are also presented in this thesis.

As the practical side of this research software applications were developed to illustrate the usage of synergetics in the design process. The software applications are based on the AutoCAD platform and were developed with the ObjectARX technology, which is an object-oriented C++ application programming interface for developers to use, customize, and extend AutoCAD and AutoCAD-based products. For instance, one of the programs developed can autonomously identify the room geometry on a given building's plan, extract the room's outer peripheries geometry and calculate the areas of the room's outer surfaces (outer walls, windows, floors). Possible ways for connection of this computational module to other applications, for example in the case of heat losses calculation, are outlined. The recognition algorithm uses solely visible vector graphics based objects geometry and, being built on synergetics principles, is considered to be very similar to the visual perception mechanisms of biological species (including humans). The principles of this algorithm and the software module developed can also be successfully implemented in other CAD and BIM systems for different purposes.

One of the important contributions of this work is the introduction and definition of the novel notion of the Autonomous Design System (ADS) and its realization in one particular HVAC use case.

This research is an interdisciplinary exploration between engineering design, theoretical physics (synergetics) and computer science.

In this work, some of the recent advances in Artificial Intelligence (AI) theories and techniques (synergetics, synergetic computer and Synergetic Neural Network or SNN) are used, alongside with the practical/theoretical and philosophical consideration of traditional AI technologies, e.g., such as artificial neural networks (ANN), genetic algorithms and knowledge-based systems. The research work done is an extension of the author's original investigation in the field of automation of the engineering design process, started already during postgraduate studies.

10.2 Final Conclusions

The goal of this research was the following:

"To analyze the problem of modeling the creative parts of the engineering design in the HVAC design process by using the synergetics approach (theory)."

The objectives were to find out some of the most useful application domains of synergetics and synergetic computer in the given field, to optimize it and to test in real-life situations, in practice. This was accomplished through preparing a solid theoretical and philosophical basis for synergetic modeling, and through developing a software package based on synergetics technology.

The goal and objectives of this thesis were achieved. In this dissertation we showed that it is possible to use synergetics (along with other AI technologies, such as ANN/SNN) to aid the engineering design process and to model the creative parts ofe engineering design. We identified the most useful applications for synergetic computer algorithms in the HVAC design. In addition, we tested one of them in real life design practices. For that purpose, a special software framework was developed. The framework's modules can be used separately or together with the already existing software packages, e.g., the HeatLoss application. In either case the software facilitates and simplifies the design process, minimizing design errors.

We have two groups of results in this work: theoretical and practical ones. In the theoretical part we optimized several synergetic computer technology algorithms in the HVAC design and introduced the novel notion of the Autonomous Design System (ADS). Practical results were successful developing and testing of a synergetic computer software application that uses the developed advanced synergetic computer algorithm capable of recognizing vector graphics geometry, automatic selection of a building plan's rooms and their outer peripheries and extraction of their geometry parameters. These data can be further used for, e.g., rooms' heat loss calculations. The software application is essentially an ObjectARX program that uses advanced permutation engines and runs on the AutoCAD platform.

As there was previously no information about synergetics and synergetic computer technology application possibilities in the HVAC design process, theoretical and practical results of this dissertation may be considered as an innovation in this particular field of expertise. The computer programs developed can, after some additional optimization, be used in real life engineering practices. This constitutes additional value of this dissertation.

The problem of modeling the creative component of engineering design has been analyzed from the general and philosophical point of view. This work has prepared both theoretical and practical knowledge platform for future research and implementations. The author suggests that the subject considered in this work needs further investigation and research.

10.3 Directions for further research

The directions for further research connected with this thesis discussion are outlined below. They may generally be organized into two categories. Firstly, the research and improvement of the synergetic computer recognition algorithm, its further optimization for the HVAC ADS modules, and the analysis/identification of its potential usage cases. Secondly, further development of the HVAC ADS software module itself.

As for the system's computational performance improvement, it is possible to create a parallel numeric synergetic computer model, i.e., synergetic neural network (SNN). The rationale of this is to improve the performance of our model. This additional functionality is not obligatory, nor does it affect the general properties of the model (except for the speed).

Another technical improvement possibility is in the area of attention parameter optimization. In our model, the balanced attention parameters were used (according to (7)) and $\lambda_k = C = B = 1$. Yet it is possible to use the unbalanced attention parameter technique e.g. $\lambda_1 = 0,6$ and $\lambda_2 = 0,4$. In that case the biggest value influences the evolution of the order parameters and it is possible for the initially smaller order parameter to eventually win the competition. Attention parameters could thus be used as an additional instrument to guide the selection of order parameters in situations where selection criteria based solely on ξ_k values are not sufficient. Those situations are likely to arise when dealing with more complicated patterns and process objectives, as e.g., in ADS structures, noisy patterns or, even as simple, as in treatment of vectors of different size, like in

$$v_0 = (L_{01}, L_{02}, L_{03})$$

$$v_1 = (L_{11}, L_{12})$$

$$v_2 = (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}).$$

In that case, if the dot product based order parameter generation is used, depending on the initial values of q_0 and according to the equation $\xi_k = v_k^+ q_0$ theoretically is possible that the right prototype pattern may be recognized erroneously. In such cases e.g. the unbalanced attention parameter techniques could be used in order to correct the system's performance. For example, when the unbalanced λ_k is used the winning order parameter eventually becomes equal to $\xi_k = \sqrt{\lambda_k}$ that is smaller than one (in the case of balanced attention parameters all λ_k -s are equal to one and the winning order parameter becomes equal to one).

It is strongly suggested to use the developed synergetic computer-based recognition mechanism in modeling other creative components of the HVAC engineering design process. Moreover, the principles of synergetics and self-organization could theoretically be used for the improvement and optimization of the engineering design process in general, and in other technical disciplines. Additional studies and research are needed to identify these possible useful usage cases.

Vast potential and need for further development can be seen in the field of extending the module functionality of the created prototype software. Some additional work has to be done to accommodate the outer peripheries recognition computational module into a fully functional and market-ready software application. This, however, is rather a system implementation (programmatic) challenge, as the basic functionality pertaining to the use of the synergetic recognition engine has already been created and tested. An obvious and easily accomplishable step is the connection of the synergetic recognition module with the HeatLoss software, created as a result of the author's previous work (see chapter 7 for details).

To sum up, the recommendations for further research could be as follows:

- Improvement and further optimization of the synergetic computer algorithm. Introducing "arrow of time", making it more "self-organizing".
- Application of this algorithm in modeling of other creative parts of the (engineering) design process.
- Making a fully functional real-life software application ready for commercial use, implementing complete automation for the outer-inner walls recognition mechanism and autonomous calculation of a building's heat losses based solely on vector data representation (i.e., visual perception).

ACKNOWLEDGEMENT

Estonian Ministry of Education and Research is greatly acknowledged for funding and supporting my Ph.D. studies. This support has ensured and kept my motivation during the past four years. Also the contribution of Archimedes Foundation was excellent during several conference presentations. Finally, Doctoral School project “Civil Engineering and Environmental Engineering” code 1.2.0401.09-0080 is acknowledged.

BIBLIOGRAPHY

Papers presented by the candidate

Loginov, Dmitri (2012). On some technical issues and challenges in development and implementation of Autonomous Design System's (ADS) functionality. International Journal of Applied Mathematics and Informatics, 6(3), 142 - 151.

Loginov, D. (2012). Implementation of the Synergetic Computer Algorithm on AutoCAD Platform*. In: Recent Researches in Applied Information Science: 5th WSEAS World Congress on Applied Computing Conference (ACC '12), University of Algarve, Faro, Portugal, May 2-4, 2012.* (Toim.) Nikos Mastorakis, Valeri Mladenov, Zoran Bojkovic. WSEAS Press, 2012, (Recent Advances in Computer Engineering Series; 2), 112 - 117.

Loginov, Dmitri (2012). Advanced Synergetic Computer Algorithm for Vector Graphics Geometry Recognition and Its Realization as Part of HVAC ADS on AutoCAD Platform . *In: Advances in Mathematical and Computational Methods: 14th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering (MACMESE '12), Sliema, Malta, September 7-9, 2012.* (Toim.) Mihaiela Iliescu, Roman Prokop. WSEAS, 2012, (Mathematics and Computers in Science and Engineering Series), 121 - 126.

Loginov, D. (2011). Possibilities of Modeling the Creative Part of Engineering Design Process using the Synergetic Approach. INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES, 5(1), 95 - 104.

Loginov, D. (2010). Synergetic modelling – application possibilities in engineering design. *In: Selected topics in system science and simulation in engineering: 9th WSEAS International Conference on system science and Simulation in engineering (ICOSSSE '10), Iwate Prefectural University, Japan, October 4-6, 2010. (Toim.) Hamido Fujita, Jun Sasaki.* WSEAS Press, 2010, 111 - 117.

Inventions

ADVANCED SYNERGETIC COMPUTER BASED METHOD FOR VECTOR GRAPHICS OBJECTS RECOGNITION. Owner: Dmitri Loginov; Authors: Dmitri Loginov; Priority date: 31.07.2012. Priority number: not yet assigned, the

preliminary decision was the positive one; the final decision will come on 02.2013.

List of references

[1] K. Badhrinath and J. R. Jagannatha Rao, “Modeling for Concurrent Design Using Game Theory Formulations,” *Concurrent Engineering: Research and Applications*, Vol. 4, No. 4, December 1996, pp. 389-399.
[2] Research Opportunities in Engineering Design, *NSF Strategic Planning Workshop Final Report*, (ed.) J. Shah, April 1996.
[3] Y. Reich and S.J. Fenves, “The Formation and Use of Abstract Concepts in Design,” *Concept Formation: Knowledge and Experience in Unsupervised Learning*, Edited by D. H. Fisher, M. J. Pazzani, and P. Langley. Morgan Kaufmann Publishers, Inc., 1991, pp. 323-353.
[4] Council of Graduate Schools, “Graduate education: The backbone of American competitiveness and innovation.” CGS Publication, Washington, D.C., 2007, p. 18
[5] Panos Y. Papalombros and Douglas J. Wilde, *Principles of Optimal Design: Modeling and Computation*, Second Edition, Cambridge University Press, New-York, 2000.
[6] Raymond B. Landys, *Studying Engineering. A Road Map to a Rewarding Career*, Discovery Press, 1995.
[7] J. R. Dixon, “Knowledge-based Systems for Design”, *Journal of Mechanical Design*, Vol. 117, June 1995, pp. 11-16.
[8] J. R. Dixon, “On Research Methodology towards a Scientific Theory of Engineering Design”, *AI EDAM*, 1(3), 1987, pp. 145-157.
[9] N. Cross, *Engineering Design Methods*, John Wiley & Sons Ltd., 1989.
[10] S. E. Lander, “Issues in Multi-agent Design Systems”, *IEEE Expert: Intelligent Systems and their Applications*, Vol. 12, No. 2, 1997, pp. 18-26.
[11] S. Mittal and A. Araya, “A Knowledge-based Framework for Design”, *Artificial Intelligence in Design*, (eds) C. Tong and R.D. Sriram, Academic Press, Inc., 1992, pp. 273-293.
[12] S. M. Kannapan and K. M. Marshek, “Engineering Design Methodologies: A new Perspective”, *Intelligent Design and Manufacturing*, A. Kusiak (Ed.), John Wiley & Sons, Inc., 1992, pp. 3–38.
[13] N.P. Suh, “Axiomatic Design of Mechanical Systems”, *Journal of Mechanical Design*, Vol. 117, June 1995, pp. 2-10.
[14] D. C. Brown, “Routineness Revisited”, *Mechanical Design: Theory and Methodology*, (Eds.) M. Waldron & K. Waldron, Springer-Verlag ,1996, pp. 195-208.
[15] D. C. Brown, B. Chandrasekaran, “Design Problem Solving: Knowledge Structures and Control Strategies”, *Research Notes in Artificial Intelligence Series*, Morgan Kaufmann Publishers, Inc., 1989.

[16] S. Dasgupta, “The structure of Design Processes”, *Advances in Computers*, Vol. 28, M. C. Yovits (Ed.), Academic Press Inc., 1989, pp. 1-67.

[17] M. Sobolewski, “Multiagent Knowledge-Based Environment for Concurrent Engineering Applications”, *Concurrent Engineering: Research and Applications*, Vol. 4, No. 1, March 1996, pp. 89-97.

[18] K. MacCallum and Sue Green, THESYS - Implementation of a Knowledge-Based Design System with Multiple Viewpoints”, *Intelligent CAD Systems II: Implementation Issues*, V. Akman, P. J. W. ten Hagen, and P. J. Veerkamp (Eds.), Springer-Verlag, 1989, pp. 228-244.

[19] N. Cross, *Engineering Design Methods*, John Wiley & Sons Ltd., 1989.

[20] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, “Organizing the Tasks in Complex Design Projects,” in Proc. of Design Theory and Methodology Conference, DTM-90, Chicago, ASME 1990, pp. 39-46.

[21] M. Klein, “Supporting Conflict Resolution in Cooperative Design Systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 6, Nov./Dec. 1991, pp. 1379-1389.

[22] S. Dowlatshahi, “Design Optimization in Concurrent Engineering: A Team Approach,” *IEEE Concurrent Engineering: Research and Applications*, Vol. 5, No. 2, June 1997, pp. 145-154.

[23] S. H. Clearwater, B. A. Huberman, and T. Hogg, “Cooperative Problem Solving,” *Computation: The Micro and the Macro View*, B. Huberman (Ed.), World Scientific, 1992, pp. 33-70.

[24] G. A. Hazelrigg, *Systems Engineering: An Approach to Information-Based Design*, Prentice-Hall, Inc., 1996.

[25] J. Sobieszczanski-Sobieski and R. T. Haftka, “Multidisciplinary Aerospace Design Opimization: Survey of Recent Developments,” 34th AIAA Aerospace Sciences Meeting and Exhibit, 1996.

[26] Cirrus Shakeri, *Discovery of Design Methodologies for the Integration of Multi-disciplinary Design Problems*, Worcester Polytechnic Institute, 1998.

[27] J. W. Forrester, *Urban Dynamics*, MIT Press, 1969.

[28] F. Pena-Mora, R.D. Sriram, and R. Logcher, “Conflict Mitigation System for Collaborative Engineering,” *AIEDAM*, 9, 1995, pp. 101-124.

[29] B. A. Wujek, J. E. Renaud, S. M. Batill, and J. B. Brockman, “Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment,” *Concurrent Engineering: Research and Applications*, Vol. 4, No. 4, December 1996, pp. 361-377.

[30] William M. Newman, Robert F. Sproull, *Principles of interactive computer graphics*, McGraw-Hill, 1979.

[31] James D. Foley, Andries van Dam, and alt., *Computer Graphics. Principles and Practice*, Addison-Wesley, 1987.

[32] National Building Information Model Standard Project Committee (accessed: 2012, 2 March). [Online]. Available: http://www.buildingsmartalliance.org/index.php/nbims/faq/

[33] Chuck Eastman. (August 2009). What is BIM? (accessed: 2012, 2 March) [Online]. Available: http://bim.arch.gatech.edu/?id=402

[34] MP Gallaher, AC O'Connor, JL Dettbarn, Jr., and LT Gilday (August 2004). Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry (Report). National Institute of Standards and Technology. p. iv. (accessed: 2012, 14 Aprill). [Online].Available: http://nvlpubs.nist.gov/nistpubs/gcr/2004/NIST.GCR.04-867.pdf

[35] Yezioro, A; Dong, B; Leite, F, "An applied artificial intelligence approach towards assessing building performance simulation tools", *Energy and Buildings*, 40, (4), 2008, p. 612.

[36] Dan W. Patterson, *Artificial neural networks. Theory and applications*, Prentice Hall International, Inc., 1996

[37] H. Haken, H. Knyazeva, "Arbitrariness in nature: synergetics and evolutionary laws of prohibition," *Journal for General Philosophy of Science*, No.31, 2000, pp. 57–73.

[38] H. Haken, *Synergetics. Introduction and advanced topics*, Springer, 2004.

[39] Heylighen, F. "Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs." *Kybernetes*, 30(5/6), 2001, pp. 691–725.

[40] Heylighen, F. "Representation and Change: A Metarepresentational Framework for the Foundations of Physical and Cognitive Science. Communication and Cognition", 1990

[41] Macklem, P. T. "Emergent phenomena and the secrets of life". *Journal of Applied Physiology,* 104 (6), 2008, pp. 1844–1846.

[42] T. Petrosky; I. Prigogine. "The Liouville Space Extension of Quantum Mechanics". Adv. Chem. Phys.. Advances in Chemical Physics 99: 1–120, 1997, (accessed: 2012, October 3) [Online].Available: http://onlinelibrary.wiley.com/doi/10.1002/9780470141588.ch1/summary

[43] Hermann Haken, *Brain Dynamics. An Introduction to Models and Simulations*, Second Edition, Springer, 2008

[44] Hermann Haken, „*Information and Self-Organization. A Macroscopic Approach to Complex Systems*", Third Enlarged Edition, Springer, 2006

[45] H. Haken, *Synergetic Computers and Cognition. A Top-Down Approach to Neural Nets*, Springer, 2004.

[46] J. Kennedy, R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[47] D. Loginov, "Software Agent-Based CAD Systems: Application in HVAC Field," *Master's Thesis, Dept. Environmental Eng*., Tallinn University of Technology, 2006.

[48] Radu Dobrescu, Matei Dobrescu, Dan Popescu, "Using self-organized living structures principles in the design of Complex Adaptive Systems," in

Proc. 4th WSEAS International Conference on Communications and Information Technology (CIT '10), Corfu Island, Greece, 2010, pp. 155–160.

[49] Jorma Tuomaala, *Creative Engineering Design*, University of Oulu P.O.B 191 FIN-90101 Oulu Finland, 1999.

[50] B. Coffman. (December18, 1996). Anatomy of the Creative Process, MG Taylor Corporation, (accessed: 2012, October 12) [Online]. Available: http://www.mgtaylor.com/mgtaylor/jotm/fall96/7stanat.htm

[51] Nikos Mastorakis, Olga Avramenko, "Fuzzy models of the dynamic systems for evolution of populations," in Proc. 3rd WSEAS International Conference on Mathematical Biology and Ecology, Gold Coast, Queensland, Australia, 2007, pp. 27–32.

[52] G. Colonro, A. Mosca, F. Sartori, "Towards the Design of Intelligent CAD Systems: An Ontological Approach," *Advanced Engineering Informatics*, Vol.21, No.2, 2007.

[53] Martináková-Rendeková Z., "Systems theoretical modelling in musicology," In: Mastorakis, N. (ed.), *Mathematics and Computers in Modern Science. Acoustics and Music, Biology and Chemistry, Business and Economics*, World Scientific and Engineering Society (WSEAS) Press, 2000, pp. 122–127.

[54] Leo Näpinen, "Understanding of the World and the Scientific Paradigm of Self-Organization*", Studia Philosophica,* Vol.IV, No.40, 2004, pp. 156–177.

[55] Supakit Siripanadorn, Wipawee Hattagam, Neung Teaumroong, "Anomaly detection using self-organizing map and wavelets in wireless sensor networks," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 291–297.

[56] Leo Näpinen, "Philosophical Foundations of Synergetic Modelling," Proceedings of the Estonian Academy of Sciences. Humanities and Social Sciences, Vol.4, No.42, 1993, pp. 378–390.

[57] Karl Popper, All Life is Problem Solving, Routledge Books, 1999, pp. 85–86.

[58] Вихалемм, Р, "К вопросу о концептуальных системах химии (Conceptual systems in chemistry)", А. А. Печенкин (Отв. ред.). История науки в философском контексте. Посвящается памяти Владимира Ивановича Кузнецова (1915–2005) СПб: РХГА, 2007, pp. 49–70.

[59] A. Mets, "Ilya Prigogine'i raamatu "Tõsikindluse lõpp. Aeg, kaos ja uued loodusseadused" filosoofiline analüüs ja loodusseaduste käsitus (The philosophical analysis of I. Prigogine's book "The End of Certainty: Time, Chaos, and the New Laws of Nature" and treatment of natural laws)," Magistriväitekiri (Master's Thesis), Tartu, 2007.

[60] Leo Näpinen, "The need of the historical understanding of nature in physics and chemistry*", Foundations of Chemistry*, Vol.1, No.9, 2007, pp. 65–84.

[61] S. J. Gould, “The evolution of life on the earth”, *Scientific American*, 271(4), 1994, pp. 84–9.

[62] Stuart A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*, New York: Oxford University Press, 1993.

[63] Stuart A. Kauffman, *At Home in the Universe: The Search for the Laws of Self-organization and Complexity*, New York: Oxford University Press, 1995.

[64] Prigogine Ilya, Isabelle Stengers, “The new alliance, part two: An extended Dynamics: towards a human science of nature”, *Scientia*, 112, 1977, pp. 643–653.

[65] Zwierlein, Eduard, “The paradigm of self-organization and its philosophical foundation”, *Springer Series in Synergetics*, 61, On Self-Organization, 1994, pp. 288–298.

[66] F. A. Hayek, *The Fatal Conceit. The Errors of Socialism*, The University of Chicago Press, 1988.

[67] Leo Näpinen, “Iseorganiseerumismõtlemine, selle omandamise vajadus Eesti ühiskonnas (Self-organizing thinking, the need for acquirement it in Estonian society)”, *Teaduslugu ja nüüdisaeg, IX, Teadusfilosoofia ja teadusmetodoloogia Eestis: olukord ja perspektiivid. Tallinn: Teadusajaloo ja Teadusfilosoofia Eesti Ühendus, Tartu Ülikooli Filosoofia Osakond*, 1994, pp. 158–181.

[68] Francis Heylighen, “The science of self-organization and adaptivity,” Center "Leo Apostel", Free University of Brussels, Belgium, 2001.

[69] W. D. Hillis, *The Connection Machine*, MIT Press, Cambridge, MA, 1985.

[70] W. S. McCulloch, W. H. Pitts, *Bull. Math. Biophys.*, 5, 1943, pp. 115–133.

[71] R. Rosenblatt, *Principles of Neurodynamics*, Spartan Books, New York, 1962.

[72] M. Minsky, S. Papert, *Perseptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, 1969.

[73] T. J. Seinowski, C. R. Rosenberg, “Complex systems,” 1987.

[74] A. Newell, H. A. Simon, *Human Problem Solving*, Prentice-Hall, Engelwood Cliffs, NJ, 1972.

[75] M. L. Minsky, *The Society of Mind*, Simon and Shcuster, New-York, 1986.

[76] P. Smolensky, *Behavioral and Brain Sciences*, 11, 1988, pp. 1–23.

[77] Tohru Irie, Hiroshi Maeda, Norikazu Ikoma. “Synergetic Stereo Matching Algorithm for Occlusion and Reversal Position”, Journal of Advanced Somputational Intelligence and Intelligent Informatics. Volume 7, Number 2, 2003, pp. 178-188

[78] U. R. Dhond and J. K. Aggarwal, “Structure from Stereo – A Review”, IEEE Trans. on SMC. Vol. 19, No. 6, 1489-1510, 1989

[79] T. Matsuura, K. Satoh, Y. Nakamura, Y. Ohta "Occlusion Detectable Stereo", The Trans. of the Institute of Electronics Information and Communication Engineering, Vol. J80-D-II, No. 6, 1432-1440, 1997

[80] M. Tagaki, H. Shimoda. "Handbook of image analysis", University of Tokyo Press, 1991

[81] Y. Ohta, H. Yamada. "Pattern Matching Using Dynamic Programming", The Trans. of the Information Processing Society, Vol. 30, No. 9, 1058-1066, 1989

[82] Sanya Liu, Zhi Liu, Jianwen Sun, Lin Liu. "Application of Synergetic Neural Network in Online Writeprint Identification*", International Journal of Digital Content Technology and its Applications*. Volume 5, Number 3, March 2011, pp. 126-135

[83] Jiexun Li, Rong Zheng, Hsinchun Chen, "From fingerprint to writeprint*", Communication of the ACM*, ACM press, vol. 49, no. 4, pp.76-82, 2006.

[84] Burrows, J. F., "Word patterns and story shapes: The statistical analysis of narrative style", Literary and Linguistic Computing, vol. 2, pp.6-67, 1992.

[85] Lucia-Antoneta Chicos, Gheorghe Oancea, Camil Lancea, Daniel Bancila, "Software system of integrated and simultaneous engineering," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 238–241.

[86] Zuzana Martináková-Rendeková, "Musical archetypes and memes – basic natural principles of musical work," in Proc. 7th WSEAS International Conference on Acoustics & Music: Theory & Applications, Cavtat, Croatia, 2006, pp. 13–17.

[87] Anon Sukstrienwong, "Genetic algorithms for multi-objectives problems under its objective boundary," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 38–45.

[88] Fisher, R.A.; Yates, F., *Statistical tables for biological, agricultural and medical research* (3rd ed.), Oliver & Boyd, London, 1948, pp. 26–27.

[89] Knuth, D. E., *"Generating All Tuples and Permutations". The Art of Computer Programming.*, 4, Fascicle 2. Addison-Wesley, 2005, pp. 1–26

[90] Cosmin Marian Poteras, Mihai Mocanu and Constantin Petrişors, "A Distributed Design for Computational Steering with High Availability of Data", *International Journal Of Systems Applications, Engenering & Development, (NAUN),* Issue 1, Volume 6, 2012, pp. 52–61.

[91] Sever Alexandru HABA, Gheorghe OANCEA, Anisor NEDELCU, "Data Exchange Software Module between AutoCAD and CATIA Environments," in Proc. 11th WSEAS International Conference on Systems Theory And Scientific Computation (ISTASC '11), Florence, Italy, August 23-25, 2011, pp. 134–137.

[92] Howard Hinnant. (accessed: 2012, May 10). Stackoverflow C++ discussion. [Online]. Available:

http://stackoverflow.com/questions/2211915/combination-and-permutation-in-c

[93] Hervé Brönnimann. (accessed: 2012, May 10). *Algorithms for permutations and combinations, with and without repetitions*. [Online]. Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2008/n2639.pdf

[94] Howard Hinnant. (accessed: 2012, May 12). *Combinations and Permutations*. [Online]. Available: http://home.roadrunner.com/~hinnant/combinations.html

CURRICULUM VITAE

1. Personal data

Name: Dmitri Loginov
Date and place of birth: 02.07.1977, Mihailov, Russia
Citizenship: Estonia

2. Contact information

Address: Järveotsa tee 1-35, Tallinn, Estonia
Phone: +372 5226541
E-mail: dmitri.loginov@gmail.com

3. Education

Educational institution	Graduation year	Education (field of study/degree)
Tallinn University of Technology	2006	Environmental Engineering/MSc
Tallinn University of Technology	1999	Environmental Engineering/BSc
Tallinn Lasnamäe Gymnasium (secondary school № 12)	1995	Secondary education

4. Language competence/skills (fluent, advanced, intermediate, beginner)

Language	Level
Russian	Fluent
Estonian	Advanced
English	Advanced
French	Beginner
Swedish	Beginner

5. Professional Employment

Period	Organisation	Position
2012 - …	OÜ EMP A&I	CEO
2007 - 2012	OÜ EMP A&I	Project manager/IT manager
2001 - 2007	OÜ EMP A&I	HVAC engineer/LAN administrator
1999 - 2001	OÜ EMP A&I	HVAC engineer

6. Scientific work

Loginov, Dmitri (2012). On some technical issues and challenges in development and implementation of Autonomous Design System's (ADS) functionality. International Journal of Applied Mathematics and Informatics, 6(3), 142 - 151.

Loginov, D. (2012). Implementation of the Synergetic Computer Algorithm on AutoCAD Platform. *In: Recent Researches in Applied Information Science: 5th WSEAS World Congress on Applied Computing Conference (ACC '12), University of Algarve, Faro, Portugal, May 2-4, 2012.* (Toim.) Nikos Mastorakis, Valeri Mladenov, Zoran Bojkovic. WSEAS Press, 2012, (Recent Advances in Computer Engineering Series; 2), 112 - 117.

Loginov, Dmitri (2012). Advanced Synergetic Computer Algorithm for Vector Graphics Geometry Recognition and Its Realization as Part of HVAC ADS on AutoCAD Platform . *In: Advances in Mathematical and Computational Methods: 14th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering (MACMESE '12), Sliema, Malta, September 7-9, 2012.* (Toim.) Mihaiela Iliescu, Roman Prokop. WSEAS, 2012, (Mathematics and Computers in Science and Engineering Series), 121 - 126.

Loginov, D. (2011). Possibilities of Modeling the Creative Part of Engineering Design Process using the Synergetic Approach. INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES, 5(1), 95 - 104.

Loginov, D. (2010). Synergetic modelling – application possibilities in engineering design. *In: Selected topics in system science and simulation in engineering: 9th WSEAS International Conference on system science and Simulation in engineering (ICOSSSE '10), Iwate Prefectural University, Japan,*

October 4-6, 2010. (Toim.) Hamido Fujita, Jun Sasaki. WSEAS Press, 2010, 111 - 117.
7. Defended theses

Software agent-based CAD systems: application in HVAC field, Master's degree

VAV HVAC system of the chemical laboratory premises in Tallinn Waterworks, Bachelor's degree

8. Main areas of scientific work/Current research topics

Natural Sciences and Engineering, Construction and Municipal Engineering (Engineering design process theory, CAD in HVAC field, ADS, synergetics, artificial intelligence in design, self-organizing systems)

9. Other research projects

Autonomous Design Systems (ADS) in HVAC field.
Modeling of the creative part of the engineering design process.

ELULOOKIRJELDUS

1. Isikuandmed

Ees- ja perekonnanimi: Dmitri Loginov
Sünniaeg ja -koht: 02.07.1977, Mihailov, Venemaa
Kodakondsus: Eesti

2. Kontaktandmed

Aadress: Järveotsa tee 1-35, Tallinn, Estonia
Telefon: +372 5226541
E-posti aadress: dmitri.loginov@gmail.com

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Haridus (eriala/kraad)
Tallinna Tehnikaülikool	2006	Keskkonnatehnika/ Tehnikateaduste magister
Tallinna Tehnikaülikool	1999	Keskkonnatehnika/ Tehnikateaduste bakalaureus
Tallinna Lasnamäe Gümnaasium (keskkool № 12)	1995	Keskharidus

4. Keelteoskus (alg-, kesk- või kõrgtase)

Keel	Tase
Vene	Kõrgtase
Eesti	Kesktase
Inglise	Kesktase
Prantsuse	Algtase
Rootsi	Algtase

5. Teenistuskäik

Töötamise aeg	Tööandja nimetus	Ametikoht
2012 - …	OÜ EMP A&I	Tegevdirektor
2007 - 2012	OÜ EMP A&I	Projektijuht/IT juht
2001 - 2007	OÜ EMP A&I	KV insener/LAN administraator
1999 - 2001	OÜ EMP A&I	KV insener

6. Teadustegevus

Avaldatud artiklid ja konverentsiettekanded on ära toodud ingliskeelses elulookirjelduses.

7. Kaitstud lõputööd

Tarkvara agentidel baseeruvate masinprojekteerimise süsteemide kasutamise võimalused KV alal, magistritöö

Tallinna Veepuhastusjaama laborikompleksi muutuva õhuhulgaga ventilatsioonisüsteemi ja õhu konditsioneerimise lahendus, bakalaureusetöö

8. Teadustöö põhisuunad

Loodusteadused ja tehnika, Ehitus- ja kommunaaltehnika (Projekteerimise teooria. Masinprojekteerimise süsteemid, nende rakendamine kütte- ja ventilatsioonivaldkonnas. Tehisintellekti kasutamine masinprojekteerimises. Autonoomsed masinprojekteerimise süsteemid. Iseorganiseeruvad süsteemid.)

9. Teised uurimisprojektid

Autonoomsed masinprojekteerimissüsteemid (ADS) kütte- ja ventilatsioonivaldkonnas.
Insenerliku projekteerimisprotsessi loomingulise osa modelleerimine.

PAPER I

Loginov, D. (2011). Possibilities of modeling the creative part of engineering design process using the synergetic approach. *International Journal of Mathematical Models and Methods in Applied Sciences. Issue 1, Volume 5, 95–104.*

Possibilities of modeling the creative part of engineering design process using the synergetic approach

Dmitri Loginov

Abstract**—**In this paper the problem of modeling the creative part of the engineering design process has been analyzed from the synergetics perspective. The analysis possessed from a general point of view. The characteristics of the creative tasks of the engineering design have been defined and novel notion of the Autonomous Design System (ADS) has been introduced. ADS is considered as an advanced CAD system that has Artificial Intelligence (AI) functionality and particularly the functionality to deal with creative components of the engineering design process. A couple of cybernetic models which may be further optimized by the methods of synergetics were proposed. The presented discussion forms theoretical foundations and philosophical motivation for an ongoing research in this field. This work constitutes the introduction to the extension of the author's original research in the field of CAD systems' optimization.

Keywords**—** Self-organization, Engineering design, Synergetics, Artificial intelligence, Complex systems.

I. INTRODUCTION

ENGINEERING design can be viewed as an articulate process composed of phases, where each phase represents a combinatorial action on the parts the composite object is consisted of. To realize an object meeting the desired market requirements, engineering designers have to deal at the same time with different kinds of knowledge about objects or, ontological knowledge (which is often represented in a declarative form), and "dynamic" knowledge about processes (which is often represented in "procedural terms") [1].

Synergetics can be considered as one of the modern, most promising research programs. It is oriented towards the search for common patterns of evolution and self-organization of complex systems of any kind, regardless of the concrete nature of their elements or subsystems.

These parts of design process that are numerically analyzable could be modeled numerically. The numeric model could then be further improved and optimized. We can use all the benefits and achievements of the digital revolution, including Artificial Intelligence (AI), to automate the process of engineering design.

A. Creative part of engineering design

On the other hand, the design process consists of the "creative" part that is not numerically describable by traditional numerical algorithms, at least not yet. As today's traditional CAD (Computer Aided Design) systems are based on numerical (digital) computational machines (i.e. personal computers), there are no ready standard solutions to automate these creative parts of the design process. Note that "routine" parts of the engineering design process could be modeled, divided in parts and automated relatively simply, and there are a lot of examples on the market today.

On the contrary, there is only one stage in the true model of the creative process. At the simplest level, creativity is the act of being and doing folded into a state of flow called life. We naturally spend all of our time in a state of flow, despite claims in the popular press to the contrary. Even when we are analyzing a problem, we are *doing* something, and employing tools of some sort. We simultaneously embrace a rapidly evolving picture of what we want to do that unfolds just before we do it. What the popular press describes as a state of flow occurs when the execution of the creative process becomes jubilant, and consequently high performance.

We divide the creative process into pieces in an effort to understand and picture the complexity of the entire process. But let us not fall into the trap of believing that we actually execute the pieces in some sort of lock step fashion. It is convenient and instructive to perceive that creativity has certain stages and that we can all emotionally, physically and mentally relate to these stages, but to hold any model of the creative process as a precise description of creativity, and to force others to adhere strictly to its application is foolish. Stuart Kauffman uses an expression to describe the difficulty of modeling any living system: "the algorithm is incompressible." In other words, there is no shorter method, routine or program to describe life or living systems than life or the living system itself. Models are representations of reality but they are not the reality itself. There is no algorithm or equation that we can force creativity into that is shorter than the creative act itself [2]. However, there is a hope that we can still approximate the model to its representation by introducing (and using) the emergent AI technology, its tools and algorithms.

B. Synergetics and self-organization

Synergetics (Greek: "working together") is an interdisciplinary field of research originated by Hermann Haken in 1969. Synergetics deals with material or immaterial systems, composed of, in general, many individual parts. It focuses its attention on the spontaneous, i.e. self-organized emergence of new qualities which may be structures, processes or functions. The basic question dealt with by Synergetics is: are there general principles of self-organization irrespective of the nature of the individual parts of a system? In spite of the great variety of the individual parts, which may be atoms, molecules, neurons (nerve cells), up to individuals in a society, this question could be answered in the positive for large classes of systems, provided attention is focused on qualitative changes on macroscopic scales. Here "macroscopic scales" means spatial and temporal scales that are large compared to those of the elements. "Working together" may take place between parts of a system, between systems or even between scientific disciplines. See e.g. [3], [4] for further reading.

A simple example of self-organization is a case of a fluid heated from below which may form patterns in the form of hexagons or rolls based in an upwelling of the fluid (Fig. 1).

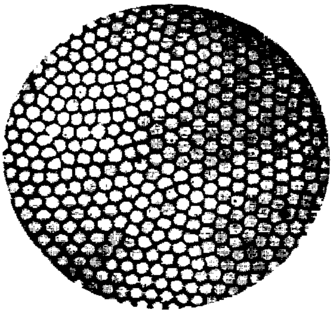


Fig. 1 Top view of a liquid in a circular vessel heated from below

When the temperature gradient exceeds a critical value, hexagonal cells are formed. In the middle of each cell the liquid rises, sinking back down to the edges of the hexagon. Further examples are provided in physics by the production of coherent light of lasers, in chemistry by the macroscopic rings or spirals formed in chemical reactions and in biology by morphogenesis during the grows of plants and animals.

The sunflower head, for instance, is composed of two counter rotating spirals, which must hit under a quite specific angle (Fig. 2).

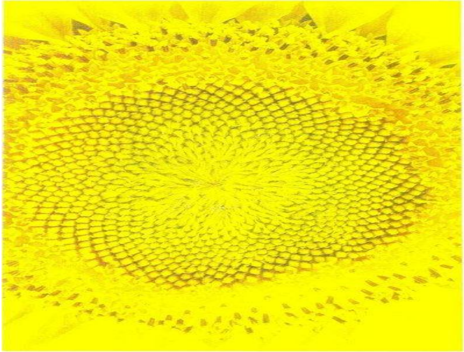


Fig. 2 The sunflower head is composed of two counter rotating spirals

The reason for this is not yet fully understood. Another example is in behavioral patterns which may range from the gaits of horses to the specific movements of human beings. In all of these cases the systems acquire their structures as a result of rather unspecific changes of their environment, for instance when the fluid is heated more strongly or when the concentration of a chemical is changed and so on. In other words, the structures evolving in the system are not prescribed in a specific manner from the outside. More precisely, the system forms its new structure by self-organization.

II. PROBLEM FORMULATION AND ANALYSIS

The creative parts of the design process are characterized by the higher intelligence needed to deal with them. Therefore, if we want to model that part of engineering design we need far more powerful AI technologies then those existing today.

It could not be even possible to model/automate these creative components by means of today’s computers. Maybe some new revolutionary technology is needed to do that. However, there is a belief that some of these components still may be approximated (to some degree) by mathematical methods that are readily available now (by their improvement) or by the newest methods that have emerged recently, or those, still under development at the moment. The AI tools and technologies in cooperation with synergetics, for instance, may help us to achieve that goal.

Synergetics (in the meaning of H. Haken’s school of thought) provides mathematical tools to cope with the self-organization phenomenon. These tools are based on the combination of the differential equations theory and the stochastic modeling. The principles of synergetics can be used in a great variety of scientific disciplines ranging from the theoretical physics to musical [5] and social sciences. Let us look first at the synergetics from the general, epistemological point of view.

Synergetics reflects the surrounding natural systems in a sense of soft or coherent action principle. The natural phenomena develop along the evolutionary paths according to evolutionary principles. There are no hard or external actions which can successfully drive (manage) the complex natural system in its development pathway. From that we may learn how to arrange our activities in order to achieve optimal results. It turns out that managing influence must not be energetic, but rightly topologically organized according to the general and universal laws of self-organization. There must be certain organization of actions. It is the topological configuration, the symmetric "architecture" that is important, not the intensity of the influence. Synergetics defines how it is possible to multiply reduce time and required efforts to generate, by a resonant influence, the desirable and, what is no less important, feasible structures in a complex system. These principles are equally applicable to the case of modeling the complex parts of engineering design process.

Leo Näpinen, for example, stresses the importance of participative constructive activity as follows [6]. "The cosmos is filled with the creativity of the process of endless transformations, and human creativity derives from the creativity of the cosmos itself. Human constructive activity is justified indeed — but not a dominative construction. Instead, it has to be participative construction."

A. Overview of the related research

The question is not whether we could model the creative component on today's computers using common, traditional computational methods, because the simple and obvious answer to this is that we could not. The question is rather could we invent some new computational methods that allow to model the creative components of the design process. According to [7] it is possible by investigating the social-biological functionality of the human species. The authors note that deeper computational studies of biological and cultural phenomena are affecting our understanding of many aspects of computing itself and are altering the way in which we perceive computing proper.

The authors propose to model human intelligence by modeling individuals in a social context, interacting with each other. The important point is that while interacting they have to change their thinking process not just the content. The authors oppose that to the software agents' systems, which, according to them, are only capable of exchanging information while their own state persists unalterable. We argue here against the view of software agents' paradigm, since agents may have dynamic structures, which are capable of learning and improving their functionality over time (they could be linked to artificial neural network, for instance, or to other AI systems [8], they may be a part of Complex Adaptive System (CAS), as described in [9]). The social behavior greatly increases the ability of organisms to adapt. Minds arise from interactions with other minds.

The interesting insight into the problem gives Jorma Tuomaala [10], who considers creativity as the product of humans' subconscious mind and the intuition. He then proposes the process of formalization of the creativity based on intuition and combination of conscious and subconscious mind. The author is not naming that explicitly self-organization, but it is obvious that his methods of "capturing" creativity are tightly bound up with the notion of self-organization and synergetics (theory of self-organization).

In his book he attempts to handle the intuitive process and the possibilities of its controlled use based on his own experiences as a mechanical engineering designer. He examines in what ways one might use one's subconscious and conscious mind together in engineer's design work. He notes that this study is completely fictitious as far as subconscious mind is concerned. Nevertheless, he has tried to develop a model of it that may increase the understanding of the function and may even be useful in practice. He has also tried to expand the discussion to the area of literature and performing art, because he sees engineering design, as well, as a deeply human activity giving one all the possibilities to grow as a person (such approach is also articulated in self-organization paradigm where someone creates arts i.e. "finds out" order from chaos). He has tried, also, to build a generally applicable method of intuitive creative work. According to J. Tuomaala creativity is needed in the creation of everything new. He explains that the ideas produced intuitively seem to be, despite all arguments, usually of the best quality. This arouses the question: should one mainly seek to use intuition along with the systematics in (mechanical) engineering and

a) Can one then still define the time needed to find solution?

b) Can one consider the intuitive solution as an optimum?

c) Does the time used for solving the problem affect the quality?

d) Is it possible to lead the intuitive process?

And at last, we may ask the question, if the intuition seems to be a useful instrument in creative engineering design, is it possible to model it using some computational algorithm?

Another attempt to formalize the creative part of the design process was done by MG Taylor Corporation and brought/explained by Bryan Coffman in 1996, see [2]. Very interestingly, MG Taylor Corporation seems to deal with creative design formalization since 1982, but even today there are no working examples of an autonomous engineering design system created (i.e. system that is capable of modeling the creative part as well), although there were attempts to create a conceptual framework and general methodology to aid the design of complex adaptive systems using the principles of self-organization [9].

The ongoing research in AI domain and in the field of general technology shows that traditional methods of solving engineering problems based on formal logic and systematical approach shifts toward the new unrevealed, presently undocumented features of human mind and intelligence (more closely to the characteristics of self-organization?). There are neural networks, which try to copy the functionality of biological brain cells – neurons, fuzzy logic and modeling (for a contemporary research on fuzzy dynamic systems see e.g. [11]), expert systems, evolutionary programming/computing, knowledge-based systems, swarm and genetic algorithms and so on. In that sense we can compare this to the paradigm shifts that occurred in 20th century when the new age science transformed from its classical period (Galileo-Newton physics) out to non-classical (quantum mechanics, static laws and systems) and post non-classical (open non-linear systems etc.) forms.

B. Synergetic modeling

The routine parts (numerically describable) of the engineering design process could be successfully modeled with the help of cybernetics. It is really the art of combinatorial manipulation and constructing (constructive rationality) to fulfill the goal, using the already known or novice technology, IT in this case. As it is based on cybernetics, it falls down to organizational theories, contrary to self-organization paradigm, and therefore is not the subject of interest of this paper.

Let us take a look at the notions of organization and self-organization from the concept point of view. The concept of organization denotes the process that leads to the rise of goal-oriented structures due to conscious human goal-directed action or some external ordering influence, and the concept of self-organization would denote the process that leads to the rise of goal-oriented structures beyond conscious human goal directed action or some external ordering influence. Although the term “self-organization” is widely used (and more appropriate) in the field of synergetics, it has been utilized in cybernetics as well. In cybernetics, however, it has different meaning (from the philosophical point of view). In cybernetics and systems engineering self-organization is understood as an effect of an external ordering factor (e.g. self-organizing map in [12]). In synergetics self-organization is understood as the rise of harmonious behavior distinguished from man's intervention and from external (with regard to the system) ordering factors. External factors (e.g. strong non-equilibrium) are indispensable for self-organization, but only as conditions, not as ordering forces.

Hopefully, it is possible to imitate the creativity (at least to some degree) by means of synergetic modeling. Could we model the creative part of the (engineering) design process as well? To answer this question we must analyze the synergetic approach and compare it with traditional information technology modeling instruments e.g. with cybernetics.

In cybernetics as well as in synergetics the objective processes are modeled in order to control them. The cybernetic models make it possible for man to strive for the desirable results using the program created by him. The synergetic models take into account that the programs form in the course of self-organization [13].

All exact sciences (and also the traditional scientific cognition) are model-based. They are exact only within that model. Therefore it is not possible to explore/predict/study adequately the real world by means of “exact” sciences by definition. We can use exact sciences to explore models. CAD systems, ADS (Autonomous Design System, ADS is considered to be a further development of a conventional CAD system, which takes into account the creative component of the design process) frameworks are examples of design systems’ models. Both cybernetics and synergetics are exact sciences as well. So we can use these disciplines only for the development and research of models of the underlying real world’s phenomena and not for the investigation of the real world itself. It must be underlined that in exact sciences the approach to the interaction between organization (management) and self-organization does not go (and due to the specificity of exact sciences must not go) farther from certain boundaries.

The limits mean that exact sciences in their models of influence upon self-organization give only such recommendations according to which the future state of an object of management is given from the outside. Exact sciences do not make any contribution to the opening of the creative potential of the elements of the system [13]. So we cannot use standalone synergetic methods (a kind of exact science) to explore the creative potential of the system (and self-organization). As the synergetics is exact science and is based on mathematics, it has known limitations in its capability to explore the real world. But still we can use it to create the better models of the real life systems, not to understand these systems completely. On the other hand, building more adequate models of the environment leads to a better understanding of the environment itself. And, therefore, may lead us to a new level of understanding, to help us form a new paradigm and from within it - to model even more precisely, closely to the real world.

Synergetics better than cybernetics models the processes of the real world which is ultimately the self-organizing system. So we can use principles of synergetics in conjunction with traditional computing technology to model some aspects of the real systems. It is worth showing how creativity is understood in synergetics. The meaning of the word creative is the unpredictability and unavoidability of the unknown. The creative chaos is the field of unknown and unpredictable chances. The meaning of the word is closely related to such concepts as non-equilibrium condition and conditions close to equilibrium.

Synergetics accentuates also one necessary condition of self-organization: the order arises from chaos only under the condition of *strong non-equilibrium*. It is necessary to distinguish strictly chaos under the conditions close to equilibrium (in which, generally speaking, self-organized structures can only decompose) from chaos under the strongly non-equilibrium conditions (in which composing of structures through self-organization can take place) [13]. The former type of chaos is non-creative, the latter is creative.

In engineering design process theory the meaning of the words “creative” (and “creativity”) is slightly different. Here the word “creative” denotes a non-routine part of the design process. Contrary to the routine procedures where inputs and outputs of the system are known or predictable, the creative part of the process deals with output data that is mainly unknown, although the field of possibilities (possible outputs) is generally defined. This is true in ordinary design scenarios where the ultimate goal of the design procedure is known.

When the output data of the system is completely undefined and unknown, then we are dealing with the system that generates some new design information (i.e. invention mechanism). Note, that the input data in majority of cases is defined (both in ordinary design scenarios and in invention apparatus). The modeling of the technical invention processes is even more complicated (if not impossible) than imitating the creative part of the conventional design process (i.e. the process where the field of the possibilities of the output information is defined). There is a hope that using the methods of synergetics and the philosophy of self-organizing systems we can try to address the problems of modeling creative design in a more precise and better manner. The new science which accepts creativity based on chance and irreversibility in nature, and considers the fundamental indeterminacy of the whole history of nature and of human society should evolve to acknowledge the potential of this approach.

Basically, we can consider a model as an idealized version of the real system. The model is always a simpler and more primitive than the real system. The traditional tool for creating engineering design models nowadays is a Computer Aided Design (CAD) system. For a creation of a new CAD system we use CAD programming. Thus, CAD programming is essentially construction of the model (computer program) for the model (CAD application) of a model (engineering design, project) of the system (e.g. engineering installation). Such models' cascading occurs e.g. in a case when we are programming under some existing CAD platform, let's say under AutoCAD®. On this level of abstraction the model itself is very precise (it is nested into surrounding model etc.) and perfectly describable by mathematics.

The aim is to try to add to this model the properties/specifications of the self-organizing systems' behaviors. The author does not really think that the model will be capable of substituting the engineer completely in the process of producing creative design. But there is a hope that the model built in the spirit of synergetics could facilitate the emergence of the elements of the creativity in engineering design in which the human participates as well. It is likely that these models in cooperation with the operator (engineer) can function more effectively in creating new designs. Moreover, the engineer and the model in conjunction both virtually constitute a self-organizing system and the number of degrees of freedom of that resulted system is bigger than in each of its separate part. Thus, the probability of emergence of interesting and usable design scenarios is larger. It must be stressed here that we do not know whether the useful design cases ever emerge as a result of using the synergetic model. It is impossible to specify when and what kind of outputs from the model will be created. This would be a kind of system with a rather probabilistic behavior, therefore, in theory it could even downgrade (to some degree) the developments of the design process, but, nevertheless, even in that case it still will be operating according to the principles of self-organization. And maybe, who knows, the wrong output (as it seems at present time) will be considered over time quite a better one. The big question is how to compound such a system that it could be, so to say, maximum self-organizing, because we still have to construct it, i.e. the system does not emerge as a result of self-organization in principle. Maybe the wiser behavior would be the passive one – not to construct, but be inactive, wait till the systems will arise by themselves? Or just create some very simple systems with minimum "dominative construction" attributes and let the general outer self-organizing world finish the model according to its intrinsic implicit laws (as we know from synergetics, it seems that just the simpler laws drive complicated phenomena)?

Here is a quote from K.Popper's book "All Life is Problem Solving" [14], where he emphasizes the value of free opinion's formation, which in conjunction with simple clear language (analogy to simple "laws" that drive complex phenomena?) may be considered as characteristics of self-organization (in social context). "Why does simplicity of language matter so much to Enlightenment thinkers? Because the true Enlightenment thinker, the true rationalist, never wants to talk anyone into anything. No, he does not even want to convince: all the time he is aware that he may be wrong. Above all, he values the intellectual independence of others too highly to want to convince them of important matters. He would much rather invite contradiction, preferably in the form of rational disciplined criticism. He seeks not to convince but to arouse— to challenge others to form free opinions. Free opinion formation is precious to him: not only because this brings us all closer to the truth, but also because he respects free opinion formation as such. He respects it even when he considers the opinion so formed to be fundamentally wrong. "

It is possible to possess experiments with some candidate synergetic models in order to select the more appropriate one (we must remember, however, that in experimental situations, the determinants of the organization or process are contingent on the subject i.e. the experimenter; for a detailed description of the philosophical interpretation of humans' constructive activity (based on Aristotle's four causes that in unity form the philosophical category of self-organization) see e.g. [15]). But again, what are the criteria for the selection and are these criteria adequate enough? Self-organization is impossible to describe adequately in details, so how it is possible to define adequately its characteristics for the selection criteria?

III. PROBLEM SOLUTION

In this section a short overview of the cybernetic models, which are suitable to combine with synergetic methods, is given. Note that these models are novice and they are under research right now. Initially they were intended to use as a standalone frameworks for a creative components of the Autonomous Design System (ADS). ADS is defined as an advance CAD system, which has AI functionality and *particularly the functionality to solve the creative tasks* of the engineering design process. ADS is opposed to the conventional CAD systems, (see e.g. [16]) which normally automate routine parts of the design process and generally have no AI capabilities.

A. General remarks

The properties of self-organizing systems in general could be discovered (if ever) using the methods of historical cognition (although in relatively small isolated groups/systems it is possible to use methods of classical exact sciences for that purpose). These methods are closely connected with the notion of time. The analysis of the historical phenomena is not possible without any knowledge on their past, as they develop in the process of irreversible evolution. Thus the future is unknown and unpredictable.

From the philosophical point of view it seems that the hypothetical-deductive knowledge in principle cannot explain historical phenomena. But it is possible to use the hypothetical-deductive theories of exact sciences in the modeling of some aspects of historical phenomena. It must be regarded as the cooperation of reconstructive (historical, descriptive-theoretical) and constructive (hypothetical-deductive) approaches but not as an attempt to replace one by the other [13]. It is also possible to use the reconstructive approach in modeling some parts of the cybernetic systems. In this case we can call this cooperation as well. From the perspective of modeling of the creative part of the engineering design process this type of cooperation seems to be more appropriate. We have to try to introduce the irreversible "arrow of time" (i.e. the reconstructive approach) into cybernetic models (e.g. by cooperation techniques; in this case it will be a rather synergetic model; this approach maybe needs a different computational algorithms/paradigm not invented yet) or, at least, to try to connect the computational system based on cybernetics with synergetic model. This should improve the initial models' architecture, their functionality and bring these models closer to the real life phenomena. The question is whether we could model the synergetic system that has characteristics of irreversible "arrow of time" by means of traditional computer (i.e. serial computational machine), which is based on reversibility principles.

As I. Prigogine showed in the language of mathematized science, the real situations are orientated in time, the states and the laws are closely connected with one another and that the initial conditions of the system emerge as the result of its previous evolution [13]. For a critical analysis of I. Prigogine's attempts to use exact science (based on mathematics) for understanding the natural-historical phenomena see e.g. [17]).

I. Prigogine has written that irreversibility can no longer be identified with a mere appearance that would disappear if we had perfect knowledge. Instead, it leads to coherence, to effects that encompass billions and billions of particles. He noted that we are actually the children of the arrow of time, of evolution, not its progenitors.

It seems that most of self-organizing systems have their inner goal, towards which they are constantly evolving, but to make this goal clear for us, humans, it is very hard (if not impossible) task. The essential characteristic of a self-organizing system is its autonomous purposive behavior. The characteristics of a self-organizing system cannot be constructed according to an external purpose (from the outside, regarding to the system). "Self-organizing systems … have their own (i.e. autonomous) goals…" [18].

If we get a successfully constructed synergetic model, which operates according to internal purpose, we will get a kind of a self-organizing system. Then we can introduce some external agent that performs reasoning upon the internal characteristics of the system in order to construct external conclusion or view, thus achieving an external purpose (interpreting the internal information). By doing that we could get a compound self-organizing system (i.e. synergetic model), which has an external goal. And this is relatively straight forward activity to construct such a system, since all (traditional) cybernetic and systems' engineering models function according to the external purpose. Maybe, in such a fashion, we can use the benefits of self-organizing systems in CAD applications and in ADS.

B. Modeling (reasoning) by analogy

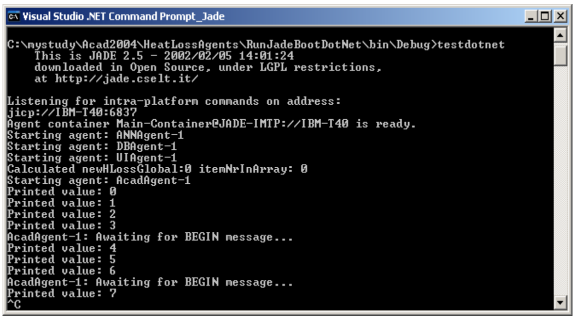
This activity conforms to the evolutionary theory of systems development (including e.g. culture, human society). It happens that the human brain functions largely in the same way. Also the "learning" processes of the majority of biological species base on the principle of analogy. The idea is to try to create a model of such a learning system. The model could, in general, function as follows. Some cases/situations are presented to the system from which it may learn, i.e. acquire some information. The system (model) remembers this information and then in the future it may be capable not only of finding the exact learned cases but also the analogical cases. To accomplish this, the system must have some reasoning mechanism that allows recognizing analogies in the presented/surveyed information. In order to improve the model (in the sense of self-organization) we can add here the "historical component". The system remembers the case in the historical context, in real time; with the characteristics of the environment such as time, the source of the information etc. It is the system that takes into consideration the initial conditions of the process of information acquisition. It is important to underline that these initial conditions are not arbitrary as in conventional (classical) cybernetic models. It also could be possible to put the system under conditions of strong non-equilibrium in order to stimulate the emergence of the creativity. In such a way we could get a synergetic model that, in addition, functions similar to the majority of biological systems on Earth, including humans and the functioning of human's brain (in creativity context?).

C. Dreams' modeling

The idea of human dreams' modeling comes from the fact that sometimes the products of creativity arise during the sleep, when dreaming. Although the dreaming mechanism is not known well yet, it is possible to model it at the most primitive level of abstraction. Namely, it is suggested that dreams are composed of previously acquired information, of the interpretations of previous experiences and of the combinations of this data. The exact mechanism of the combining process is still unknown, but at the most simple level it is asserted to be a random combinatorial activity. In that case it is possible to model that combining process by means of traditional computing (IT). Assumption: the dream may consist of the entities previously known by the system. We need to create the algorithm of combining these entities (possibly fuzzy, random), and the mechanism of interpretation of these combinations. Some of the combinations may be useful in system's work. So we can state that dreams' modeling is a kind of combinatorial (random) activity on some known information segments, which, hopefully, may lead to the useful combinations of data that could be considered as a product of creativity. Again it is possible to add to the initial model the properties of the self-organizing system to optimize it and to improve system's performance. We can also use principles of synergetics in combination with other non-classical branches of science, for example with memetics. In [19] such an approach was used to explain and to create the alternative theoretical base of a given musical system. Memetics is a theory of mental content based on an analogy with Darwinian evolution. It purports to be an approach to evolutionary models of cultural information transfer. A meme, analogous to a gene, is an idea, belief, pattern of behavior (etc.) which is "hosted" in one or more individual minds, and which can reproduce itself from mind to mind. Thus what would otherwise be regarded as one individual influencing another to adopt a belief is seen memetically as a meme reproducing itself.

D. *Software agents*

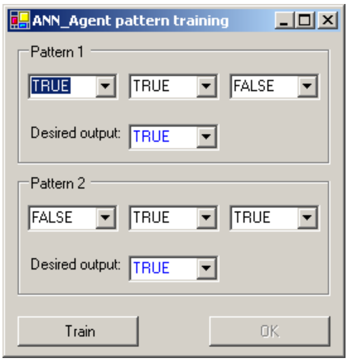
Another AI technology that may be further improved using the synergetic approach is autonomous software agents. With this type of computational model it is suitable to model the behaviors of social systems. As a self-organization of society is connected with freedom of individuals, we can use a system of relatively independent (free) software agents to model a self-organizing system. The software agents' technology may be successfully combined with other AI technologies (neural networks, genetic algorithms e.g. [20] etc.) in order to improve a system even more. In [8] for example, a suite of independent software agents was developed that run on AutoCAD® platform. Software agents were connected with Artificial Neural Network (ANN) in the form of a separate agent that has ANN functionality and that is capable of being trained in a particular way. See Fig. 1 and Fig. 2 for an example of some visual interfaces from the system implemented. (For an overview of the software agents' technology and its hybridization possibilities see [8].)



```
C:\mystudy\Acad2004\HeatLossAgents\RunJadeBootDotNet\bin\Debug>testdotnet
    This is JADE 2.5 - 2002/02/05 14:01:24
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.cselt.it/

Listening for intra-platform commands on address:
jicp://IBM-T40:6837
Agent container Main-Container@JADE-IMTP://IBM-T40 is ready.
Starting agent: ANNAgent-1
Starting agent: DBAgent-1
Starting agent: UIAgent-1
Calculated newHLossGlobal:0 itemNrInArray: 0
Starting agent: AcadAgent-1
Printed value: 0
Printed value: 1
Printed value: 2
Printed value: 3
AcadAgent-1: Awaiting for BEGIN message...
Printed value: 4
Printed value: 5
Printed value: 6
AcadAgent-1: Awaiting for BEGIN message...
Printed value: 7
^C
```

Fig. 1 Agent system waiting for BEGIN message



Pattern 1
TRUE TRUE FALSE
Desired output: TRUE

Pattern 2
FALSE TRUE TRUE
Desired output: TRUE

Fig. 2 ANN_Agent's user interface form

We then should include into the resulting system the characteristics of the self-organization, mentioned above, to get a candidate for a successful synergetic model.

IV. CONCLUSION

A. *Philosophical outlook*

In this section some concluding philosophical remarks on the subject are brought.

Stephen J. Gould in his works (e.g. [21]) stresses the importance of the historical character of the life. As we remember, different life forms may be considered as an example of self-organizing systems. The important property of self-organizing systems is their historical character of evolution. In his definitions he also notes that the human species are not necessarily the highest expression of the life on our planet and therefore, to put it objectively, we should not put ourselves into belief that our cognition mechanism might be the supreme and the right one. He notes that in order to understand the events and generalities of life's pathway, we must go beyond principles of evolutionary theory to a paleontological examination of the contingent pattern of life's history on our planet which is the *single actualized* version among millions of plausible alternatives that happened not to occur. Such a view of life's history is highly contrary both to conventional deterministic models of Western science and to the deepest social traditions and psychological hopes of Western cultures for a history culminating in humans as life's highest expression and intended planetary steward. Stuart A. Kauffman [22], [23], one of the leading figures in the study of self-organization and complexity nowadays has pointed out that the evolution of the whole world appears to be a combination of selection and self-organization. Thus the understanding of evolution only by natural selection is incomplete.

If we assume that the self-organization paradigm is true, then human mind, among all other known systems in our world, must function according to it as well. From that we can conclude that the human mind's creativity has to work in accordance with synergetic models as follows. In synergetics, models should ideally reflect the transitions between different qualitative states by *positive feedback* (a system exhibiting positive feedback, in response to perturbation, acts to increase the magnitude of the perturbation). These transitions are possible only if the influence of external environment on the system is so changeable that the amplification of the fluctuations may cause the system to move away from the equilibrium so far that it cannot return to the former state and there may appear new possibilities of development (i.e. creativity?).

These qualitative transitions are simultaneously both determined and undetermined. The fundamental objective indetermination lies in their basis. It is not determined into which qualitative state from some (or many) possibilities the system really goes after the selection. But the field of possibilities is determined. [9] These model' characteristics should comply with the theory of self-organization or synergetics that was introduced and thoroughly developed by Hermann Haken.

We also have to remember (while constructing, modeling some natural phenomena) that the scientist himself and his activity together with its products can now be treated as part of modeling process, of nature. Nature is understandable as a living being who, thanks to the conceptual and technical idealization, is indeed predictable, even transformable and manipulable, but only locally, partially and relatively [24].

As Zwierlein notes [25], one cannot approach any question at all from a neutral or objective standpoint. Every questioning grows out of a tradition and its underlying pre-understanding that opens the space of possible answers. To grow and to expand the horizons does not mean to surpass the condition of having a background of pre-understanding in principle. We will always operate within the framework the “Lebenswelt” provides for us. And it is definitely impossible that our understanding will ever be neutral or objective or complete.

We agree with Zwierlein at that point, but does it mean that mankind will never be capable to understand the surrounding integral world completely? Should we try to approach the problem of understanding from another perspective that is not based on traditional cognition (i.e. logical, cumulative way of thinking) or as Zwierlein and Kant note, is ultimately rooted in anthropology (scientific understanding)? For instance, one can try Taoist approach (meditative practices etc), but, although, the so-called enlightened adepts are reputed to realize (specifically, to realize not to understand – in Taoist practices these words have different meaning) the true meaning of life and integral world (Lebenswelt?), is this realization the true one? How one can check this against the truth and what is the truth itself? Moreover, every Zen master, for example, still has got slightly different realization (understanding) of things, although they insist that the core understanding is the same and the one. All these points just amplify doubts and uncertainties about humans' cognition mechanisms.

Hayek [26] suggests that while known biological species have adapted with fixed and rather limited environmental “niches” beyond which they cannot exist, human and some animals, rats, for instance, were capable to adapt almost everywhere on the planet. Hardly this was achieved simply by the individual adaptability alone. The author points out that the overall success of the human species our civilization owes to the social ability and cooperation of the individuals (which derives from self-organization?).

Let us conclude this section with a rather philosophical outlook on the problem. What would be the directions for further evolution of the mankind? Some people think that the acceleration of the development of the technocracy (in the meaning of hypothetic-constructive-deductive way of cognition, opposed to self-organization thinking) is an answer. And we can see that this way of thinking really dominates in today's scientific and domestic (every-day) domains. The alternative understanding, on the contrary, respects the appearances of Mother Nature in all aspects of reality and promotes the soft management and participative construction in accordance with the global self-organization. As Leo Näpinen notes [27], the big question is whether the human mind (spirit) will grow belonging to the general determinacy (comprising the creative chance) derived from the integrity of the world —, i.e. belonging to the self-organization, or the human mind will restrict itself to the splitting of the integral world into pieces and manipulating with them — i.e. will restrict itself to organizing the organizations.

B. Artificial creativity

Going back to creativity modeling, we should take into consideration that in order to model human' creative process we can use the analogy principle, for instance, only to some degree. This means that we do not know exactly how this process occurs in reality (e.g. dreams, emotions etc). We can rely only on some possibly true facts (knowledge) that today science has about it. Therefore, as a result, we can get only an approximation of the real system (artificial human creativity). Another point is that we are not really interested in examination of how this creativity really works (i.e. the objective of the research is not to ultimately expose the mechanism of creativity but to build the mathematical/synergetic model that is relatively creative (mathematical model's "creativeness" – in a sense of engineering design creativity – see above)), instead we want to model this phenomenon and use it in practical applications, which may help us do better design (engineering) work and automate engineering design process. On the other hand, in order to model the system successfully, it is useful to know how the real system works, at least on the conceptual level.

In addition, in all of these implementation examples, in synergetic models it is possible to use the principle of new mereology - the philosophical study of wholes and parts, which states that in dissipative structures (i.e. self-organizing systems) parts are modified by their composition into a whole. The existing versions of mereology rely on the assumption that parts are not changed by being associated into wholes. To put it simple, the sum of the single components' properties is not equal to the compound property (as a whole) of these components (in qualitative sense). In synergetic models combining parts of the e.g. information (composing into a whole) may lead to the emergence of new properties of the resulted compound system.

There is a need for a further and better research of the phenomenon of self-organization in the natural systems and in the synergetic models. Future developments in the science of self-organization are likely to focus on more complex computer simulations and mathematical methods. However, the basic mechanisms underlying self-organization in nature are still far from clear, and the different approaches need to be better integrated [28].

Learning the principles of self-organization, however, is not a simple task and needs careful and thinking approaches. While approaching this conception we must remember that our species (humankind) is not necessarily the life's utmost creation on this planet nor our understanding of the surrounding world is unconditionally adequate.

REFERENCES

[1] G. Colonro, A. Mosca, F. Sartori, "Towards the Design of Intelligent CAD Systems: An Ontological Approach," *Advanced Engineering Informatics,* Vol.21, No.2, 2007.

[2] B. Coffman. (December18, 1996). *Anatomy of the Creative Process*, MG TaylorCorporation,.[Online].Available: http://www.mgtaylor.com/mgtaylor/jotm/fall96/7stanat.htm

[3] H. Haken, H. Knyazeva, "Arbitrariness in nature: synergetics and evolutionary laws of prohibition," *Journal for General Philosophy of Science*, No.31, 2000, pp. 57–73.

[4] H. Haken, *Synergetics. Introduction and advanced topics*, Springer, 2004.

[5] Martináková-Rendeková Z., "Systems theoretical modelling in musicology," In: Mastorakis, N. (ed.), *Mathematics and Computers in Modern Science. Acoustics and Music, Biology and Chemistry, Business and Economics*, World Scientific and Engineering Society (WSEAS) Press, 2000, pp. 122–127.

[6] Leo Näpinen, "Understanding of the World and the Scientific Paradigm of Self-Organization", *Studia Philosophica*, Vol.IV, No.40, 2004, pp. 156–177.

[7] J. Kennedy, R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[8] D. Loginov, "Software Agent-Based CAD Systems: Application in HVAC Field," Master's Thesis, Dept. Environmental Eng., Tallinn University of Technology, 2006.

[9] Radu Dobrescu, Matei Dobrescu, Dan Popescu, "Using self-organized living structures principles in the design of Complex Adaptive Systems," in Proc. 4th WSEAS International Conference on Communications and Information Technology (CIT '10), Corfu Island, Greece, 2010, pp. 155–160.

[10] Jorma Tuomaala, *Creative Engineering Design*, University of Oulu P.O.B 191 FIN-90101 Oulu Finland, 1999.

[11] Nikos Mastorakis, Olga Avramenko, "Fuzzy models of the dynamic systems for evolution of populations," in Proc. 3rd WSEAS International Conference on Mathematical Biology and Ecology, Gold Coast, Queensland, Australia, 2007, pp. 27–32.

[12] Supakit Siripanadorn, Wipawee Hattagam, Neung Teaumroong, "Anomaly detection using self-organizing map and wavelets in wireless sensor networks," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 291–297.

[13] Leo Näpinen, "Philosophical Foundations of Synergetic Modelling," *Proceedings of the Estonian Academy of Sciences. Humanities and Social Sciences*, Vol.4, No.42, 1993, pp. 378–390.

[14] Karl Popper, *All Life is Problem Solving*, Routledge Books, 1999, pp. 85–86.

[15] Вихалемм, Р, "К вопросу о концептуальных системах химии (Conceptual systems in chemistry)", *А. А. Печенкин (Отв. ред.). История науки в философском контексте. Посвящается памяти Владимира Ивановича Кузнецова (1915–2005) СПб: РХГА*, 2007, pp. 49–70.

[16] Lucia-Antoneta Chicos, Gheorghe Oancea, Camil Lancea, Daniel Bancila, "Software system of integrated and simultaneous engineering," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 238–241.

[17] A. Mets, "Ilya Prigogine'i raamatu "Tõsikindluse lõpp. Aeg, kaos ja uued loodusseadused" filosoofiline analüüs ja loodusseaduste käsitus (The philosophical analysis of I. Prigogine's book "The End of Certainty: Time, Chaos, and the New Laws of Nature" and treatment of natural laws)," Magistriväitekiri (Master's Thesis), Tartu, 2007.

[18] Leo Näpinen, "The need of the historical understanding of nature in physics and chemistry", *Foundations of Chemistry*, Vol.1, No.9, 2007, pp. 65–84.

[19] Zuzana Martináková-Rendeková, "Musical archetypes and memes – basic natural principles of musical work," in Proc. 7th WSEAS International Conference on Acoustics & Music: Theory & Applications, Cavtat, Croatia, 2006, pp. 13–17.

[20] Anon Sukstrienwong, "Genetic algorithms for multi-objectives problems under its objective boundary," in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 38–45.

[21] S. J. Gould, "The evolution of life on the earth", *Scientific American*, 271(4), 1994, pp. 84–9.

[22] Stuart A. Kauffman, *The Origins of Order: Self-organization and Selection inEevolution*, New York: Oxford University Press, 1993.

[23] Stuart A. Kauffman, *At Home in the Universe: The Search for the Laws of Self-organization and Complexity*, New York: Oxford University Press, 1995.

[24] Prigogine Ilya, Isabelle Stengers, "The new alliance, part two: An extended Dynamics: towards a human science of nature", *Scientia*, 112, 1977, pp. 643–653.

[25] Zwierlein, Eduard, "The paradigm of self-organization and its philosophical foundation", *Springer Series in Synergetics*, 61, On Self-Organization, 1994, pp. 288–298.

[26] F. A. Hayek, *The Fatal Conceit. The Errors of Socialism,* The University of Chicago Press, 1988.

[27] Leo Näpinen, "Iseorganiseerumismõtlemine, selle omandamise vajadus Eesti ühiskonnas (Self-organizing thinking, the need for acquirement it in Estonian society)", *Teaduslugu ja nüüdisaeg, IX, Teadusfilosoofia ja teadusmetodoloogia Eestis: olukord ja perspektiivid. Tallinn: Teadusajaloo ja Teadusfilosoofia Eesti Ühendus, Tartu Ülikooli Filosoofia Osakond*, 1994, pp. 158–181.

[28] Francis Heylighen, “The science of self-organization and adaptivity,” Center "Leo Apostel", Free University of Brussels, Belgium, 2001.

PAPER II

Loginov, D. (2012). On some technical issues and challenges in development and implementation of Autonomous Design System's (ADS) functionality. *International Journal of Mathematical Models and Methods in Applied Sciences. Issue 3, Volume 6, 142–151.*

On some technical issues and challenges in development and implementation of Autonomous Design System's (ADS) functionality

Dmitri Loginov

Abstract— In this paper technical issues and challenges regarding implementation of ADS functionality on AutoCAD platform are discussed. The results of the experiments for adaptation of the synergetic computer algorithm and its basic functionality to AutoCAD environment are presented. Technical aspects of the possible implementation of this algorithm in ADS system, as well as vectors' elements combination/permutation routine performance are analyzed. A brief philosophical outlook on the problem of modeling the engineering creativity in ADS/CAD systems was given.

Keywords— Self-organization, Synergetic computer, ADS, CAD, Pattern recognition, Synergetic neural networks, HVAC

I. INTRODUCTION

THE main characteristic of Autonomous Design System (ADS) is the ability to process complex non-routine tasks of the engineering design domain. In our approach the principles of synergetics and synergetic computer are used as a main tool in providing such functionality.

The concept of the synergetic computer was introduced by Hermann Haken in late eighties of the last century and it is based on the profound analogy between pattern formation and pattern recognition. Thus, the mathematical theory of synergetics was possible to use in derivation of the basic equations of synergetic computer. Synergetics (H. Haken's interpretation) can be considered as one of the modern, most promising research programs. It is oriented towards the search for common patterns of evolution and self-organization of complex systems of any kind, regardless of the concrete nature of their elements or subsystems (see e.g. [1], [2]).

In this paper the technical details of the theory of the synergetic computer and its possible realization on CAD (Computer Aided Design) and ADS platforms are discussed.

While the algorithm implemented is based largely on H. Haken models (for other synergetics-based models see [3]), the significant differences to the basic characteristics of the classical model were introduced and successfully tested in one of the most popular CAD environment. For the first time ever the synergetic computer was implemented on AutoCAD platform. The presented research constitutes the next intermediate step to the development and research of the fully functional Autonomous Design System (ADS).

II. STANDARD HAKEN MODEL

In this section a short overview of the mathematical background of the synergetic computer concept is presented. For in-depth discussion of the standard model see [4].

The basic dynamic equation of the synergetic computer or synergetic neural network is as follows:

$$\dot{q} = \sum_{k} \lambda_k v_k (v_k^{+} q) - B \sum_{k' \neq k} (v_{k'}^{+} q)^2 (v_k^{+} q) v_k \\ - C(q^{+} q) q + F(t), \tag{1}$$

where q is the state vector of a test (input) pattern with initial value q_0, λ_k is attention parameter, v_k is the prototype pattern vector, v_k^{+} is the adjoint vector of v_k, which obeys the orthonormality relation

$$(v_k^{+} v_{k'}) = \delta_{kk'}. \tag{2}$$

B, C are positive constants and $F(t)$ is fluctuating forces, which may drive the system out from its equilibrium state. Expression $v_k \cdot v_k^{+}$ acts as a matrix. This matrix has occurred in number of other publications and is called the learning matrix. The term

$$\xi_k = (v_k^{+} q) \tag{3}$$

is called the order parameter. The equation (1) describes the dynamics, which pulls the test pattern $q(t)$ into one of the prototype patterns v_{k_0}, namely the one to which $q(0)$ was closest. This means the pattern is being recognized by the system.

The corresponding dynamic equation of order parameters reads:

$$\dot{\xi}_k = \lambda_k \xi_k - B\sum_{k' \neq k} \xi_{k'}^2 \xi_k - C\sum_{k'=1}^{M} \xi_{k'}^2 \xi_k. \tag{4}$$

The order parameters obey the initial condition

$$\xi_k(0) = (v_k^+ q(0))$$

by which the initial values of order parameters in the evolution series are determined.

The equations (1) and (4) could be derived from corresponding potential function equations (5) and (6). That is

$$\dot{q} = -\frac{\partial V}{\partial q^+},\ \dot{q}^+ = -\frac{\partial V}{\partial q},$$

$$V = -\frac{1}{2}\sum_{k=1}^{M} \lambda_k (v_k^+ q)^2 + \frac{1}{4} B \sum_{k \neq k'} (v_k^+ q)^2 (v_{k'}^+ q)^2 + \frac{1}{4} C(q^+ q). \tag{5}$$

And

$$\dot{\xi}_k = -\frac{\partial \tilde{V}}{\partial \xi_k},$$

$$\tilde{V} = -\frac{1}{2}\sum_{k=1}^{M} \lambda_k \xi_k^2 + \frac{1}{4} B \sum_{k' \neq k} \xi_{k'}^2 \xi_k^2 + \frac{1}{4} C(\sum_{k'=1}^{M} \xi_{k'}^2)^2. \tag{6}$$

The potential function is used to represent the potential field in the space of k order parameters in which the fictitious particle, representing the dynamics of the test pattern or the corresponding order parameter, moves. The example of the potential V is shown on Fig.1.

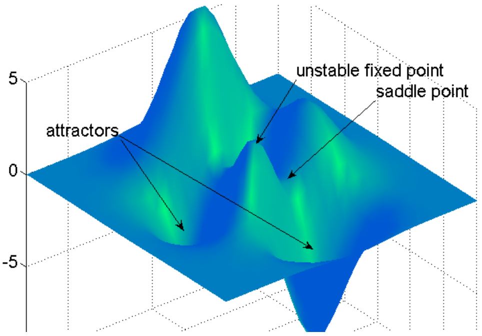


Fig.1 Example of potential function

In this plot the attractors are clearly visible. The attractors are the stable fixed points, which are represented by a bottom of each valley. The top of each mountain is an unstable fixed point. All points in the landscape from which the particle can roll down to the same attractors form the basin of attraction. Points of minimal height on ridges are saddle points.

The stable fixed points are at $q = v_k$, i.e. at the prototype patterns, and there are no other stable fixed points. The stable fixed points are equally characterized by $\xi_k = 1$, all other ξ's = 0.

The Haken's classical model is built up upon a number of assumptions. The most important of which are as follows:

- all attention parameters are equal and positive (i.e. balanced attention parameters)

$$\lambda_k = \lambda > 0, \quad \lambda = C \tag{7}$$

- the number of patterns is smaller than or equal to the number of features

$$M \leq N \tag{8}$$

- vectors v_k are subject to the condition

$$\sum_k v_k = 0 \tag{9}$$

- the following normalization holds

$$(v_k^T v_k) \equiv \sum_{j=1}^{N} v_{kj}^2 = 1 \tag{10}$$

- the number of features per pattern should be the same for all prototype and test vectors (equality of vectors' meaningful dimensions). That is, for each

$$v_1(k), v_2(l) \ldots v_n(m) \ ; \qquad k = l = \ldots = m, \tag{11}$$

where k, l,…,m are vectors' meaningful dimensions.

As vectors v_k are not necessarily orthogonal to each other, we need to construct the adjoint vectors, which may be formed as superpositions of the transposed vectors v_k^T :

$$v_k^+ = \sum_{k'} a_{kk'} v_{k'}^T. \tag{12}$$

The coefficients $a_{kk'}$ must be determined to satisfy the orthogonality condition (2). This may be done by multiplying (12) by $v_{k'}$ and interpreting $a_{kk'}$ and scalar products $(v_{k'}^T v_k)$ as elements of the corresponding matrices A and W

$$A = (a_{kk'})$$
$$W = [(v_k^T v_{k'})].$$

Equation (12) then can be written in the form

$$I = AW \tag{13}$$

and can be solved formally by $A = W^{-1}$.

A. Synergetic neural network

Synergetic computer may be realized by artificial neural networks, which act in a fully parallel manner. The resulting system is then called Synergetic Neural Network or SNN. SNN may be realized e.g. as one- or three-layer network. By using order parameter concept the network can be considerably simplified. As order parameters are defined by (3), and satisfy

$$\dot{\xi}_k = \xi_k (\lambda - D + B\xi_k^2), \tag{14}$$

where

$$D = (B + C) \sum_{k'} \xi_{k'}^2, \tag{15}$$

then, for a three layer network, we may use order parameters' as neurons' representation in the network's second layer. The input layer is represented by input (test) pattern vectors $q_j(0)$, and if SNN has to act as an associative memory, the third layer should consist of

$$q_j(t) = \sum_k \xi_k(t) v_{kj}, \tag{16}$$

where q_j is the activity of the cell j at the output layer, ξ_k is the final state of order parameter cell layer with $\xi_k = 1$ for $k = k_0$ (i.e. the pattern has been recognized) and $\xi_k = 0$ otherwise. The network may be further simplified by introducing a common reservoir D as in (14). In this way the number of connections may be further reduced.

B. Benefits of the synergetic computer (SNN) approach

The classical model has a number of advantages over traditional neural computers (networks). That is, compared to e.g. Hopfield Neural Network (HNN) it has following advantages.

1. The model training time is short.
2. The space complexity is low, for SNN it is np, where n is the number of features, p is the number of patterns and $p \ll n$, while for HNN it is n^2 .
3. The processor time complexity for the recognition process is also low. For SNN it requires p^2 multiplications and p additions and for HNN n^2 multiplications, n additions.
4. There are no so-called *pseudo-states*. This is most important property. It may be proved that besides the prototype vectors there are no other attractors.

In HNN, the system has the following potential functions. In the case of discrete HNN:

$$V = -\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} w_{ij} v_i v_j + \sum_{i=1}^{N} \theta_i v_i$$

And in the case of continuous HNN:

$$V = -\frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} v_i I_i + \sum_{i=1}^{N} \frac{1}{\tau} \int_0^{v_i} g^{-1}(t) dt$$

This potential function can not guarantee that all of the attractors are actually the desired ones. In the construction of HNN, no matter how carefully to learn and adjust the weights w_{ij} and threshold value θ_i, it is still difficult to avoid/control the generation of pseudo-state. In Haken model, considering the system's dynamic behavior, the precise control of the potential function (energy function) in the energy potential field and not the type of connection of neurons nor the non-linear mappings of them, allow thereby to eliminate the pseudo-states.

5. The association effect: all of the prototype patterns can be clearly and equally identified.

C. *Limitations of the synergetic computer*

The limitations of the synergetic computer model, as it often happens, are extensions of its advantages. Namely, the most prominent disadvantage worth noticing is that the system always selects the winning pattern out from the patterns presented, even if they are not actually the right ones. This occurs due to the fact that the biggest order parameter formed initially as a dot product among all available vectors will always win the competition. Therefore, if the vectors' set does not contain the right prototype pattern, it will not be recognized by the system. It is true, however that the system will recognize test pattern (vector) that is the most close to the prototype pattern (vector).

Some problems with correct identification may occur when the vectors have different order of elements (features). This is because the inner (dot) product value's magnitude depends on the order of elements in the sequence (vector). Such identification difficulties may occur in a case of different elements combinations' testing e.g. in points/distances permutation example (see onward in section IV).

III. SYNERGETIC COMPUTER ON AUTOCAD

Implementation of the synergetic computer core functionality on AutoCAD platform is the intermediate step towards the realization of SNN in real life design process as a main tool in ADS modeling. ADS is defined as an advance CAD system, which *has AI functionality and particularly the functionality to solve the creative tasks* of the engineering design process. ADS is opposed to the conventional CAD systems, (see e.g. [5]) which normally automate routine parts of the design process and generally have no AI capabilities. In this section some technical questions of implementation of synergetic computer basic functionality are discussed.

To this end, the objective was established to create an AutoCAD application that can recognize a number of simple geometric structures. At first the MATLAB prototype was created in order to test the basic functionality of the model and then the algorithm was implemented in AutoCAD environment. For the sake of simplicity of the presentation, in AutoCAD environment only three different patterns were implemented. Actually, the number of patterns tested (in MATLAB) was bigger and the noisy patterns were elaborated as well.

The prototype patterns for our case were chosen among AutoCAD (Acad) polygon entities (more specifically, these constitute of polyline objects in Acad database), namely, the triangle, square and hexagon. As Acad is a vector graphics software, we had to invent the way of representing our prototype vectors properly. We had chosen to code vectors' elements as a relative measure between entities' endpoints i.e. the distances between polygons' vertices, as shown on Fig. 2.

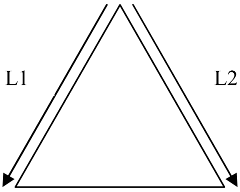


Fig. 2 Representation of prototype vectors' elements in AutoCAD graphics system.

Thus, in a case of triangle the raw prototype vector is as simple as $v_k = (L_1, L_2)$. We have now three state vectors to recognize (shown as raw vectors):

$$\begin{aligned} v_0 &= (L_{01}, L_{02}, L_{03}) \\ v_1 &= (L_{11}, L_{12}) \\ v_2 &= (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}). \end{aligned} \tag{17}$$

From (14) we may deduce a discrete equation for the order parameter evolution

$$\begin{aligned} &\xi_k(n+1) - \xi_k(n) \\ &= \gamma(\lambda_k - D + B\xi_k^2(n))\xi_k(n), \end{aligned} \tag{18}$$

where γ is the iteration speed and term D is according to (15). The corresponding evolution for the case of three order parameter is shown on Fig. 3.

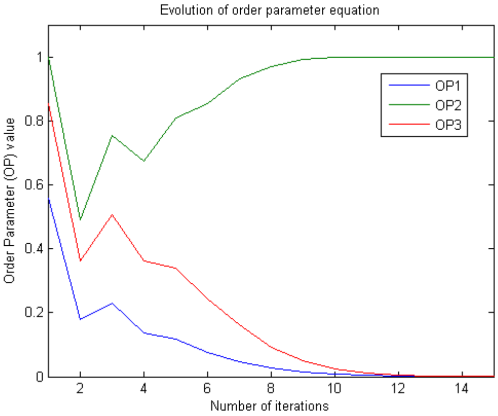
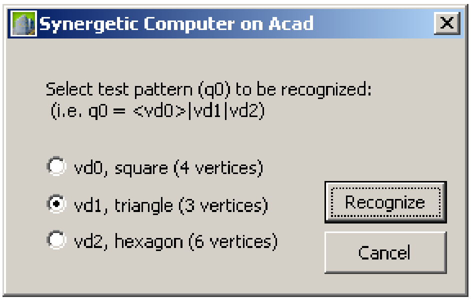


Fig. 3 Evolution of order parameter equations

From Fig. 3 is seen that the system converges after twelve steps, i.e. the winning order parameter becomes $\xi_1 = 1$ and other order parameters $\xi_2 = \xi_0 = 0$. Thus, the prototype pattern that corresponds to ξ_1 will be recognized. Note that due to the fact that the biggest initial order parameter $\xi_k(0)$ will always win the competition (in the case of balanced λ_k), the iteration (18) may be omitted and the resulted system remarkably simplified. The learning process will be then restricted to satisfying (10) and solving (13). If, however, we are dealing with normalized vectors, finding of adjoint vectors means just transposing and the whole dynamics reduces to forming the inner products $(v_k^T q_0)$, which further reduces the complexity of numerical computations.

The user interface of the resulted system is shown on Fig. 4.



Select test pattern (q0) to be recognized:
(i.e. q0 = <vd0>|vd1|vd2)

○ vd0, square (4 vertices)
● vd1, triangle (3 vertices)
○ vd2, hexagon (6 vertices)

Fig. 4 GUI of SNN application

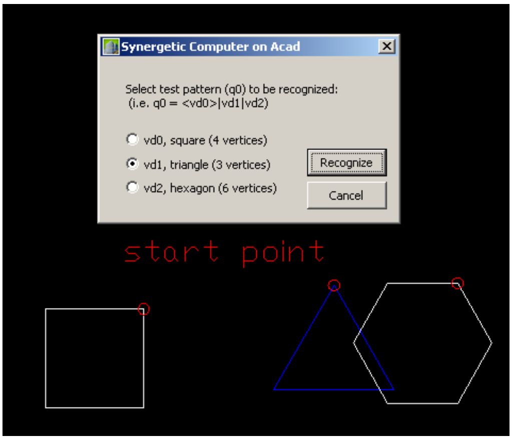
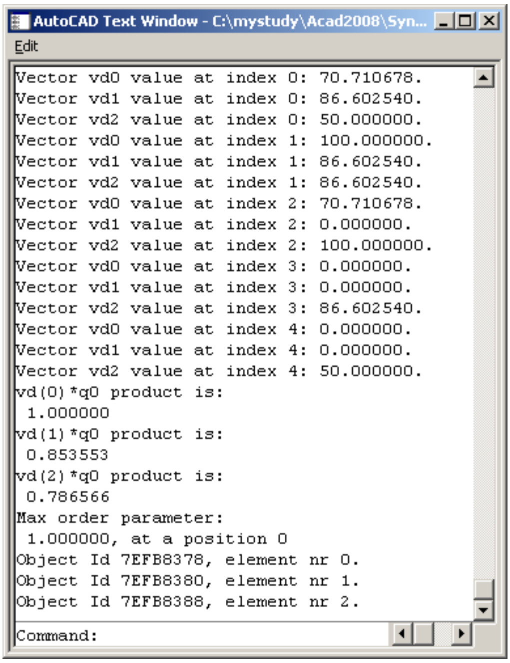


Fig. 5 AutoCAD graphics screen, the identified pattern is selected

The user may choose the pattern desired by selecting respective radio button, then by pressing the action button the system performs the recognition process and selects the identified polygon by changing its color property (to blue), see Fig. 5.

The location and position of the test patterns has no effect on the recognition results. The polygons may be rotated, or, even overlapped on each other, the system still successfully identifies the figures.

On Fig. 6 the Acad text window with implemented system's output messages is shown.



```
Vector vd0 value at index 0: 70.710678.
Vector vd1 value at index 0: 86.602540.
Vector vd2 value at index 0: 50.000000.
Vector vd0 value at index 1: 100.000000.
Vector vd1 value at index 1: 86.602540.
Vector vd2 value at index 1: 86.602540.
Vector vd0 value at index 2: 70.710678.
Vector vd1 value at index 2: 0.000000.
Vector vd2 value at index 2: 100.000000.
Vector vd0 value at index 3: 0.000000.
Vector vd1 value at index 3: 0.000000.
Vector vd2 value at index 3: 86.602540.
Vector vd0 value at index 4: 0.000000.
Vector vd1 value at index 4: 0.000000.
Vector vd2 value at index 4: 50.000000.
vd(0)*q0 product is:
 1.000000
vd(1)*q0 product is:
 0.853553
vd(2)*q0 product is:
 0.786566
Max order parameter:
 1.000000, at a position 0
Object Id 7EFB8378, element nr 0.
Object Id 7EFB8380, element nr 1.
Object Id 7EFB8388, element nr 2.

Command:
```

Fig. 6 AutoCAD text window, program control messages

A. *Differences from standard model*

The model differs from classic Haken representations by the following points: (8), (9) and (11). More specifically, in our Acad model the number of patterns allowed to be bigger or equal to the number of features $M \geq N$. Additionally, numerical simulations have shown that the model works well in situation where $M \gg N$.

We have omitted the condition (9) in our model, as tests have proved it to be redundant.

The size of the prototype vectors is different in our implementation, thus the (11) is not satisfied. However, the model still performs well. Here, of course, it is the number of meaningful dimensions that is important. For a system to be solvable, the trailing zeros should be added to vectors of different size:

$$v_0 = (L_{01}, L_{02}, L_{03}, 0, 0)$$
$$v_1 = (L_{11}, L_{12}, 0, 0, 0)$$
$$v_2 = (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}).$$

All these modifications, while simplifying the system and allowing for a greater flexibility, do not degrade the model's performance nor obscure general properties of SNN. It is worth noting, that the recognition rate of the model so far was 100%. This could be explained e.g. by the fact that only the noiseless patterns were used for the test vectors.

B. Tools and technologies used

AutoCAD Architecture 2008 as a main framework for the model, VC++ 8.0, ObjectARX 2007, Eigen3, MATLAB 7.0.

C. Further research

The directions for further research connected with this section discussion are outlined below.

In our model the balanced attention parameters were used (according to (7)) and $\lambda_k = C = B = 1$. It is possible to use the unbalanced attention parameter technique e.g. $\lambda_1 = 0,6$ and $\lambda_2 = 0,4$. In that case the biggest value will influence the evolution of the order parameters and it is possible for initially smaller order parameter eventually to win the competition. Attention parameters thus could be used as an additional instrument to guide the selection of order parameters in situations where selection criteria based solely on ξ_k values are not sufficient. Those situations are likely to arise when dealing with more complicated patterns and process objectives, as e.g. in ADS structures or, just as simple, as in treatment of vectors of different size, like (17). The implementation of SNN in ADS, as well as the treatment of noisy patterns, is a subject for a further research.

IV. POSSIBLE IMPLEMENTATION IN ADS

Of course, the implementation of the synergetic computer for the recognition of simple AutoCAD entities is not the aim of its own. Instead we want to apply these principles to the useful recognition scenarios e.g. as a component of ADS system. Although there are plenty of different application possibilities that could be elaborated, let us research the one from HVAC (Heating Ventilation Air Conditioning) engineering domain. More specifically, we want to automate the process of building's outer peripheries (e.g. outer walls) data acquisition from AutoCAD environment. To this end, we will use the Heatloss original software developed in author's earlier research as a framework for software agents testing and implementation on AutoCAD platform.

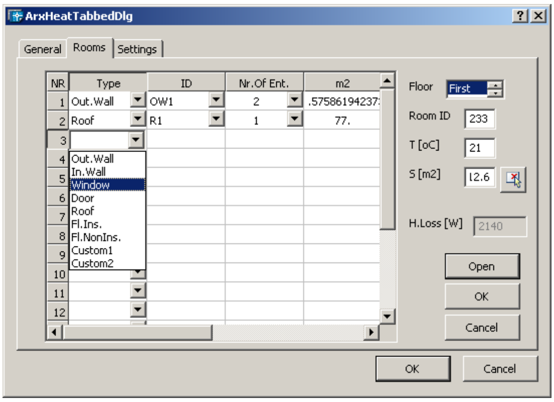
The process of the selection of outer walls of the building clearly falls into the engineer's creative activity class, if we consider this process as a dynamic synergetic system. It is quite easy for the human designer visually identify the outer walls from other geometry on the graphic screen. For the computer, however, it is not an easy task, if we treat all graphical data equally in the sense of human visual perception. This is our task to treat the underlying vector entities as patterns, such that we could apply the principles of the self-organization theory and use the synergetic tools described above. Note that the same task may be solved by "traditional" cybernetic approach methods e.g. by introducing some additional metadata to the graphical objects in order to make it possible for a straightforward computational identification. Such parametrical approach is very common nowadays in information systems design; in CAD domain it is used e.g. in BIM (Building Information Modeling) applications.

In the following subsections we briefly explain the Heatloss software's related existing functionality and describe its possible improvements by exploiting synergetic computer properties/advantages. We will also discuss the issues of optimal permutation selection realization of test patterns.

A. Additional functionality for HeatLoss software

The HeatLoss software is ObjectARX module that automates the calculation of the building's heat losses. Below is the brief explanation of related GUI.

The Rooms tab is a main working UI of the program. It has a grid control, which is similar to a spreadsheet by its functionality. The grid control used in this program is very powerful custom control. It has a rich set of different features; most of them is not used currently in HeatLoss, but are planned to the future releases. In the next version, for example, we plan to add a drag-and-drop capability to the grid.



NR	Type	ID	Nr.Of Ent.	m2
1	Out.Wall	OW1	2	.57586194237:
2	Roof	R1	1	77.
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Floor First
Room ID 233
T [oC] 21
S [m2] 12.6
H.Loss [W] 2140

Fig. 7 HeatLoss application rooms peripheries window (Rooms tab)

User selects type of the periphery, its ID and program brings his immediately to the AutoCAD drawing screen (DWG) where he picks characteristic points. Next the program computes area of the periphery and heat loss of the room, based on the data in the Settings tab and room's inner temperature and displays UI back to the user. The process repeats. Of course it is possible to modify data in the grid and in the tab after initial calculation is done. User may change, for instance, number of entities, area of the periphery, U value, inner temperature etc and program updates the heat loss value accordingly.

We want to add to this program the additional functionality, i.e. the ability for the system to select the outer periphery (outer wall) on its own, using synergetic computer approach. We shown earlier that such ability transform the conventional CAD system into ADS system.

This can be achieved by the following. When user selects the ID of the type of the periphery chosen, the system instead of the prompting the user to manually select the periphery (wall) length perform the identification process based on the prototype patterns stored. Upon the identification, the system acquires technical parameters of the room identified and calculates the heat loss of this room.

One way to compare room's characteristic vectors is to use all possible combinations (more specifically, permutations) of the representing distances (see Fig. 8) and to form corresponding test vectors. The basic principle is the same as described in section III.

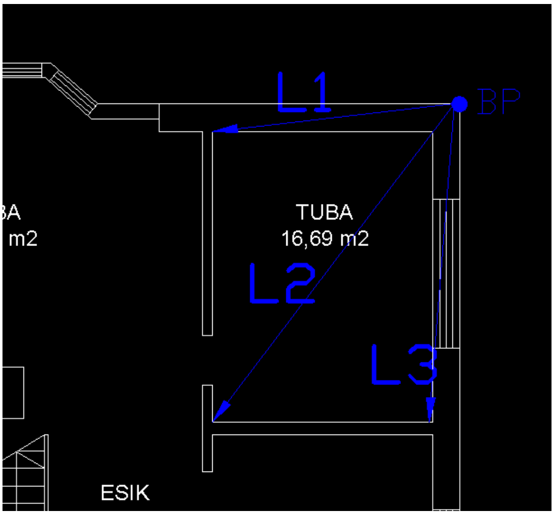


Fig. 8 Representation of the room geometry as test vector elements

Here in Fig. 8 shown three leaders L1, L2, L3 drawn from the base point BP. This combination corresponds to the prototype pattern vector being recognized. However, in order to get these points (distances) into comparison in AutoCAD, we have to select all line/polyline type entities in the vicinity of BP. Thus we will have eventually a number of different distances from BP to the corresponding entities' start/end points; then we have to permute these distances such as to get the right vector elements combination (e.g. L1, L2, L3). For example, if we have 42 distances (i.e. 42 possible elements of the test vector) and the prototype vector consisting of 3 elements, we will have as much as $C_k^r = \frac{k!}{(k-r)!} = \frac{42!}{(42-3)!} = 68880$ different permutations.

Therefore in order to preserve numerical computation efficiency it is very important what kind of permutation computation algorithm is chosen. In the next subsection we will discuss this issue in more detail.

B. Patterns' features selection optimization

As the computational speed in engineering applications is quite important [6], we do not want to degrade software performance in permutations module as well. This is particularly important for CAD and ADS systems. That is why ObjectARX (C++) technology is chosen for our systems' implementation. In comparison with other AutoCAD development technologies (VBA, .NET, AutoLisp/DCL see e.g. [7]) ObjectARX is the most powerful IDE creating the fastest and most compact ARX (DLL) modules available. The processing speed of the permutation algorithm depends on the functions and programming language constructs chosen for one particular application. Let us review some widely used C++ permutation algorithms/functions. We then compare them and select the most effective (optimal) routine for our system implementation. That is, we will perform the pattern selection process optimization for our ADS module.

Howard Hinnant [8] has performed the tests amongst most widely used permutation (combination) algorithms. Below is a brief review of the results. There were 3 different approaches, solutions A, B, and C.

Solution A.

The standard library has *std::next_permutation* and it is possible trivially build a next_k_*permutation from it* and a *next_combination* from that (Fig. 9):

```
template<class RandIt, class Compare>
bool next_k_permutation(RandIt first, RandIt
mid, RandIt last, Compare comp)
{
    std::sort(mid, last, std::tr1::bind(comp,
std::tr1::placeholders::_2
                                             ,
std::tr1::placeholders::_1));
    return std::next_permutation(first, last,
comp);
}
```

Fig. 9 Example code for solution A

The performance results of this solution are as follows:

```
N = 100, r = 5, visits = 75287520
    next_combination total = 4519.84 seconds
    next_combination per visit = 60034.3 ns
```

Fig. 10 The performance printout of solution A

Solution B.

This solution is developed by Hervé Brönnimann (called

N2639) and can be found at [9]. This proposal adds eight algorithms (*std::next_partial_permutation, next_combination, next_mapping, next_repeat_combination_counts*, their counterparts *std::prev_partial_permutation, std::prev_combination, std::prev_*-mapping, *std::prev_repeat_combination_counts*, with their overloads) to the header <*algorithm*>*,* for enumerating permutations and combinations, with and without repetitions. They mirror and extend *std::next_permutation* and *std::prev_permutation*. For sizes known at compile-time, these algorithms can generally be simulated by a number of nested loops. The performance results of this solution are shown on Fig. 11.

```
N = 100, r = 5, visits = 75287520
    next_combination total = 6.42602 seconds
    next_combination per visit = 85.3531 ns
```

Fig. 11 The performance printout of solution B

Solution C.

Finally there is a solution C found here [10]. This solution has a different signature/style and is called *for_each_combination (for_each_permutation)*, and is used much like *std::for_each*. The driver code between the timer calls is as follows:

```
Clock::time_point t0 = Clock::now();
f = for_each_combination(v.begin(), r,
v.end(), f);
Clock::time_point t1 = Clock::now();
```

Fig. 12 The driver code between the timer calls of solution C

The performance results of this solution are shown on Fig. 13.

```
N = 100, r = 5, visits = 75287520
    for_each_combination = 0.498979 seconds
    for_each_combination per visit = 6.62765
ns
```

Fig. 13 The performance printout of solution C

Solution C is 12.9 times faster than solution B, and over 9000 times faster than solution A.

We consider this a relatively small problem: only 75 million visits. As the number of visits increases into the billions, the discrepancy in the performance between these algorithms continues to grow. Solution A is already unwieldy. Solution B eventually becomes unwieldy. Solution C is the highest performing algorithm to visit all combinations/permutations author aware of.

Thus we have to choose the approach of the solution C for our systems coding.

C. Further research

The actual implementation of the algorithms described in this section in ADS system is left for a further research. More specifically, this research is currently running and we obtained some preliminary results in the realization of solution C on AutoCAD in outer wall recognition routine. The description of the results of the experiments is the subject of further publications.

V. SOME GENERAL REMARKS ON THE SUBJECT

In this section the general and somewhat philosophical remarks on the theory of ADS and modeling of the creative part of the design process are presented.

The ongoing research in AI domain and in the field of general technology shows that traditional methods of solving engineering problems based on formal logic and systematical approach shifts toward the new unrevealed, presently undocumented features of human mind and intelligence (more closely to the characteristics of self-organization?). There are neural networks, which try to copy the functionality of biological brain cells – neurons, fuzzy logic and modeling (for a contemporary research on fuzzy dynamic systems see e.g. [11]), expert systems, evolutionary programming/computing, knowledge-based systems, swarm and genetic algorithms and so on.

The routine parts of the engineering design process could be successfully modeled with the help of cybernetics. It is really the art of combinatorial manipulation and constructing to fulfill the goal, using the already known or novice technology, IT in this case. As it is based on cybernetics, it falls down to organizational theories, contrary to self-organization paradigm, and therefore is not connected with the subject of interest of this paper.

Let us take a look at the notions of organization and self-organization from the concept point of view. The concept of organization denotes the process that leads to the rise of goal-oriented structures due to conscious human goal-directed action or some external ordering influence, and the concept of self-organization would denote the process that leads to the rise of goal-oriented structures beyond conscious human goal directed action or some external ordering influence. Although the term "self-organization" is widely used (and more appropriate) in the field of synergetics, it has been utilized in cybernetics as well. In cybernetics, however, it has different meaning (from the philosophical point of view). In cybernetics and systems engineering self-organization is understood as an effect of an external ordering factor (e.g. self-organizing map in [12]). In synergetics self-organization is understood as the rise of harmonious behavior distinguished from man's intervention and from external (with regard to the system) ordering factors. External factors (e.g. strong non-equilibrium) are indispensable for self-organization, but only as conditions, not as ordering forces.

Hopefully, it is possible to imitate the creativity (at least to some degree) by means of synergetic modeling. Could we model the creative part of the (engineering) design process as well? To answer this question we must analyze the synergetic approach and compare it with traditional information technology modeling instruments e.g. with cybernetics.

In cybernetics as well as in synergetics the objective processes are modeled in order to control them. The cybernetic models make it possible for man to strive for the desirable results using the program created by him. The synergetic models take into account that the programs form in the course of self-organization [13].

All exact sciences (and also the traditional scientific cognition) are model-based. These are exact only within that model. Therefore it is not possible to explore/predict/study adequately the real world by means of "exact" sciences by definition. We can use exact sciences to explore models. CAD systems, ADS frameworks are examples of design systems' models. Both cybernetics and synergetics are exact sciences as well. So we can use these disciplines only for the development and research of models of the underlying real world's phenomena and not for the investigation of the real world itself. It must be underlined that in exact sciences the approach to the interaction between organization (management) and self-organization does not go (and due to the specificity of exact sciences must not go) farther from certain boundaries.

The limits mean that exact sciences in their models of influence upon self-organization give only such recommendations according to which the future state of an object of management is given from the outside. Exact sciences do not make any contribution to the opening of the creative potential of the elements of the system [13]. So we cannot use standalone synergetic methods (a kind of exact science) to explore the creative potential of the system (and self-organization). As the synergetics is exact science and is based on mathematics, it has known limitations in its capability to explore the real world. But still we can use it to *create the better models* of the real life systems, not to understand these systems completely. On the other hand, building more adequate models of the environment leads to a better understanding of the environment itself. And, therefore, may lead us to a new level of understanding, to help us form a new paradigm and from within it - to model even more precisely, closely to the real world.

Synergetics better than cybernetics models the processes of the real world which is ultimately the self-organizing system. So we can use principles of synergetics in conjunction with traditional computing technology to model some aspects of the real systems. It is worth showing how creativity is understood in synergetics. The meaning of the word creative is the unpredictability and unavoidability of the unknown. The creative chaos is the field of unknown and unpredictable chances. The meaning of the word is closely related to such concepts as non-equilibrium condition and conditions close to equilibrium.

Synergetics accentuates also one necessary condition of self-organization: the order arises from chaos only under the condition of *strong non-equilibrium*. It is necessary to distinguish strictly chaos under the conditions close to equilibrium (in which, generally speaking, self-organized structures can only decompose) from chaos under the strongly non-equilibrium conditions (in which composing of structures through self-organization can take place) [13]. The former type of chaos is non-creative, the latter is creative.

In engineering design process theory the meaning of the words "creative" (and "creativity") is slightly different. Here the word "creative" denotes a non-routine part of the design process. Contrary to the routine procedures where inputs and outputs of the system are known or predictable, the creative part of the process deals with output data that is mainly unknown, although the field of possibilities (possible outputs, similar to synergetics theory) is generally defined. This is true in ordinary design scenarios where the ultimate goal of the design procedure is known. When the output data of the system is completely undefined and unknown, then we are dealing with the system that generates some new design information (i.e. invention mechanism). Note, that the input data in majority of cases is defined (both in ordinary design scenarios and in invention apparatus). The modeling of the technical invention processes is even more complicated (if not impossible) than imitating the creative part of the conventional design process (i.e. the process where the field of the possibilities of the output information is defined). There is a hope that using the methods of synergetics and the philosophy of self-organizing systems we can try to address the problems of modeling creative design in a more precise and better manner. The new science which accepts creativity based on chance and irreversibility in nature, and considers the fundamental indeterminacy of the whole history of nature and of human society should evolve to acknowledge the potential of this approach.

Basically, we can consider a model as an idealized version of the real system. The model is always a simpler and more primitive than the real system. The traditional tool for creating engineering design models nowadays is a Computer Aided Design (CAD) system. For a creation of a new CAD system we use CAD programming. Thus, CAD programming is essentially construction of the model (computer program) for the model (CAD application) of a model (engineering design, project) of the system (e.g. engineering installation). Such models' cascading occurs e.g. in a case when we are programming under some existing CAD platform, let's say under AutoCAD. On this level of abstraction the model itself is very precise (it is nested into surrounding model etc.) and perfectly describable by mathematics.

The aim is to try to add to this model the properties/specifications of the self-organizing systems' behaviors. The author does not really think that the model will be capable of substituting the engineer completely in the process of producing creative design. But there is a hope that the model built in the spirit of synergetics could facilitate the emergence of the elements of the creativity in engineering design in which the human participates as well. It is likely that these models in cooperation with the operator (engineer) can function more effectively in creating new designs.

VI. CONCLUSION

This paper describes the results of the synergetic computer implementation on AutoCAD platform.

A number of useful modifications to the standard model were committed, tested and successfully implemented in the form of AutoCAD application. This is the first documented usage of SNN on AutoCAD platform.

The presented research constitutes another step to the development and research of the fully functional Autonomous Design System (ADS).

Synergetic computer has one major advantage, compared to the traditional neural computers, namely, there are no so-called pseudo-states, into which the system could be trapped in. It may be proved that besides the prototype vectors there are no other attractors. In addition, the functionality of SNN, especially pattern recognition mechanism and treatment of ambiguous and noisy patterns closely resembles the functionality of biological neural systems, including human brain [14]. This point supports the whole philosophical study of the self-organization phenomenon and is the main reason for selecting synergetic computer approach for ADS implementation.

A closer look on the problem of adding synergetic ADS functionality to the existing HVAC CAD application was taken. The technical part of the implementation of C++ permutation functionality and its performance considerations as part of the currently developing ADS system was discussed.

We also gave a short philosophical outlook on the problem of modeling the engineering creativity in ADS.

REFERENCES

[1] H. Haken, H. Knyazeva, “Arbitrariness in nature: synergetics and evolutionary laws of prohibition,” *Journal for General Philosophy of Science*, No.31, 2000, pp. 57–73.

[2] H. Haken, *Synergetics. Introduction and advanced topics*, Springer, 2004.

[3] Hermann Haken, *Brain Dynamics. An Introduction to Models and Simulations*, Second Edition, Springer, 2008

[4] H. Haken, *Synergetic Computers and Cognition. A Top-Down Approach to Neural Nets*, Springer, 2004.

[5] Lucia-Antoneta Chicos, Gheorghe Oancea, Camil Lancea, Daniel Bancila, “Software system of integrated and simultaneous engineering,” in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 238–241.

[6] Cosmin Marian Poteras, Mihai Mocanu and Constantin Petrişors, “A Distributed Design for Computational Steering with High Availability of Data”, *International Journal Of Systems Applications, Engenering & Development, (NAUN),* Issue 1, Volume 6, 2012, pp. 52–61.

[7] Sever Alexandru HABA, Gheorghe OANCEA, Anisor NEDELCU, “Data Exchange Software Module between AutoCAD and CATIA Environments,” in Proc. 11th WSEAS International Conference on Systems Theory And Scientific Computation (ISTASC '11), Florence, Italy, August 23-25, 2011, ISBN: 978-1-61804-027-5, pp. 134–137.

[8] Howard Hinnant. (2012, May 10). Stackoverflow C++ discussion. [Online].Available: http://stackoverflow.com/questions/2211915/combination-and-permutation-in-c

[9] Hervé Brönnimann. (2012, May 10). *Algorithms for permutations and combinations, with and without repetitions*. [Online].Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2008/n2639.pdf

[10] Howard Hinnant. (2012, May 12). *Combinations and Permutations*. [Online].Available: http://home.roadrunner.com/~hinnant/combinations.html

[11] Nikos Mastorakis, Olga Avramenko, “Fuzzy models of the dynamic systems for evolution of populations,” in Proc. 3rd WSEAS International Conference on Mathematical Biology and Ecology, Gold Coast, Queensland, Australia, 2007, pp. 27–32

[12] Supakit Siripanadorn, Wipawee Hattagam, Neung Teaumroong, “Anomaly detection using self-organizing map and wavelets in wireless sensor networks,” in Proc. 10th WSEAS International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan, 2010, pp. 291–297.

[13] Leo Näpinen, “Philosophical Foundations of Synergetic Modelling,” *Proceedings of the Estonian Academy of Sciences. Humanities and Social Sciences*, Vol.4, No.42, 1993, pp. 378–390.

[14] D. Loginov, “Synergetic Modelling – Application Possibilities in Engineering Design,” in Proc. 10th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE '10), Iwate Prefectural University, Japan, 2010, pp. 111–116.

PAPER III

Loginov, D. (2012). Advanced Synergetic Computer Algorithm for Vector Graphics Geometry Recognition and Its Realization as Part of HVAC ADS on AutoCAD Platform. *Advances in Mathematical and Computational Methods, 121–126.*

Advanced Synergetic Computer Algorithm for Vector Graphics Geometry Recognition and Its Realization as Part of HVAC ADS on AutoCAD Platform

DMITRI LOGINOV
Department of Environmental Engineering
Tallinn University of Technology
Ehitajate tee 5, 19086 Tallinn
ESTONIA
dmitri.loginov@gmail.com

Abstract: - In this paper the results of the experiments of adaptation of the synergetic computer algorithm for one particular Autonomous Design System (ADS) functionality on AutoCAD platform are presented. The system is built capable of recognizing the buildings rooms' outer peripheries in a manner close to the human visual perception. While the algorithm implemented is based largely on H. Haken models, the significant differences to the basic characteristics of the classical model were introduced and successfully tested in one of the most popular CAD environment. For example, in our approach, the number of meaningful elements of the prototype pattern vectors is varied and the number of features per pattern is allowed to be smaller than the number of patterns (M≥N). Also the order parameter generation algorithm is completely different from the standard model and highly optimized for AutoCAD environment.

Key-Words: - Self-organization, Synergetic computer, CAD, ADS, Pattern recognition, HVAC

1 Introduction

The brief overview of the concept of the synergetic computer's standard model, as well as of its basic mathematical implementation was given in the second paper [1] of the series on this subject. In [2] and [3] one may find a more extended coverage of the concepts of synergetics and synergetic computer. In [4], [5] the author has possessed the general analysis of the possibilities for using synergetics in modelling the creative part of the engineering design systems and in ADS, and brought the brief philosophical outlook on the problem. Therefore, in this paper we just remind the reader about the notion of synergetics and discuss the technical issues (e.g. order parameter generation algorithms) contributing to our latest results in more detail.

Synergetics (H. Haken's interpretation) can be considered as one of the modern, most promising research programs. It is oriented towards the search for common patterns of evolution and self-organization of complex systems of any kind, regardless of the concrete nature of their elements or subsystems (see e.g. [6]).

In this paper the technical details of the modified synergetic computer algorithm and its realization as a part of HVAC (Heating Ventilation Air Conditioning) ADS framework are discussed. The presented research constitutes the next step to the development and research of fully functional ADS subsystem.

2 Objectives and Methods

Current research is intended to answer the following research questions. The questions are ordered by its scope and each preceding question may be broken down to the subsequent one.

- Is synergetics capable of helping to model the creative part of the engineering design process?
- Can synergetic algorithms help in developing Autonomous Design System (ADS) software?
- Can synergetic computer theory be used for the outer walls recognition mechanism in HVAC software on AutoCAD platform?

The method to conduct the study includes the following. At first, the needed theoretical base should be prepared by studying the H. Haken synergetics mechanisms and particularly the theory of synergetic computer. The next step is developing numerical models and its implementation on AutoCAD platform. These are considered in [4], [1] respectively. Then a further optimization should be done to suit these models for a particular HVAC ADS scenario. As a result, a software application, which elaborates the synergetic model, should be completed.

So far, the first and second questions may be answered in a positive manner. In this paper the last question is considered bringing the implementation technical details.

2.1 Tools and technologies used

AutoCAD Architecture 2008 as a main framework for the model, VC++ 8.0, ObjectARX 2007, Eigen3, MATLAB 7.0, Boost 1.49.0.

3 Results

In this section the results of the implementation of the synergetic computer core functionality and its optimization for vector graphics CAD environment and one particular HVAC problem are discussed.

3.1 Adaptation of the synergetic computer algorithm for AutoCAD platform

In order to extend the functionality of synergetic computer on AutoCAD platform the number of additional modifications to the standard Haken model were committed. These are considered in more detail in the following subsections. This optimization together with the system implementation may be regarded as an important contribution to the classic synergetic computer algorithm. All these modifications, while simplifying the system and allowing for a greater flexibility, do not degrade the model's performance nor obscure general properties of synergetic computer. It is worth noting that the recognition rate of our model so far was 100%. This could be explained e.g. by the fact that only the noiseless patterns were used for the test vectors.

3.1.1 Quick Haken Algorithm

Normally, in classical model order parameters obey the initial condition $\xi_k(0) = (v_k^+ q(0))$ and are changing according to the dynamic equation of order parameters:

$$\dot{\xi}_k = \lambda_k \xi_k - B\sum_{k' \neq k} \xi_{k'}^2 \xi_k - C\sum_{k'=1}^{M} \xi_{k'}^2 \xi_k . \qquad (1)$$

In (1), (2) and in the previous equation the following variables are used: q is the state vector of a test (input) pattern with initial value q_0, λ_k is attention parameter, v_k is the prototype pattern vector, v_k^+ is the adjoint vector of v_k, which obeys the orthonormality relation, B, C are positive constants and ξ_k is order parameter. The corresponding evolution e.g. for the case of three order parameter is shown on Fig. 1.

The system converges after twelve steps, i.e. the winning order parameter becomes $\xi_1 = 1$ and others $\xi_2 = \xi_0 = 0$. Thus, the prototype pattern that corresponds to ξ_1 will be recognized. It is possible, however, to prove that the biggest initial order parameter $\xi_k(0)$ will always win the competition (in case of balanced λ_k), and the iteration (1) may be omitted and the resulted system remarkably simplified.

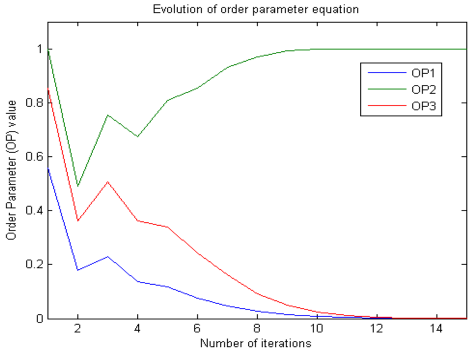


Fig. 1 Evolution of order parameter equations

Such simplified computational process is called *Quick Haken Algorithm*, since the overall learning time and the complexity of numerical computations is significantly reduced. We are using this enhanced algorithm in our application.

3.1.2 Cosine similarity OP generation

The standard model of synergetic computer assumes that the order parameters (OP) are found as a dot product (DP) between prototype and test pattern vectors:

$$\xi_k = (v_k^+ q) \qquad (2)$$

Such approach is justified in pattern recognition scenarios where the process variables satisfy the conditions of classical synergetic computer model (see [1]). The experiment showed that DP based order parameter generation does not work in our system and the winning prototype patterns are identified erroneously. The reason for this is the fact that DP value depends on the order of the vector elements and, since in permutation series this order varies, it is virtually possible that maximum OP is achievable at different elements combinations than those that are equal to the prototype vectors. In order to correct the behavior of the system we have tried to introduce additional competition among selected maximum OP values' patterns, but this approach, as a number of patterns grows, eventually leads to an unwieldy system. Instead, we decided to introduce new OP generation principle.

We have found that OP generation based on *cosine similarity* (CS, also known as a correlation coefficient) measure works well in our situation, and it is sufficient to arrange a single competition set to determine the correct winning pattern. CS OP generation equation, given two vectors A, B, is as simple as:

$$\cos\theta = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}}. \quad (3)$$

If compared vectors are identical the CS OP value obtain its maximum, and the resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

3.1.3 Test/prototype vectors dimensions

The classic Haken representations state that the number of features per pattern should be the same for all prototype and test vectors (equality of vectors' meaningful dimensions). That is, for each

$$v_1(k), v_2(l) \ldots v_n(m);$$
$$k = l = \ldots = m, \quad (4)$$

where *k, l,…,m* are vectors' meaningful dimensions.

The size of the prototype vectors is different in our implementation, thus the (4) is not satisfied. However, the model still performs well. Here, of course, it is the number of meaningful dimensions that is important. For a system to be solvable, the trailing zeros should be added to vectors of different size:

$$\begin{aligned} v_0 &= (L_{01}, L_{02}, L_{03}, 0, 0) \\ v_1 &= (L_{11}, L_{12}, 0, 0, 0) \quad (5) \\ v_2 &= (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}). \end{aligned}$$

3.1.4 The number of patterns and features

In our CAD model the number of patterns allowed to be bigger or equal to the number of features $M \ge N$. Additionally, numerical simulations have shown that the model works well in situation where $M \gg N$. The standard model assumes that $M \le N$.

3.1.5 Vectors zero-mean and normalization conditions

In the standard model vectors v_k are subject to the condition $\sum_k v_k = 0$. Our experiments showed that the model performs well even without satisfying this condition, so we had omitted this in our implementation.

In standard model the following normalization holds:

$$(v_k^T v_k) \equiv \sum_{j=1}^{N} v_{kj}^2 = 1. \quad (6)$$

In (6) term v_k^T denotes the transposed vector of v_k. We tested the system both with and without this normalization and found that the OP generation based on normalized vectors is shorter than non-normalized OP generation by 10 ms. However, the normalization operation alone takes 70 ms, so eventually the non-normalized operations become faster by 60 ms. Obviously, we chose this option as a more efficient method.

3.2 Implementation

The prototype system capable of finding and selecting the room's outer walls and its floor area in the form of ObjectARX application was built. In this subsection the working principles, outputs and user interface of the resulted program are discussed.

The prototype patterns for our case are composed of three values i.e. of the distances between the base point and the room's geometry characteristic points, as shown on Fig. 2.

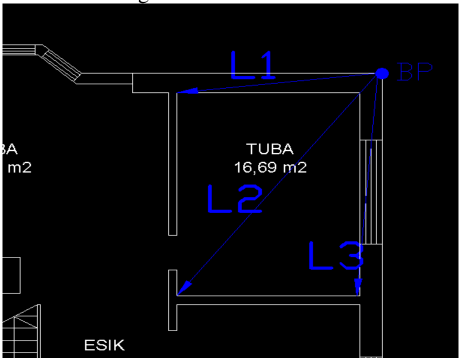


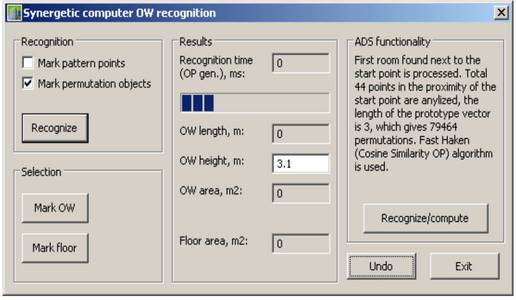
Fig. 2 Representation of the room geometry as prototype vectors' elements in AutoCAD graphics system

We have tested the system with three different prototype patterns with different vectors' lengths (shown as raw vectors):

$$v_0 = (L_{01}, L_{02}, L_{03})$$
$$v_1 = (L_{11}, L_{12}) \qquad (7)$$
$$v_2 = (L_{21}, L_{22}, L_{23}, L_{24}, L_{25}).$$

Actually, the number of prototype patterns may be arbitrary, as this only affects the computational speed of the application. In fact, in real-life applications the number of prototype patterns equals to the number of room types stored. Also, in this prototype build, for the sake of the presentation simplicity, only the first room found is analysed. Other rooms are treatable exactly in the same manner.

For convenient testing and its results' visualization the GUI of the underlying system was created. The user interface of the resulted system is shown on Fig. 3.



Recognition time (OP gen.), ms: 0

OW length, m: 0

OW height, m: 3.1

OW area, m2: 0

Floor area, m2: 0

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

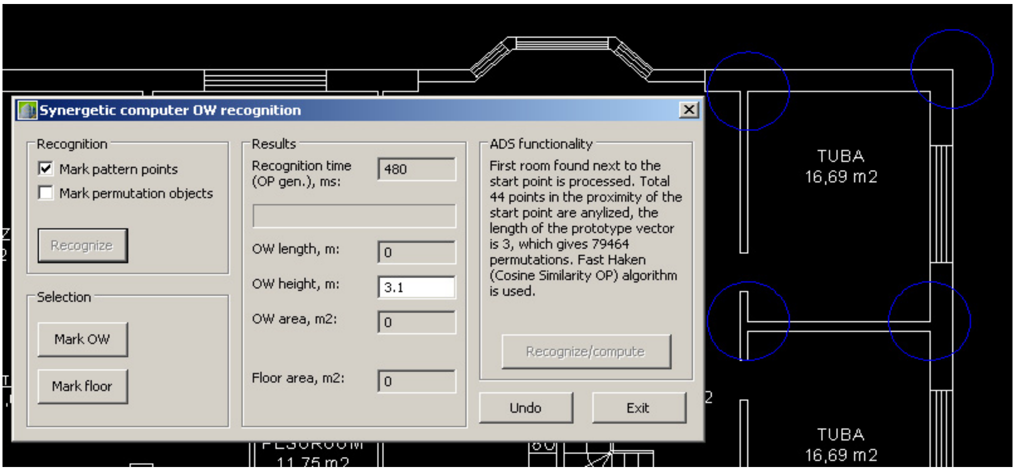
Fig. 3 GUI of the application

The dialog has four major parts: Recognition, Selection, Results and ADS functionality.

In Recognition group by pressing the Recognize button, depending on the check box selected, the system performs pattern recognition and identifies the pattern characteristic points/permutation objects on the graphics screen (Fig. 4, 5). There are total 79464 permutations (test patterns, i.e. $C_k^r = \frac{k!}{(k-r)!} = \frac{44!}{(44-3)!} = 79464$) in this room identification process among which the winning pattern is selected.

As the winning pattern is recognized, it is possible to extract the room's geometry and then use this in automated engineering calculations e.g. in rooms heat loss calculation. For visual identification of the room's geometry two buttons in the Selection section of the dialog box are devoted. That is, Mark OW (Outer Walls) and Mark floor buttons. The output of e.g. Mark OW button action is shown on Fig. 6. In the middle section of the dialog the results of the calculations are shown: the speed of the recognition process, length/area of the outer wall and the floor area of the room.

ADS functionality group has only one button, intended for ADS essence/performance visualization. That is, by pressing this button the system performs automatic recognition and computation of the room geometry (see Fig. 7).



Recognition time (OP gen.), ms: 480

OW length, m: 0

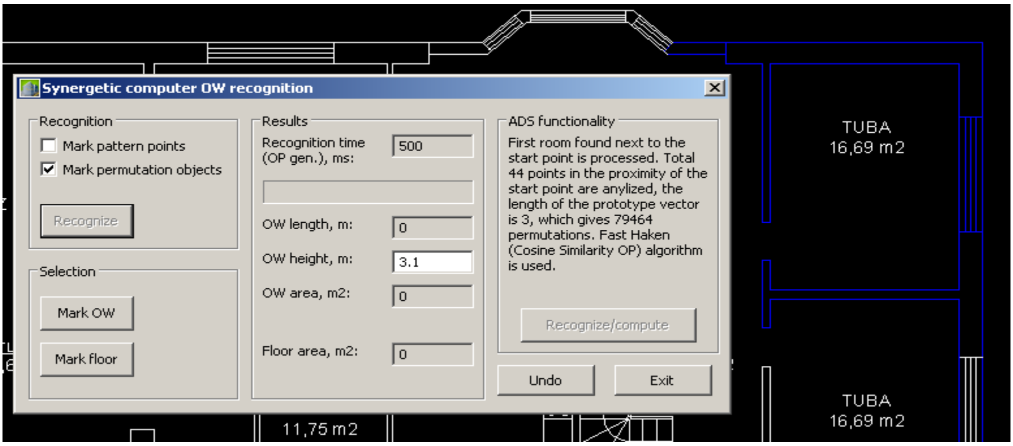
OW height, m: 3.1

OW area, m2: 0

Floor area, m2: 0

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 4 AutoCAD graphics screen – "Mark pattern points" option selected



Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☑ Mark permutation objects

Results

Recognition time (OP gen.), ms: 500

OW length, m: 0

OW height, m: 3.1

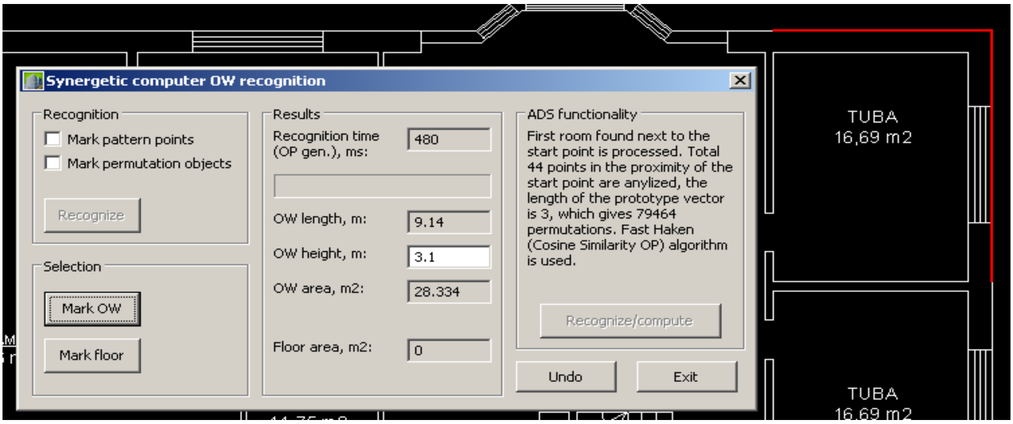
OW area, m2: 0

Floor area, m2: 0

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 5 AutoCAD graphics screen – “Mark permutation objects” option selected



Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☐ Mark permutation objects

Results

Recognition time (OP gen.), ms: 480

OW length, m: 9.14

OW height, m: 3.1

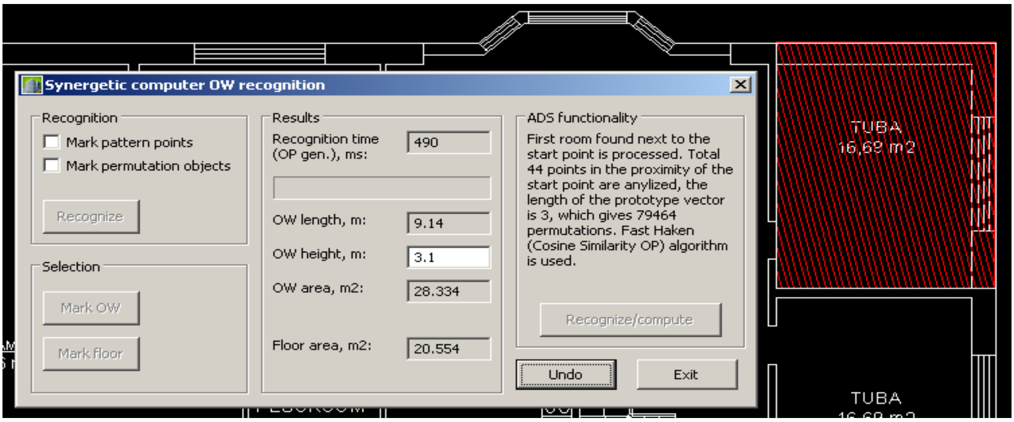
OW area, m2: 28.334

Floor area, m2: 0

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 6 AutoCAD graphics screen – “Mark OW” button pressed



Synergetic computer OW recognition

Recognition

☐ Mark pattern points

☐ Mark permutation objects

Results

Recognition time (OP gen.), ms: 490

OW length, m: 9.14

OW height, m: 3.1

OW area, m2: 28.334

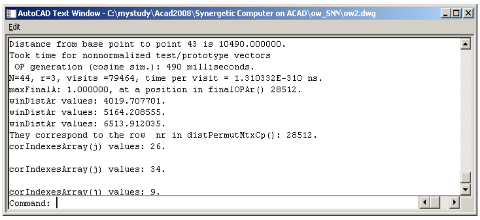
Floor area, m2: 20.554

ADS functionality

First room found next to the start point is processed. Total 44 points in the proximity of the start point are anylized, the length of the prototype vector is 3, which gives 79464 permutations. Fast Haken (Cosine Similarity OP) algorithm is used.

Fig. 7 AutoCAD graphics screen – “Recognize/compute” button pressed

The excerpt of the program debug/control messages is shown on Fig. 8.



```
Distance from base point to point 43 is 10490.000000.
Took time for nonnormalized test/prototype vectors
 OP generation (cosine sim.): 490 milliseconds.
N=44, r=3, visits =79464, time per visit = 1.310332E-310 ns.
maxFinalA: 1.000000, at a position in finalOPAr() 28512.
winDistAr values: 4019.707701.
winDistAr values: 5164.208555.
winDistAr values: 6513.912035.
They correspond to the row  nr in distPermutMtxCp(): 28512.
corIndexesArray(j) values: 26.

corIndexesArray(j) values: 34.

corIndexesArray(j) values: 9.
Command: |
```

Fig. 8 AutoCAD text window with program control messages (excerpt)

3.3 Further research

The intended future research directions are as follows.

1. Improvement and further optimization of the synergetic computer algorithm, introducing “arrow of time”, making it more “self-organizing” [4], [7], [8].
2. Application of this algorithm in modelling of other creative parts of design process in HVAC domain.
3. Making a fully functional real-life software application, implementing complete automation for outer-inner walls recognition mechanism and autonomous calculation of buildings heat losses based solely on vector data representation (i.e. virtually on visual perception).

4 Conclusion

This paper describes the custom synergetic computer algorithm implementation as part of the HVAC ADS on AutoCAD platform. A number of useful modifications to the standard model were committed, tested and successfully implemented in the form of AutoCAD application.

In the course of the experiments the limitations of the classical algorithms of the synergetic computer are identified and surpassed. The model suitable for CAD vector graphics applications is elaborated, tested and realized on AutoCAD platform.

The system implemented has the ability to select and quickly extract building’s room(s) geometry that consequently may be used e.g. in automatic calculation of the room’s heat losses. This functionality is similar to the human visual perception mechanisms and makes the system implemented a candidate for a possible real-life ADS component’s realization (for an overview of the ADS characteristics see e.g. [4]).

References:

[1] D. Loginov, Implementation of the Synergetic Computer Algorithm on AutoCAD Platform, *in Proc. 5th WSEAS World Congress on Applied Computing Conference (ACC '12), University of Algarve, Faro, Portugal, May 2-4, 2012, pp. 112–117.*

[2] H. Haken, *Synergetics. Introduction and advanced topics*, Springer, 2004.

[3] H. Haken, *Synergetic Computers and Cognition. A Top-Down Approach to Neural Nets*, Springer, 2004.

[4] D. Loginov, Possibilities of Modeling the Creative Part of Engineering Design Process using the Synergetic Approach, *International Journal Of Mathematical Models And Methods In Applied Sciences*, Vol.5, No.1, 2011, pp. 95-104.

[5] D. Loginov, Synergetic Modelling – Application Possibilities in Engineering Design, *in Proc. 10th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE '10), Iwate Prefectural University, Japan, 2010, pp. 111–116.*

[6] H. Haken, H. Knyazeva, Arbitrariness in nature: synergetics and evolutionary laws of prohibition, *Journal for General Philosophy of Science*, No.31, 2000, pp. 57-73.

[7] Leo Näpinen, Understanding of the World and the Scientific Paradigm of Self-Organization, *Studia Philosophica*, Vol.IV, No.40, 2004, pp. 156-177.

[8] Leo Näpinen, Philosophical Foundations of Synergetic Modelling, *Proceedings of the Estonian Academy of Sciences. Humanities and Social Sciences,* Vol.4, No.42, 1993, pp. 378-390.

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *CIVIL ENGINEERING*

1. **Heino Mölder**. Cycle of Investigations to Improve the Efficiency and Reliability of Activated Sludge Process in Sewage Treatment Plants. 1992.

2. **Stellian Grabko**. Structure and Properties of Oil-Shale Portland Cement Concrete. 1993.

3. **Kent Arvidsson**. Analysis of Interacting Systems of Shear Walls, Coupled Shear Walls and Frames in Multi-Storey Buildings. 1996.

4. **Andrus Aavik**. Methodical Basis for the Evaluation of Pavement Structural Strength in Estonian Pavement Management System (EPMS). 2003.

5. **Priit Vilba**. Unstiffened Welded Thin-Walled Metal Girder under Uniform Loading. 2003.

6. **Irene Lill.** Evaluation of Labour Management Strategies in Construction. 2004.

7. **Juhan Idnurm**. Discrete Analysis of Cable-Supported Bridges. 2004.

8. **Arvo Iital**. Monitoring of Surface Water Quality in Small Agricultural Watersheds. Methodology and Optimization of monitoring Network. 2005.

9. **Liis Sipelgas**. Application of Satellite Data for Monitoring the Marine Environment. 2006.

10. **Ott Koppel**. Infrastruktuuri arvestus vertikaalselt integreeritud raudtee-ettevõtja korral: hinnakujunduse aspekt (Eesti peamise raudtee-ettevõtja näitel). 2006.

11. **Targo Kalamees**. Hygrothermal Criteria for Design and Simulation of Buildings. 2006.

12. **Raido Puust**. Probabilistic Leak Detection in Pipe Networks Using the SCEM-UA Algorithm. 2007.

13. **Sergei Zub**. Combined Treatment of Sulfate-Rich Molasses Wastewater from Yeast Industry. Technology Optimization. 2007.

14. **Alvina Reihan**. Analysis of Long-Term River Runoff Trends and Climate Change Impact on Water Resources in Estonia. 2008.

15. **Ain Valdmann**. On the Coastal Zone Management of the City of Tallinn under Natural and Anthropogenic Pressure. 2008.

16. **Ira Didenkulova**. Long Wave Dynamics in the Coastal Zone. 2008.

17. **Alvar Toode**. DHW Consumption, Consumption Profiles and Their Influence on Dimensioning of a District Heating Network. 2008.

18. **Annely Kuu**. Biological Diversity of Agricultural Soils in Estonia. 2008.

19. **Andres Tolli**. Hiina konteinerveod läbi Eesti Venemaale ja Hiinasse tagasisaadetavate tühjade konteinerite arvu vähendamise võimalused. 2008.

20. **Heiki Onton**. Investigation of the Causes of Deterioration of Old Reinforced Concrete Constructions and Possibilities of Their Restoration. 2008.

21. **Harri Moora**. Life Cycle Assessment as a Decision Support Tool for System optimisation – the Case of Waste Management in Estonia. 2009.

22. **Andres Kask**. Lithohydrodynamic Processes in the Tallinn Bay Area. 2009.

23. **Loreta Kelpšaitė**. Changing Properties of Wind Waves and Vessel Wakes on the Eastern Coast of the Baltic Sea. 2009.

24. **Dmitry Kurennoy.** Analysis of the Properties of Fast Ferry Wakes in the Context of Coastal Management. 2009.

25. **Egon Kivi**. Structural Behavior of Cable-Stayed Suspension Bridge Structure. 2009.

26. **Madis Ratassepp**. Wave Scattering at Discontinuities in Plates and Pipes. 2010.

27. **Tiia Pedusaar**. Management of Lake Ülemiste, a Drinking Water Reservoir. 2010.

28. **Karin Pachel**. Water Resources, Sustainable Use and Integrated Management in Estonia. 2010.

29. **Andrus Räämet**. Spatio-Temporal Variability of the Baltic Sea Wave Fields. 2010.

30. **Alar Just**. Structural Fire Design of Timber Frame Assemblies Insulated by Glass Wool and Covered by Gypsum Plasterboards. 2010.

31. **Toomas Liiv**. Experimental Analysis of Boundary Layer Dynamics in Plunging Breaking Wave. 2011.

32. **Martti Kiisa**. Discrete Analysis of Single-Pylon Suspension Bridges. 2011.

33. **Ivar Annus**. Development of Accelerating Pipe Flow Starting from Rest. 2011.

34. **Emlyn D. Q. Witt**. Risk Transfer and Construction Project Delivery Efficiency – Implications for Public Private Partnerships. 2012.

35. **Oxana Kurkina**. Nonlinear Dynamics of Internal Gravity Waves in Shallow Seas. 2012.

36. **Allan Hani**. Investigation of Energy Efficiency in Buildings and HVAC Systems. 2012.

37. **Tiina Hain**. Characteristics of Portland Cements for Sulfate and Weather Resistant Concrete. 2012.