TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 493 • Publication 493

Béla Pátkai

An Integrated Methodology for Modelling Complex Adaptive Production Networks

Tampere 2004

Tampereen teknillinen yliopisto. Julkaisu 493
Tampere University of Technology. Publication 493

Béla Pátkai

An Integrated Methodology for Modelling Complex Adaptive Production Networks

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Hermitec Building, Auditorium B116, at Tampere University of Technology, on the 13th of September 2004, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2004

ISBN 952-15-1238-5 (printed)
ISBN 978-952-15-1842-3 (PDF)
ISSN 1459-2045

Abstract

Pátkai, Béla: *An Integrated Methodology for Modelling Complex Adaptive Production Networks*

Tampere University of Technology
Department of Mechanical Engineering
Institute of Production Engineering
Production System Design Laboratory
Finland, 2004

Keywords: agentbased modelling and simulation, complex adaptive systems, dissipative structures, evolutionary computation, methodology development, production networks.

Adaptation and learning are the most crucial skills in the survival of any complex system – the former one emphasizing the ability to perform structural reorganization and the latter one the use of previously available information – to reflect on the endlessly changing environment the particular system is embedded in. Humans are such complex systems and also manmade ones that humans manage by the aid of cooperation, science and the multitude of automated tools such as computers, robots, vehicles and their combinations. The survival fitness of individuals, organizations, societies and mankind itself depends on the successful management of the adaptation and learning process that often involves the changing of the environment.

In this interplay between man and nature it is crucial to gather useful knowledge of explanatory and predictive power in the – Aristotelian – form of science and metaphors. In addition to these, computers have provided a third form or language for knowledge gathering and representation since the middle of the XXth century. The success of a system of knowledge – a theory – largely depends on the integrated application of these knowledge acquisition methods and is measured by the fitness and survival of its users.

Since scientific methods are typically limited in scope, metaphors are used to bridge the gaps and connect seemingly distinct fields.

The general aim of this thesis is to contribute to the area of complex adaptive systems research − in particular complex adaptive production networks − by integrating scientific, metaphoric and computational knowledge in a *methodology* to complement more traditional and specialized approaches such as mathematical equation based modelling, computer simulation techniques and management methods. Building synthetic, agentbased simulation models is only part of this endeavor, providing a media for repeatable experiments that point to various scenarios leading to chaotic behavior, inflection points and bifurcations.

Since research in the area of agentbased modelling and complex adaptive systems often concentrates on building software and running simulations, the methodology developed in this work is mainly concerned about the bigger picture that includes not only a basic software library but a scientific and philosophical framework that integrates knowledge gathering techniques and languages and helps to navigate in the challenging area of complex systems by exploring limitations and opportunities systematically.

Acknowledgments

The preparation of this thesis is based on the research work carried out between the years of 2000 and 2004 by the financial support of Suomen Akatemia (Academy of Finland) in projects *ModNet* – in the framework of the TUKEVA programme, directed by Dr. Kalle Hakalehto – (proj.no. 772318) 2000-2003, and *AGITEC* (proj.no. 5104951) in 2004. In addition to this the Institute of Production Engineering – my host institute – has provided high quality infrastructure and support personnel for which I am very grateful.

I am most indebted to my supervisor, Prof. Seppo Torvinen, director of the PSD laboratory, who has encouraged and supported me in many different ways to start and complete this dissertation.

In the above mentioned *ModNet* project Mr. Janne Keskinarkaus and Mr. Zoltán Szaniszló were my cheerful workmates, who have contributed to the pleasant work environment and also to the development of the *ModNet* simulation system.

In the PSD laboratory many people have helped me at different times, and I had fruitful conversations with all of them. I would especially like to thank Mr. Petri Huhtala who has been involved in many of my efforts in various ways.

In my home country, Hungary, I benefited from the professional and friendly cooperation with Prof. Imre Rudas, rector of Budapest Tech, who has also advised me in my minor subject studies. For the last 10 years, Prof. József Kázmér Tar, also at Budapest Tech, has been an advisor and good friend of mine. His knowledge and enthusiasm for the natural sciences, literature and philosophy has revolutionarized my thinking and provided great intellectual and friendly support whenever I needed it.

During the preliminary assessment process, the comments and corrections of Prof. Graeme Britton, NTU Singapore, and Prof. Pietro Terna, University of Turin, Italy, have been very useful and encouraging. Based on their reports I could improve the thesis significantly and gained numerous ideas for further research.

I would like to express my gratitude to my parents, Eszter and Béla, and my sisters, Csilla and Boglárka for their love and support. The long journey that has lead me to Finland and to postgraduate studies started with the first steps along my family and continued with educational institutions and devoted teachers in Egyházaskozár, Bonyhád, Nyíregyháza, Budapest, Nottingham and finally Tampere. During this journey I got to know many people from many different cultures and were fortunate enough to become their friend.

Finally, I would like to thank the loving support of my fiancée, Ms. Nina Honga, without whom my last two years would not have been complete and my motivation would have faded away. Her love and friendship made me a happier worker and her example taught me a lot about how to be more organized and focused. I would also like to thank her family: Auli, Göran, Anne-Marja, Susanna, Janne and Saana for their friendship and regular inquiries about my professional progress.

Béla Pátkai
bela.patkai@tut.fi
Tampere, August 2004

Table of Contents

List of Figures

List of Tables

List of Symbols

F	Force
H	Hamiltonian
J	Thermodynamic flows
k_B	Boltzmann constant
N	Number of molecules in a gas chamber
p	In *Eqn.3.* generalized momentum, in thermodynamic equations pressure, in section 4.6 about computation it represents a prime number
P	Total entropy production
q	Generalized coordinate
Q	Heat
S	Thermodynamic entropy
dS_e	Thermodynamic entropy change due to exchange of energy and matter
dS_i	Thermodynamic entropy change due to the irreversibility of processes
σ	Entropy production
T	Temperature
U	Internal energy in the thermodynamics section and universal computer in the computation section
V	Volume
W	Thermodynamic probability

Chapter 1 Introduction

"Science and technology will shift from a past emphasis on motion, force, and energy to communication, organization, programming and control".
John von Neumann (1950)

"I think the next century will be the century of complexity."
Stephen Hawking (1999)

A theory in science compresses into a brief law or set of equations, the regularities in a vast, even indefinitely large body of data.
Murray Gell-Mann[1]

Advanced technology, telecommunications and logistics are the most typical factors that make the world change faster, and the co-evolution of these and human cognition constitute the foundation for the frequently mentioned phenomenon of globalization. Such a speedup makes us face an accelerating adaptation process, and puts more stress (selective pressure) on the individual and nature, as well as on manmade systems, i.e. infrastructure, organizations, related paradigms, decision making practices, etc. Systems emerging from this highly competitive, evolving environment typically have more degrees of freedom and are more open, but the path of their evolution inherently involves the merciless extinction of the less fit ones – individuals, organizations, ideas – something that all of them would like to avoid. The deep desire of humans to make the world more predictable led to the development of science, the tool for a more understandable and controllable world that is less vulnerable to the vagary of chance.

The incremental – and occasionally revolutionary – development of formal theories have significantly shaped our thinking, and – in a co-evolutionary manner – the way of thinking had generated either a holistic or a disintegrated science, the former in the beginnings and the latter in the XIXth century, when the representatives of the separate disciplines thought their subject is a well-defined one, and its details only need time to be worked out

[1] Murray Gell-Mann, Nobel laurate in physics, The 6th Annual Stanislaw Ulam Memorial Lecture Series, Santa Fe Institute, *The Regular and the Random*, September 22, 1999.

completely. Despite this rather optimistic endeavor multi− and cross−disciplinary research evolved in the XXth century, and the walls between biology and mathematics, physics and chemistry started to erode. Scientists started to emphasize again, that nature is unified, not disjointed, and so should be the science we develop. For a long time it seemed that the applications of science were lagging behind this integration, as engineering disciplines had a firm foundation on *Classical* (or *Newtonian*) *Mechanics* [29], based on the the so-called *reductionist* way of thinking [43]: the idea was to *reduce* all problems to the level of understandable, tractable, mechanistic equations.

However, the aforementioned changes in the − scientific − world dislocated the focus of interest from particulars to *systems*, and in the first half of the XXth century a number of new ideas emerged related to systemic thinking, e.g. Cybernetics [52][2][17], General System Theory [9][50], Artificial Life [30], followed by Complexity Science [36] (called in short *Complexity,* also called Plectics, i.e. the "science of the simple and the complex" [16]). This latter one is also advocated by two eminent scientists − both Nobel laureates − Ilya Prigogine, who carried out and inspired a lot of work related to nonequilibrium thermodynamics, and Murray Gell-Mann, the discoverer of quarks and co-founder of the Santa Fe Institute that is generally considered one of the leading centers of complexity research.

The related paradigms, theories, tools, all embraced by or closely related to Complexity Science − including Soft Computing techniques such as Artificial Neural Networks, Fuzzy Sets, Genetic Algorithms, Evolution Strategies, Genetic Programming, Population Based Incremental Learning − have swiftly started to diffuse into engineering practice, and nowadays they are commonly used. Also − demonstrating the change of thinking about the strict separation of disciplines − dedicated university courses are trying to bridge the gap between seemingly distant principles. Mechanical engineers apply the principles of simulated evolution for aircraft wing profile optimization, electrical engineers run genetic programming algorithms to design patentable electrical circuits, logistic experts use computer simulations of ant colonies, control engineers investigate chemical reactions [47], computer scientists make software that evolves programs without expert intervention and materials scientists make use of the far-from-equilibrium conditions to treat materials [36].

Throughout this thesis the concept of *complex adaptive systems* (one of the central paradigms in complexity science) is going to be used often and in different contexts. A more detailed discussion about complex adaptive systems is found in Chapter 3, but to eliminate the need for forward referencing a summary is provided to show the main characteristics of such systems. A complex adaptive system is characterized by [16][36][27][13]:

1. its ability to find regularities in the data present in its environment, and compress it in "schemata"
2. react to changes in the environment in order to maintain its boundaries, increase its fitness and "survive"
3. learn/adapt through testing and updating its schema (i.e. compressed experience, knowledge)
4. exhibit chaotic behavior and emergence of complex phenomena from simple interactions.

Typical complex adaptive systems are our mind, science, businesses, organizations, supply chains, cities, nations, economies, evolution of species, thermodynamic systems, etc. In case of a business firm the schema includes business practices, in a society they are laws, customs and myths, in science they are theories. These schemata are constantly tested against the environment – that consists of other complex adaptive systems – and depending on their success they remain or get replaced by other schemata. The systems themselves are the "unfolding" of these schemata, just as the DNA unfolds in the environment and "builds" a human body; e.g. in case of a business firm external changes make the firm react – according to rules and information included in its schema. In case the reaction does not achieve the expected result, part of the schema may be replaced and tested against the environment, as long as the conditions of survival are reestablished. In extreme conditions, when the selection pressure is very high, the adaptation process may fail, and the firm may leave the market – this is evolution at its routine, survival or extinction.

The central theme of this thesis is the consideration of production networks as complex adaptive systems and the development of a methodology that acknowledges the complexity of the problem and integrates different tools and methods of inquiry.

There are several influences of high potential in today's world that motivate the way of thinking lurking in this thesis:

1. The last decades' shift from reductionism to systemic thinking in science and engineering
2. The possibility for quick and relatively cheap physical transportation of goods and people around the world
3. The practically unlimited and unrestricted flow of information through the Internet
4. The possibility for computational modelling of systems and processes
5. The high computational power available for a relatively low price and the expected breakthrough in the next few decades (nanotube semiconductors, molecular transistors, analogue microprocessors, quantum computation, etc.) and the challenge they represent (i.e. are we ready to use them?)

It is also useful to think about this thesis as a *component* of a complex adaptive system. The schema includes the ideas, concepts, theories, claims, theses, postulates and software, all of these unfolding in the manuscript, the mind of its writer and a software library. During its production and years of research many competing ideas, software components were tested against the "real world", such as the opinion of colleagues, supervisors, friends, paper reviewers, conference attendants, information in books and papers, computer simulations. Many of the ideas have not survived but new ones were born – all these part of the adaptation/evolution process. Such a work – along with the knowledge of the researcher and the continuously developed software tools – is nothing like a finished, static entity, but – true to its subject – a snapshot at a certain point of time, aiming at survival while interacting with its environment – a number of interacting complex adaptive systems including human minds and organizations.

1.1 Structure of the Thesis

The thesis consists of eight chapters, starting with an introduction and ending with conclusions plus references. Its structure can be seen in Fig. 1.1 below.

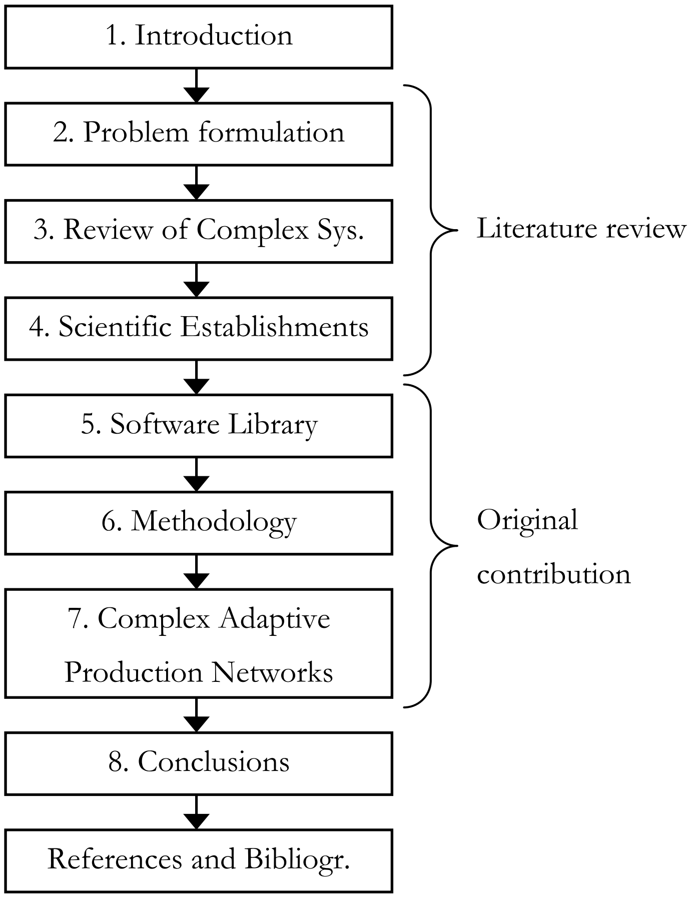

Figure 1-1 The structure of the thesis

Following the introduction chapter 2 is about the problem formulation where it is shown how the simple and straightforward formulation of problems becomes a system or network of interconnected problems − forming a complex system. In chapter 3 a review of complex systems and some related topics are discussed including complex adaptive systems, agentbased modelling, enterprise modelling, some economical and organizational issues, and enterprise modelling software along with some examples of commercial products addressing related problems. The scientific establishments of chapter 4 provide some powerful metaphors and analogies that will be used in various ways throughout the thesis. Chapters 3 and 4 constitute the literature review and background research of the thesis and include a considerable amount of originality in its selection, form presentation and interpretation.

The original contributions are included in three chapters, 5-7. Chapter 5 describes the ModNet production network modelling toolbox that was designed by the author, but its coding is mainly attributed to a research project team member as described in the Foreword. This toolbox and the synthetically developed methodology in chapter 6 provide the tools for complex systems modelling for potential users, and points out various limitations of such a methodology.

Chapter 7 builds on the literature review and the original contributions, too, and introduces a new concept of *complex adaptive production networks*. In harmony with the methodology and the scientific background presented earlier, this chapter draws analogies between dissipative structures from non−equilibrium thermodynamics, evolution and production networks. This chapter includes ideas that are not fully worked out because of the natural temporal limitations and the nature of this thesis. However, it provides a good starting point for others who will apply and refine the methodology.

Chapter 8 is a summary that concludes and analyses the achievements and includes a section on further plans. The last chapter lists the references in alphabetic order.

1.2 The Content of the Thesis

For formal requirements a hypothesis is provided in the next subsection, followed by a list of objectives and a summary of original contributions.

1.2.1 Hypothesis and Objectives

The methodology and the accompanying software tool presented in this thesis are designed and developed to contribute to the area of complex systems and agentbased modelling research. This contribution enables other researchers to carry out systematic modelling of complex systems, share information about it and discuss it in a coherent, integrating framework that includes and enforces a terminology, a viewpoint and practices. The use of this framework and methodology by other researchers has the potential to eventually lead to an accumulation of modelling knowledge useful in commercial computer advanced modelling products.

The detailed objectives are the following:

1. Establish the epistemological background for the research, including the critical view of scientific and rhetoric knowledge
2. Define a new problem class based on the observations made related to complex adaptive systems and production networks
3. Survey state-of-the-art methods and tools related to the identified problem, formulate this from an original point of view to make it a minor original contribution
4. Identify and explore scientific tools showing potential for successful application either as exact methods or as applicable analogies
5. Formulate an integrated methodology for buttom−up complex system modelling
6. Identify the limitations of the methodology with special attention to computational and philosophical issues
7. Design an agentbased modelling software library that enables researchers to run comparable and reproducible simulations of distributed production systems with different distribution of control
8. Describe an analogy transfer in detail, showing how knowledge is transferred through a scientific metaphor into a model

1.2.2 Original Contributions

Previous work in the area of complex systems, agentbased modelling and systems modelling has consumed lots of effort in the field of software development, conceptual clarification, phenomenological studies and modelling theory. However, mainly due to the novelty and the challenging limitations of the field, no standard methodology has been developed, especially none that would place agentbased modelling of complex systems in an integrated scientific, philosophical and software framework.

This work claims to have made the following original contributions to this problem:

1. A literature review reflecting a special view of the subject
2. A new problem definition of production networks that exhibit complex behavior
3. The synthetic development of an integrating modelling methodology
4. Drawing a new analogy between dissipative structures and production networks

1.3 Research Method

The definition of the problem, exploration and exploitation of potential solutions and new developments all are addressed in this thesis in harmony with the views expressed in the beginning of this chapter about complex adaptive systems and their "soft" formulation. A more conventional method would have been:

- Definition of a problem mathematically
- Exploration of available results (with quantitative measures of success)
- Proposal of a new method or algorithm
- Quantitative comparisons

However, in the problem domain of this thesis – as it will become more apparent in the next chapters – the classical engineering approach fails, because complexity and complex systems cannot be controlled by classical methods, despite of the research effort consumed in reducing these problems to classical ones. In addition to this serious challenge the problem is considered an open system – similar to the ideas of non–equilibrium thermodynamics – and admit, that sharp boundaries of the problem cannot be drawn. However, many things *can* be done, and therefore a wide range of topics are explored and integrated in the next chapters in a true systemic manner.

Chapter 2 Problem Formulation

"Natural science does not simply describe and explain nature;
it is part of the interplay between nature and ourselves; it
describes nature as exposed to our method of questioning."
Werner Heisenberg[2]

The correct, clear-cut definition of the problem, aims, goals and ideals is evidently crucial in any thesis. However, as it was have suggested previously, it is plausible from a pragmatic point of view to let the problem be what it is in reality: fuzzy. Following the two main points made in the previous chapter it is assumed that the problems associated with production organizations are large and complex enough to consider them complex adaptive systems[3]. In addition to this let's assume that the problems we are facing is not a single problem, but a set or system of related and − to a varying degree − coupled problems (forming a CAS), including standard mathematical problems and issues more "blurry", related to the "human factor" or simply too difficult to formulate because of the nature of the problem or because the lack of knowledge available. In the next section it is shown how industrial problems formulated in an exact manner shape up to form a complex system. In section 2.2 some of the typical optimization problems are listed that appear frequently in a production environment, and in section 2.3 problems related to supply chains/production networks are summarized, that require a combination of scientific and rhetoric knowledge even for their definition. This latter one is claimed in this thesis to be the realistic approach for problem formulation and methodology development, therefore at the end of the chapter a synthesis of the problem is done, and is explained in a figure to relate it to other problems and methods.

---

[2] Werner Heisenberg (1901-1976), physisist, one of the founders of quantum mechanics.
[3] A certain amount of forward referencing to concepts was unavoidable in this work, however, it is kept to a minimum.

2.1 From Problems to a System of Problems

In *Fig. 2.1* the concept of virtual engineering is presented, summarized by Wörn in [54]. The concept emphasizes that a manufacturing environment – from initial plant design to end product – is very strongly computerized and is full of optimization problems, especially combinatorial ones. In the same time these optimization problems are interconnected, coupled, and form a complex system.

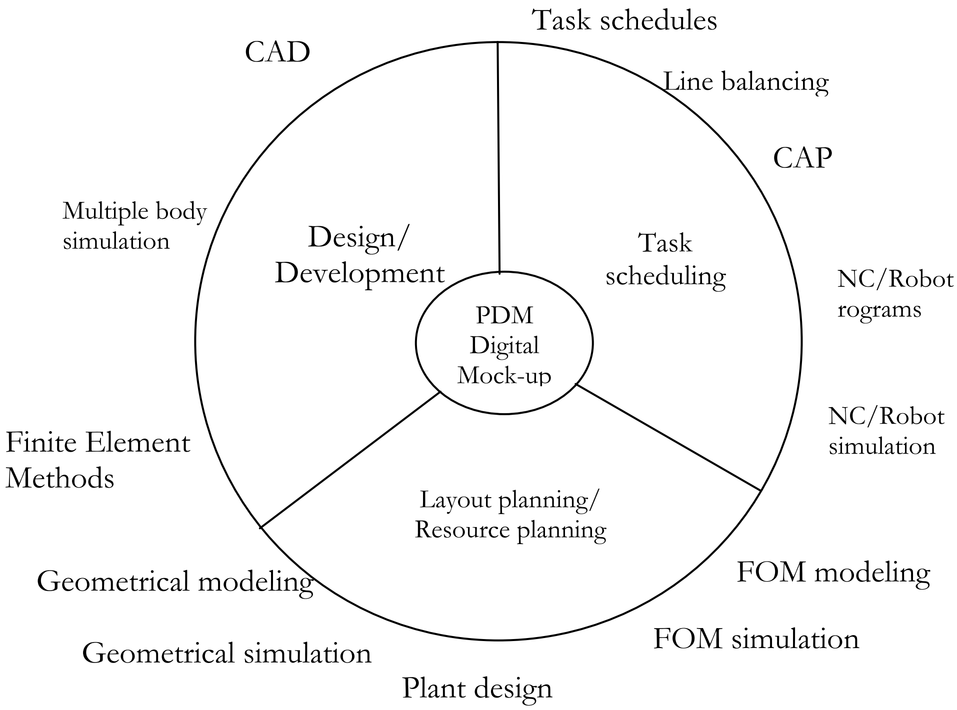

Figure 2-1 The concept of Virtual Engineering

In the context of supply chains/production networks it is customary to look at problems as optimization problems, in fact it is an “attitude” – also criticized – towards problem solving [23].

2.2 Complex Systems and Production Networks

Complex adaptive systems have different formulations, two important ones being developed by John Holland and Murray Gell-Mann [16]. The difference between them is that they put the emphasis on different points. Holland starts with the *internal model -> adaptive agent -> complex adaptive system,* while Gell-Mann starts with *schemata -> complex adaptive system -> a set of complex adaptive systems*. Gell-Mann’s formulation is very suitable for application in this thesis, since it reflects the view that the problem is a problem set of a "loose aggregation" of complex adaptive systems which interact [16][13][1][24][27][36].

The problem with such interactions is that changing one of the complex adaptive systems in the set changes the interaction of the parts. Kauffman describes it as the "patch procedure" (see it schematically in *Fig. 2.2*): *"The basic idea of the patch procedure is simple: take a hard, conflict laden task in which many parts interact, and divide it into a quilt of non-overlapping patches. Try to optimize within each patch. As this occurs, the couplings between parts in two patches across patch boundaries will mean that finding a "good" solution in one patch will change the problem to be solved by the parts in adjacent patches. Since changes in each patch will alter the problems confronted by neighboring patches, and the adaptive moves by those patches in turn will alter the problem faced by yet other patches, the system is just like our model co-evolving ecosystems."* ([24] pp. 252–3).

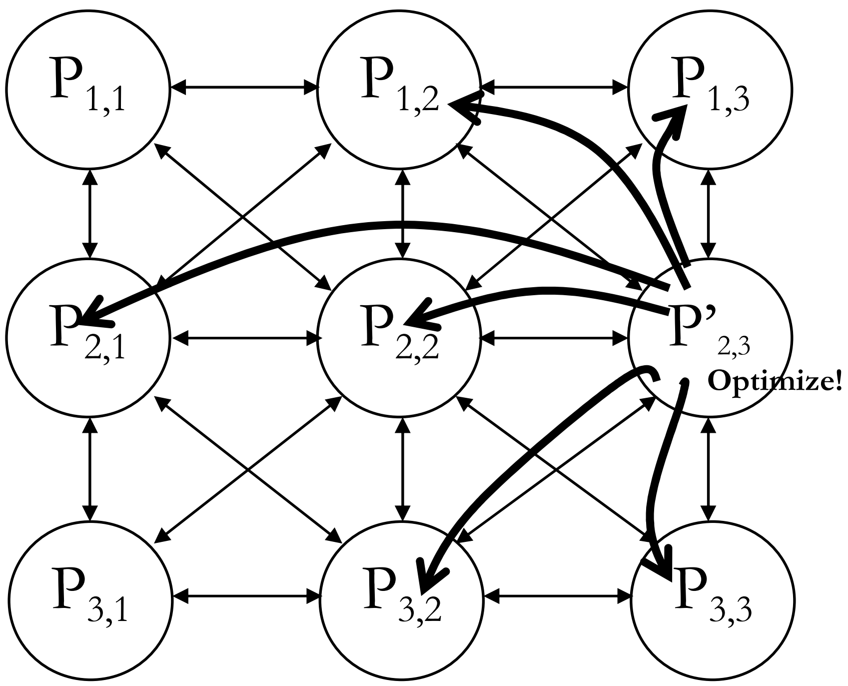

Figure 2-2 The patch procedure on the partitioned problem of coupled subproblems

This *illustrates* our concern that such a systemic mechanism causes combinatorial explosion in its environment, since small changes in problem size and chaotic changes might cause the entire reconfiguration of the system, if not its collapse. This single remark about coupling in complex adaptive systems itself makes us change our focus and direct us to abstract system theories and related methods.

Instead of looking at such problem categories separately, it seems more suitable to handle them as part of a dual problem set including optimization problem instances as subproblems. As Ingalls points out in [23], optimization problems in supply chains often end up in simplified form of mathematical programming (linear, dynamic, mixed-integer), and miss important points such as:

- Demand forecasting details
- Earnings estimates and their feedback
- Variance (in various contexts)
- Strategic planning
- Supplier reliability

This clearly indicates that reducing the global problem to optimization issues avoids complexity and can have serious consequences, like missing the point of organizational modelling.

2.2.1 Supply Chains

Since this thesis is concerned about complex systems, and is applied to problems related to production networks supply chains are briefly introduced in this section and its supersets in the next two sections.

Supply chains have various generally accepted definitions but all are very similar. Listing a few:

- *A supply chain is a network of autonomous or semi-autonomous business entities collectively responsible for procurement, manufacturing, and distribution activities associated with one or more families of related products* [99].

- *A supply chain is a network of facilities that procure raw materials, transform them into intermediate goods and then final products, and deliver the products to customers through a distribution system* [87].

- *A supply chain is a network of facilities and distribution options that performs the functions of procurement of materials, transformation of these materials into intermediate and finished products, and the distribution of these finished products to customers* [15].

Other terms for supply chains are [44]:

- *demand network*
- *value network*
- *demand chain*
- *commodity chain*
- *production chain*
- *activities chain*
- *value chain*
- *value web*
- *pipeline*

All these terms emphasize a different aspect or characteristic of supply chains.

2.2.2 Production Networks

The concept of *production network* is one level higher in the organizational hierarchy than supply chains. According to the definition of Prof. Sturgeon *"a production network is a set of two or more value (or supply-) chains that share at least one actor."* [44] This is an evident extension of the supply chain paradigm, and a deverticalization, too.

In comparing supply *chains* and production *networks* it is important to see that the word *chain* implies a vertical orientation and *network* implies also horizontal interconnection.

Production networks emerged due to global changes favoring the construction of larger and larger production organizations, often spanning through continents. One of the main reasons for this increase in size and complexity is cheap and fast transportation available worldwide.

Depending on mainly cultural factors, it is posible to distinguish between *captive, relational* or *turn-key* type production networks [7], though other finer classifications are possible.

- A *captive* type of production network is dominated by an organization larger than the others involved.
- The *relational* type is the most typical one in Europe, that is largely built on spatial and social links, building on trustful relationships in the business environment.
- *Turn-key* type production networks are usually represented by American organizations, that emphasize flexibility and innovation the most.

The evolution of production networks starts with isolated production systems and continues with supply chains. The second evolutionary step resembles production networks with even more degrees of freedom, including possibilities for virtual networking.

2.2.3 Virtual Production Networks

The highly developed state of the information infrastructure and logistics enables a dynamic networking of production, i.e. the formation of virtual production networks. Companies developing such a virtual network usually make use of their know-how and *link innovative but deverticalized lead firms with sets of highly functional suppliers* [44]. The lead firms in such a network typically provide the innovative part and marketing power and the rest of the functions such as manufacturing, process engineering, assembly, packaging, distribution are taken care of by supplier[4]s [44][45][46]. There are inevitably dangers in this production model, since a significant part of the physical process lies in the hands of the subcontractors, who easily gain expertise, get a grip on the rest of the process and get loose to become dominant lead firms themselves.

2.2.4 Social Capital and Production Networks

An important aspect of production networks is their relation to social capital[42]. The previous section already pointed out that cultural and historical motivation has a significant effect on production networks. In the same way production networks have an effect on culture and politics – this way the two coevolve and therefore influences any work carried out in the field of enterprise modelling. Though in this work it's not possible to go into more details about this issue, it is important to note that in case of evolutionary exploration

[4] The great danger of this strategy is apparent in the Japan-USA virtual networking, where the suppliers have learned the lessons of the american lead firms, and became functional without them.

or optimization of network structure it is not useful to develop organizations that are not viable because issues related to social capital have been ignored. Some social capital examples that need to be taken into account include:

1. In certain countries subcontracting is seriously constrained by personal relationships between company managers.
2. A factory cannot be shut down in a small community because of political reasons.
3. A reorganization of an enterprise has to be done so as to preserve good spirit amongst workers.
4. The headquarters of an enterprise cannot be moved to a more favorable country because homeland buyers would penalize the change by changing purchasing habits.
5. The introduction or removal of environment friendly products on the market changes the acceptance of other actions.
6. Information sharing practices between cooperating/competing companies influence strategic decisions.

2.2.5 Typical Problems Occurring in Supply Chains/Production Networks

In the three previous subsections we have seen some major organizational configurations that are at the centre of our interest.

In the following some typical problems arising in supply chains and production networks are listed[5]:

1. plants are usually dedicated to one product
2. typically not much excess capacity available
3. Forecasting and information sharing/communication problems
4. forecast date has high error margin
5. plant capacity is imprecise
6. coordinating distributed production is an increasingly complex task

[5] This list is a collection of comments and documented problems from real-world supply chains and production networks, collected from conference and journal papers and websites.

7. government regulations may be an issue in case of large network (tax and duties, trade restrictions, wages, requirements)
8. handling of imprecise data
9. handling of long-term planning
10. how to explore and exploit the whole solution space
11. how to handle multiple objectives
12. how to rank solutions
13. distribute decision making across the system
14. what should be the systems architecture be like
15. plan the capacity and location of new plants or suppliers
16. data fuzziness all over the system
17. out-of-stocks cause 4-5% loss to manufacturer, not including other intended purchases at time of the visit (Source: Retailer Operating Data, Prism Partner Store Audits, Coca Cola Retail Council Independent Study, 1996)manufacturer and retailer forecasts are not integrated
    - sales history is used as a predictor
    - forecasts do not include future programs
    - manufacturers are not building to retailer and consumer demand
    - forecasting of promotional, seasonal and new items remain a critical issue
19. bullwhip effect (i.e. supply chains are chaotic) causing huge oscillations in case the individual units are trying to solve their own problems and:
    - overreact to backlogs
    - poor communication in the supply chain
    - poor coordination
    - variable delay times
    - customers order more unnecessarily to keep their inventories safe

- partners order too small amounts to keep their inventories low

2.3 Problem Definition for this Thesis

In this chapter we have seen some aspects of the problems related to complex systems, systems of problems and production networks that resemble complex systems. The problems related to production networks are vast in number, they embrace scientific, managerial, financial, political, cultural issues.

The methodology developed in chapter 6 is only concerned about modelling the complexity aspect of these problems conceptually and computationally. To provide more focus, a figure from the related literature is used in Fig. 2.3 and extended with new boxes and explanatory notes. The summary Fig. 2.3 is a taken from the area of knowledge engineering, and is quite self-explanatory [10]. It shows that Knowledge-based Expert Systems (KBES) can be viewed at the knowledge and at the computational level.

In Fig.2.4 agentbased modelling and agent-oriented representation (programming) had been added to the figure because the implementation paradigm of the methodology is agentbased, and the corresponding software engineering methodology is agent-oriented, therefore it is depicted as a shaded text box.

The four numbers at different points of the figure provide a good summary of what is expected of the methodology in chapter 6:

1. At the knowledge and problem level
   - Where are the boundaries of the problem (i.e. what to model)?
2. At the border between the knowledge– and computational level (i.e. the mapping of knowledge on a computational structure):
   - How to make the mapping (methodology)?
   - What is lost in the mapping (what type of mapping is it)?
   - What can be represented by computation and what can't be?
   - When and why use a certain representation?

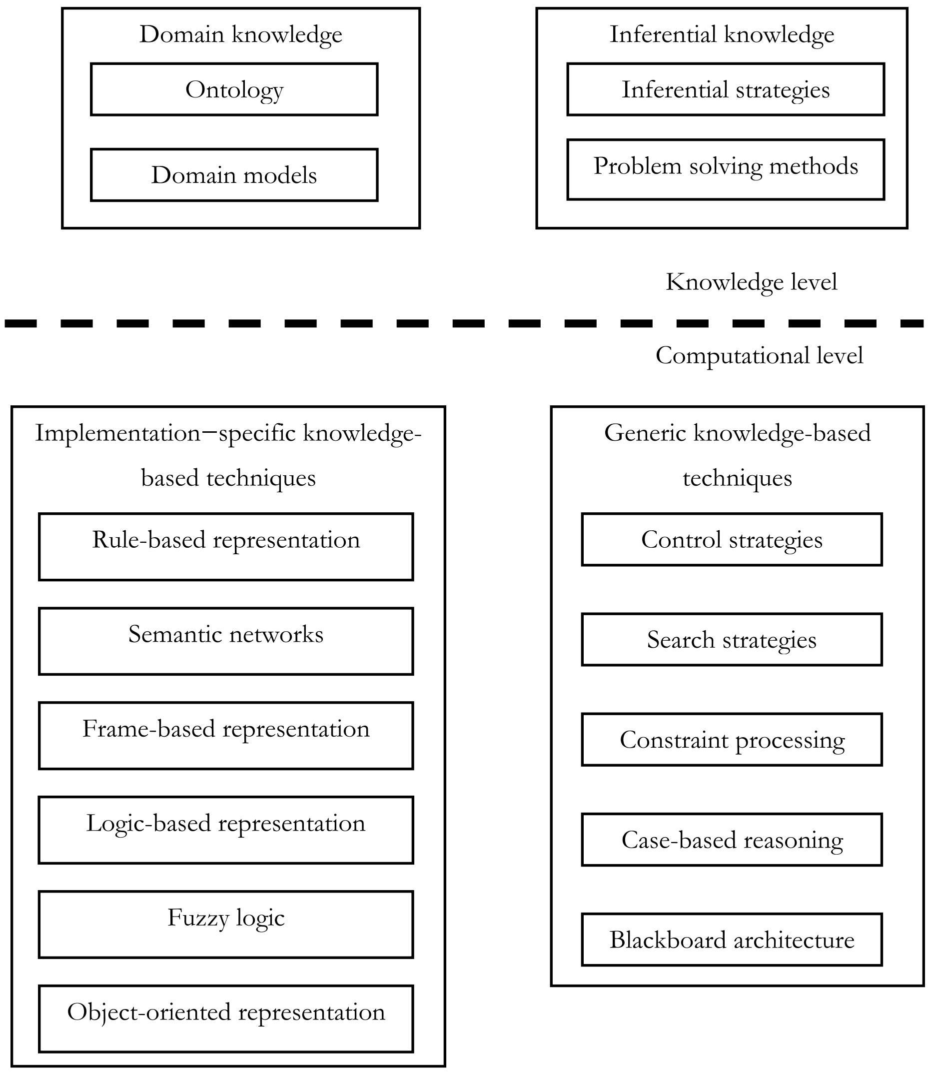

Figure 2-3 Knowledge Engineering (figure reproduced from Britton et al. [10])

3. At the generic techniques:
   - How to formulate the model?
4. At the implementation specific part:
   - How to represent the model?
   - How to validate and verify it?

These questions are expected to be answered by the methodology. Summing up the problem: *our aim is to develop a methodology by synthesis that supports the effective, correct and well grounded mapping of human– and domain knowledge of complex systems into agentbased models by integrating scientific– and rhetoric knowledge by the use of analogies and computer experiments.*

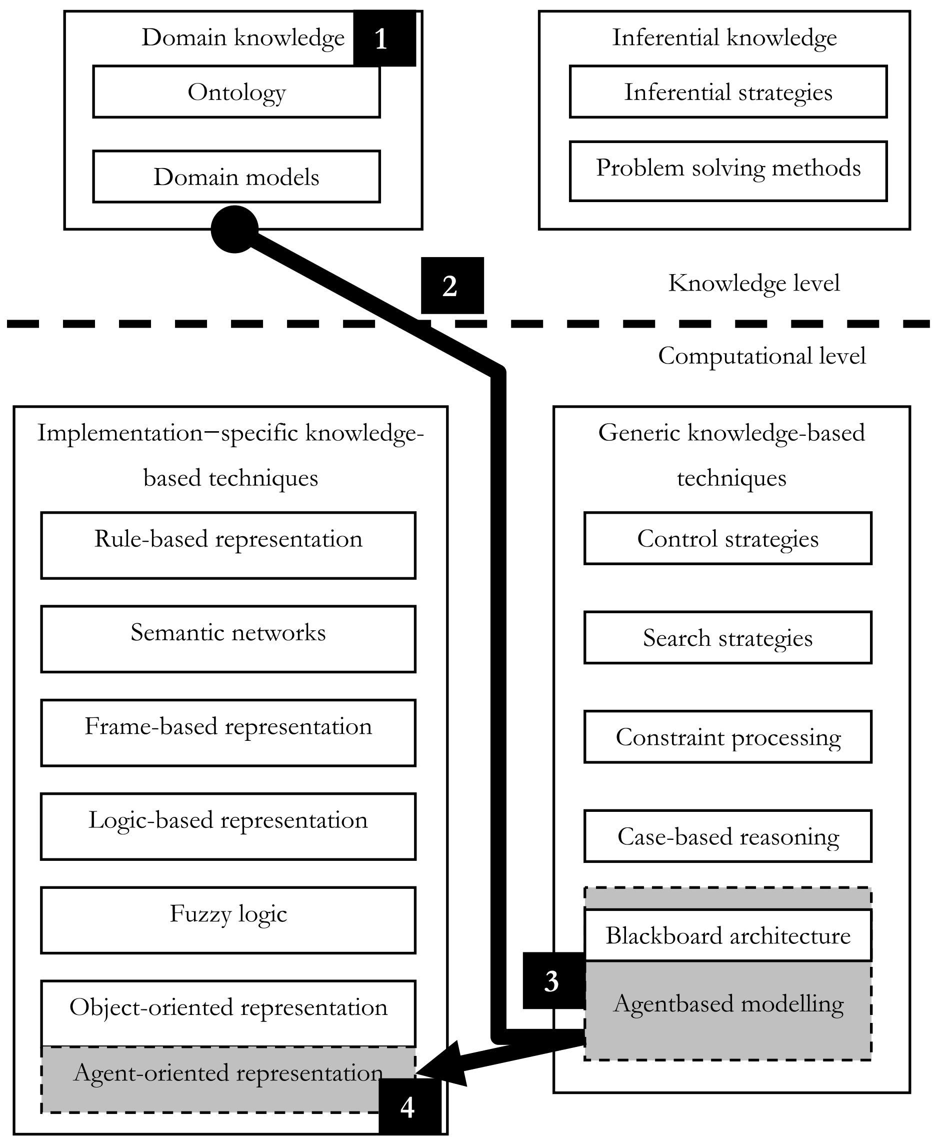

Figure 2-4 Explanation of the role of the developed methodology

Chapter 3 Review of Complex Systems

"Scientists must use the simplest means of arriving at their results and exclude everything not perceived by the senses."
Ernst Mach

"Things have to be as simple as possible, but not simpler."
Albert Einstein

In accordance with the introductory chapters we can see that the tools and theories associated with supply chains and production networks can be vast and their choice is strongly influenced by corporate culture, geographical location, management, owners, operating conditions, competition, etc.

Complex systems are omnipresent, and their description is found in chemistry, physics and engineering principles, however, the field is lacking tools mainly because of the nature of the problem. Maybe the most promising tool is agentbased modelling, but its slow development and the lack of standardized methodologies have disappointed the public and researchers in the last decade.

Because of this omnipresence of complex systems, in this chapter an effort is made to introduce the state-of-the-art of different aspects of scientific and engineering advances in related fields. Therefore this chapter cannot focus on a single topic, it has to sum up at least some of those complex systems concepts, software development methodologies and even some commercial enterprise software products to show in how many ways the challenges of complexity are met. In a way this method of showing typical examples only is true to the philosophy of this thesis, because it would be impossible to handle such a wide area by going into details at every topic.

3.1 Complex Systems − Complex Behavior

Complex systems methods use systemic inquiry to build fuzzy, multivalent, multi-level and multi-disciplinary models of reality. (This definition of the problem acknowledges − opposite to those who circumvent it − that in a real complex problem the problem boundaries are usually not known.) The way to understand these models and reality through them is to look for patterns that seem to have some meaning.

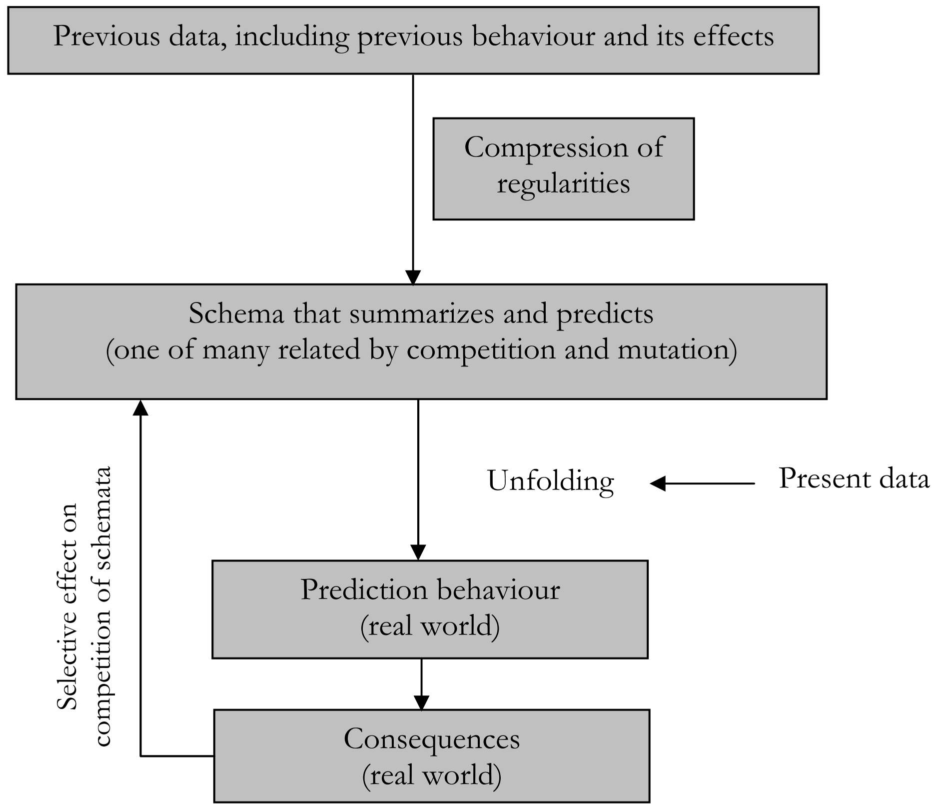

Figure 3-1 The operation of a complex adaptive system (figure from Gell-Mann [16])

The reason why it is customary to start with extremely simple simulation models of complexity is that at that simple level it should be possible to understand how the computational model is working and why. In case the mechanism of the model is understood one step can be made further and so on, until a model is available with known mechanisms, i.e. something science aims at, working models with predictive power which are also understandable. Following this method the results may very well be user dependent and might only provide complimentary insight. Complex systems are elusive, because they transition between different equilibria, self-organize and "control and order is emergent rather than predetermined" [13].

In Fig. 3.1 taken from the book of Murray Gell−Mann an intuitive representation of a complex adaptive systems shows how these systems work. Basically this is an information processing framework that is processing schemata, compressed knowledge, that can have all kinds of meaning in different environment.

3.1.1 A Computational Complexity Paradigm − Classifier Systems

Classifier systems are the most characteristic and abstract, computational models of CAS. They are well researched and lots of software is available free on the web for experimentations. They represent the same idea as the CAS of the previous figure, but are less abstract, giving a specific representation and schemata processing mechanism. In *Fig. 3.2* and *3.3* we can see two simple classifier systems, and find how they process schemata in order to adapt to the environment.

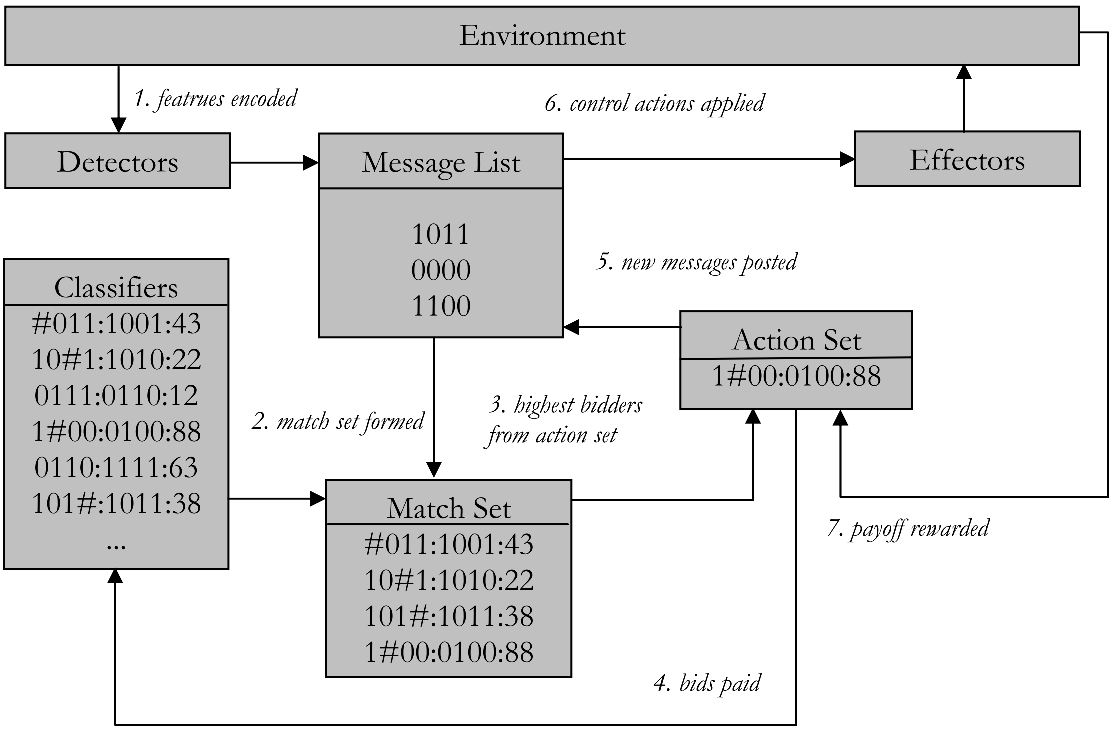

Figure 3-2 Classifier System (figure from Flake [13])

The metaphor of this schemata processing is used in evolutionary biology, sociology and in complexity science as well. However, a classifier system is only a low level representation of a complex system, i.e. it is at the computational, implementation level that doesn't provide any help in how to map a system on it.

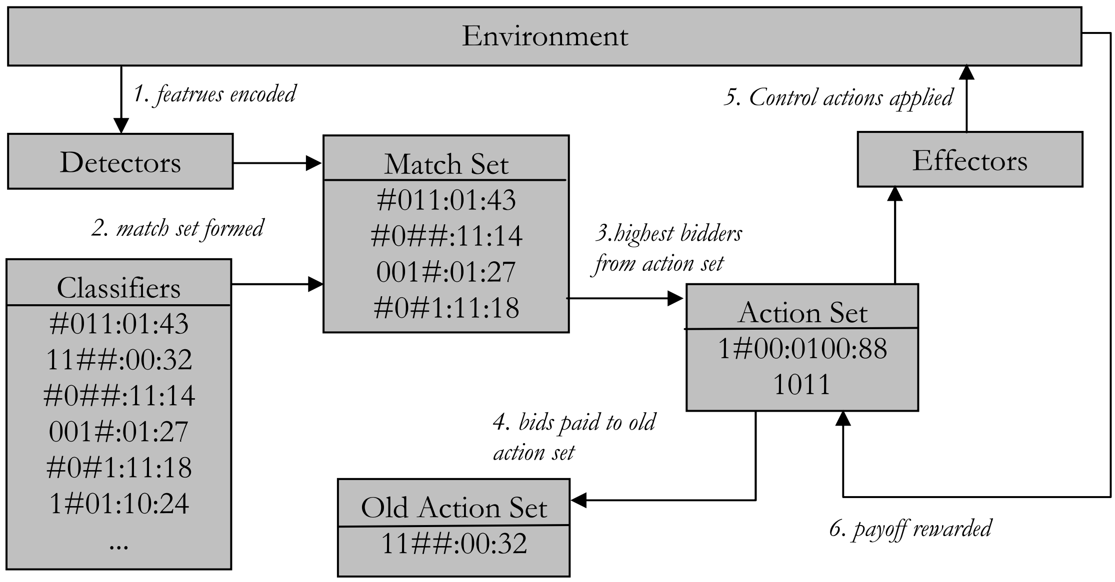

Figure 3-3 Wilson's zeroth level classifier system (figure from Flake [13])

3.1.2 "Harnessing Complexity" – a Framework

In the recent book from Prof. Axelrod et al. *"Harnessing Complexity – Organizational Implications of a Scientific Frontier"* [3] a rhetoric framework is developed from the experience the authors had with agentbased modelling and simulations. The framework is organized around three main actions with further recommendations:

1. Variation
    a. Arrange organizational routines to generate a good balance between exploration and exploitation.
    b. Link processes that generate extreme variation to processes that select with few mistakes in the attribution of credit.[6]
2. Interaction
    a. Build networks of reciprocal interaction that foster trust and cooperation.

---

[6] This means – in the terminology of the referenced author – that processes with high variation (i.e. good at the exploration of possibilities, e.g. high mutation rate in a genetic algorithm) should be connected to effective selection processes (e.g. the objective function of an algorithm) that exploit this variety. This rule is related to the exploration vs. exploitation tradeoff.

    b. Assess strategies in light of how their consequences can spread.
    c. Promote effective neighborhoods.
    d. Do not sow large failures when reaping small efficiencies.
3. Selection
    a. Use social activity to support the growth and spread of valued criteria.
    b. Look for shorter-term, finer-grained measures of success that can usefully stand in for longer-run, broader goals.

The importance of Axelrod's work is at least twofold, not only his conclusions are important, but also how he arrived at them. At the beginning of the book he makes it clear what the title of the book suggests: it is time to give up the illusion that we are in control, complex systems cannot be controlled, that is why they are called *complex*. Classical control problems and even the problems solved by Cybernetics and General Systems Theory consider only negative feedback control, no positive [24], so in case we would like to use the term "control" in complex systems we have to redefine it. Complexity, however, can be harnessed − if not controlled by classical methods − if the basic guiding principles are known it is possible to adjust to them. This only means a higher probability of success, but still this is the best we can do. It is important from the point of view of this thesis, that the conclusions of Axelrod's framework were developed by observing complex systems and building numerous agentbased simulation models. (These results can be found in [4]).

3.2 Management Cybernetics

While in the previous sections we found CAS and CS representations of complex systems and in Axelrod's work a positive example of how agentbased modelling of complex systems can lead to strategic/rhetoric knowledge, we still don't know anything about how the systems are mapped on computational models.

Complex systems research is typically associated with the "complexity hype" of the 1990's. Interestingly, the books and publications of this hype rarely make any notice that a similar one had taken place in the 1950's under the name of Cybernetics and General Systems Theory [52][9]. In those days digital computers were already available and the development of technology and mathematics inspired systemic thinking and the recognition of new problems. Many of these theories were too much ahead of their times, e.g. John von Neumann talking about the necessity of individual-based modelling [35].

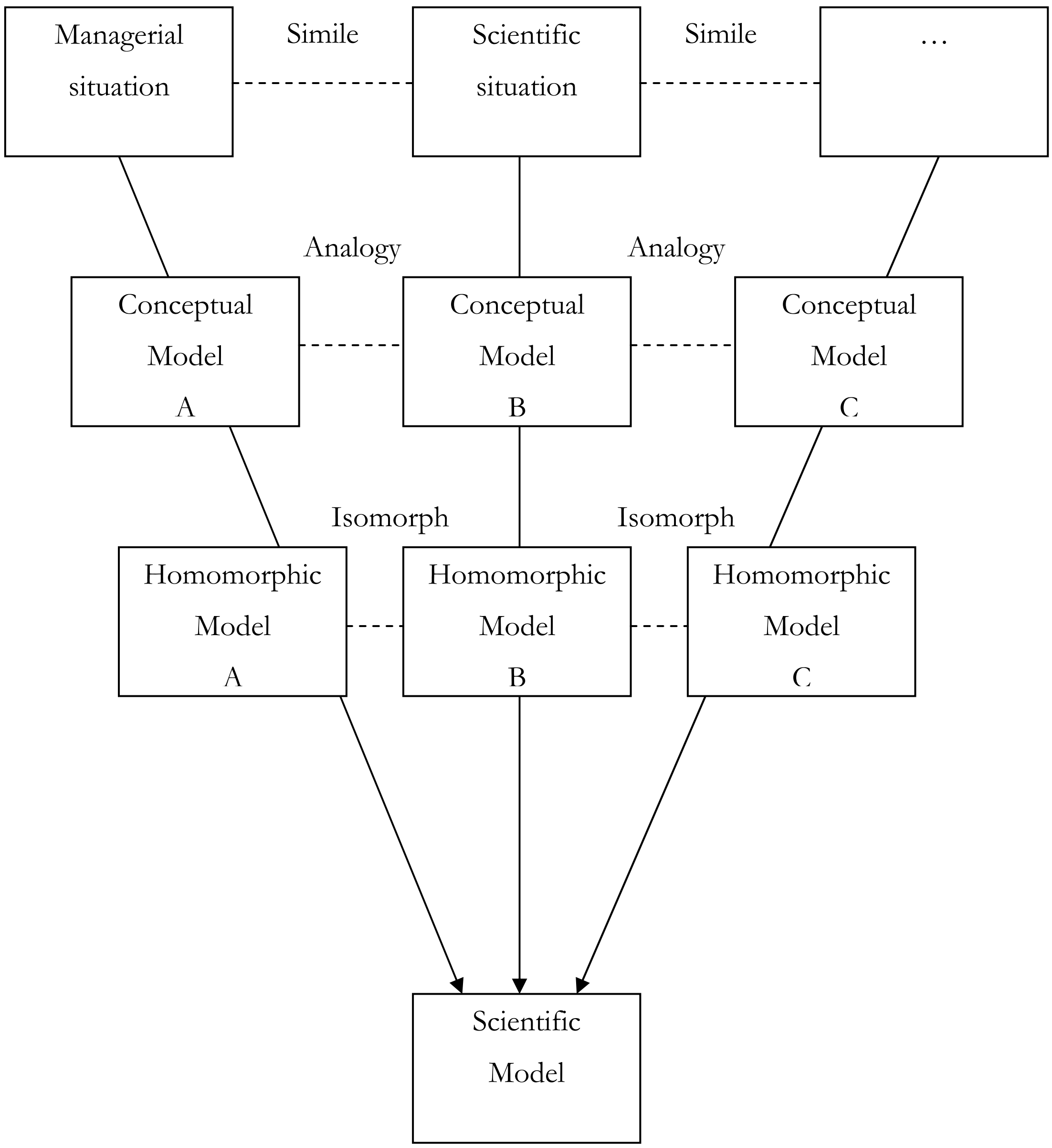

Figure 3-4 Application of the theory of models (figure from Beer: Decision and Control [6])

One of the most worked-out applications of these theories is the work of Stafford Beer, whose explanation and ideas about using metaphors, analogies, similarity in modelling are very relevant to this thesis, too. In Fig. 3.4 similarity is shown at different levels of abstraction as simile, analogy and isomorphism. His claim in [6] *"the model of any one system stands in some sort of correspondence with the model of any system: the question is whether the* *correspondence is great or small – and therefore more useful or less useful"* is very important, too, because on the one hand it encourages us to draw analogies between seemingly distant phenomena, but on the other hand warns that "some sort of correspondence" doesn't necessarily mean anything, it has to be tested for usefulness. These ideas will be applied in the developed methodology of chapter 6.

3.3 Software and Methodologies

Software engineering translates models into code. In case of complex and component-based problems and models (e.g. agentbased models) the software engineering methodology has crucial importance.

Object Oriented Design (OOP) became an everyday tool for engineers and it is considered a proven technology. However, in case we are aiming at building agents of all kinds the paradigm has to be extended, because a higher level of independence for agents is necessary than in case of objects. Therefore Agent Oriented Design was developed to translate agent models to software. Agent Oriented Design and Programming is still using OOP for implementation and it is a very suitable one, but the implemented agents reflect a special programming philosophy, as agents can have their own goals, communicative actions and protocols that enable them "to live their own life" on the basis of preprogrammed goals and actions.

In the following sections some tools, development– and modelling methodologies are introduced, all related to software.

3.3.1 Agentbased Modelling

Agentbased modelling (also called Individual Based Modelling, IBM) has a central importance in modelling and representing complex systems. It shouldn't be confused with agent-oriented software engineering, because that is a software development method, while ABM is a modelling method. However, they are not far from each other, since agentbased models are implemented as software. The following list includes the main properties of agentbased modelling (ABM):

- it's a bottom-up modelling approach

- agents are the basic entities of the simulation, they have behaviors and actions
- agents are completely independent, i.e. they have their own intelligence, make their own decisions
- the modeled system is typically a large set of agents
- the behavior of the system of agents "emerges" from the basic properties of the agents, and this emergence is the subject of study
- agentbased systems are very close to natural systems, and consequently they are potentially:
  - flexible
  - robust
  - often simpler than manmade systems
  - distributed in space and control
  - evolving
  - learning/adaptive
- it is very closely related to complexity science

ABM therefore seems – and is proven – to be a viable alternative of exact mathematical methods for handling high complexity. Various software packages are offered for ABM, e.g. Ascape, AgentSheets, Repast and the earliest one called Swarm.

3.3.2 Agentbased Simulation Systems

Agentbased simulation software are the most important for our investigations since they are capable of building bottom-up/synthetic models and incorporating many of the requirements we impose. In the next section the Swarm simulation system is summarized – as we use it in the ModNet toolbox – but there are many similar ones available. Most of them are free to use, but this only shows that they are not ready for commercialization yet, so it's not such a good news. To mention some of the other known agentbased simulation software:

- Ascape
- AgentSheets
- Jade

- JAS
- NetLogo
- RePast (this would probably be the next best candidate for our toolbox implementation, because it has a structure and philosophy similar to Swarm and is a simple to use Java component)
- Zeus

3.3.3 The Swarm Simulation System

Swarm is a free agentbased simulation package – www.swarm.org – licensed under the GNU General Public License (GPL).

The characteristics and advantages of this package over others are:

1. very heavily tested by numerous applications
2. free of charge
3. its *Objective-C* source code is available
4. it can be programmed in any programming language that can speak to the Java Virtual Machine (most importantly the Java language)
5. its binaries are compiled for all major platforms (Win32, Unix/Linux, Solaris, MacOS/X)
6. since it is a developers kit, it can be integrated with any software
7. it has toolboxes developed by its user community, e.g. statistics, genetic algorithms, evolution, fuzzy, GIS
8. it is freely and reliably supported by the Swarm Development Group (Santa Fe, NM, USA)
9. it has a user community of several thousand from all over the world (Swarm User Community
10. it is used for the research of complex adaptive systems and agentbased systems in Biology, Computer Science, Engineering, Economics, Ecology, Culture/Anthropology, Political Science, Geography and Defense

3.3.4 The Java Enterprise Simulator – *jES*

The jES built on top of the Swarm simulation system is very similar to the ModNet simulator[7] with two differences: it is much more advanced and it is concerned much more with questions related to economics, not only structural and production system level issues. jES uses abstract representations and enables the user to discover also imaginary systems and investigate model problems [48].

jES basically consists of units, orders and recipes that build up the simulations. The recipes are equivalent to a process plan. Three formalisms of the enterprise production are abbreviated as WD, DW, WDW – What to Do, which Doing What and When Doing What, i.e. orders, process plans and schedules. The package models fixed and variable costs, including production unit setup costs, and news propagation/information sharing.

The most recent development of the package (jESevol) includes tools for the investigation of evolutionary steps in enterprise development. This package is under development, but some preliminary results[8] suggest that this will be a heavily used and widely explored tool in the near future, especially because it is the only one providing all the tools for the exploratory modelling of virtual enterprises.

3.3.5 Gaia Methodology

The Gaia methodology is one of the rare methodologies, which are concerned about agent-oriented analysis and design. It is a general methodology, which supports both micro-level (agent structure) and macro-level (agent society and organization structure) aspects of the system. Existing methodologies cannot represent the autonomous nature and problem-solving ability of agents and also fail to model agent's ways of performing interactions and creating organizations. Using this methodology the software designer can develop an implementation-ready design based on system requirements.

---

[7] Developed at the Production System Design Laboratory, Institute of Production Engineering, Tampere University of Technology, Finland, work supported by the Academy of Finland, TUKEVA programme, 2000-2003.

[8] See http://web.econ.unito.it/~terna/jes for details.

The methodology is used for both analysis and design. Analysis and design can be thought of as a process of developing increasingly detailed models of the system to be constructed.

By breaking down the problem and modelling its different aspects we have a methodology that has the potential of producing high quality agent-oriented software model [53].

3.4 Commercial Business Software

In *Table 4.1* we can see a list of commercial software systems that obviously concentrate on those problems that can be formulated as optimization or reasoning problems. One of their practical advantages is that they can all be integrated with other software, creating a fully integrated software environment. It is encouraging to see that NuTech Solutions makes good use of agentbased modelling and turned the just recently maturing technology into a commercial product.

3.5 Economic and Organizational Issues

As the problem formulation in the second chapter has emphasized the goal of the development of a methodology in this thesis is the modelling of complex systems in general but production networks are pointed out as an example that is discussed in more detail. Because of this focus it is important to have a concentrated look on the non-orthodox literature of organizational theory and point out some interesting observations very relevant to this work.

Until the beginning of the 1980's the firm was usually considered as a black box problem and it was handled in that manner [18]. The opening up of the black box and the accumulation of slow development shows interesting parallels with scientific and technological advances. Organizational and economic theory faced those challenges that were mentioned in chapter 2 in a little different context, i.e. the problem's boundaries are blurred (what to model?) and analytic tools may not be suitable to handle the problem.
The availability of cheap computers and simulation software, and the idea of agent/individual-based modelling along with developments in complexity science has had a significant effect on organizational theory as it introduced a new way of inquiry.

As Gibbons claims in [18], the self-interested agents are the key to explain why so many policies crucial for the performance of the firm are not implemented. On the one hand this brings the focus to the individuals making up the firm and on the other hand encourages the use of such alternative modelling methods as agentbased modelling. In a similar way to the "harnessing complexity" ideas of Axelrod introduced in section 3.1.2, Axtell inquires agentbased simulation models to draw conclusions about the evolutionary mechanisms of firms [5]. Such work involves the definition of detailed metrics, goals, individual behaviors comprising a modelling network − and is certainly sensitive to the methodology applied during the process. Axtell's work is true to the complexity of the task and complex systems research, because it demonstrates how growth, stability and cooperation emerge from individual behavior of the actors in the firm.

The use of metaphors and analogies is beautifully demonstrated in the work of Padgett et al. [38], who − at the Santa Fe Institute − have developed an economic model of production networks analogue to chemical systems. They also demonstrate their "complex systems thinking" by considering product flow as a coevolutionary process: products change the enterprise, just as the enterprise changes the products. They point out that the learning process related to changing contracts/partners and therefore products is often ignored and is worth modelling.

All these ideas from the literature illustrate the trend of systemic thinking, agentbased modelling and simulation, the use of metaphors, analogies and rhetoric, and the importance of conducting research in the area of complex systems. Also, these examples shed some light on how new methodologies emerge and how complex systems can be investigated.

Table 4.1 Software systems tackling the complexity of enterprises

Company	Software	Purpose
SAP	MySAP SCM	Supply chain management, share information, knowledge, resources
SAS	Supply chain analysis	Analytics, process intelligence modelling, intelligence architecture
NuTech Solutions (Biosgroup)	Supply network optimization	Decrease inventory, reduce out-of-stocks, adaptive agents
Oracle	SCM	Real-time collaboration, low inventory, logistics, supply chain planning
Prowess Software	MarketProwess	Trade-off analysis
I2	Logistic Optimization	Design logistic network, bidding and rate negotiations, planning
IBM	Supply Chain Simulator (SCS)	▪ model and analyze supply chain issues ▪ site location ▪ replenishment policies ▪ manufacturing policies ▪ transportation policies ▪ stocking levels ▪ lead times ▪ customer service

Chapter 4 Scientific Establishments

"I have never doubted the truth of signs, Adso; they are the only things man has with which to orient himself in the world. What I did not understand was the relation among signs. I arrived at Jorge [i.e. the murderer] through an apocalyptic pattern that seemed to underlie all the crimes, and yet it was accidental. I arrived at Jorge seeking one criminal for all the crimes, and we discovered that each crime was committed by a different person, or by no one. I arrived at Jorge pursuing the plan of a perverse and rational mind, and there was no plan, or, rather, Jorge himself was overcome by his own initial design and there began a sequence of causes, and concauses, and of causes contradicting one another, which proceeded on their own, creating relations that did not stem from any plan. Where is all my wisdom, then? I behaved stubbornly, pursuing a semblance of order, when I should have known well, that there is no order in the universe.-But in imagining an erroneous order you still found something… The order that our mind imagines is like a net, or like a ladder, built to attain something. But afterward you must throw the ladder away, because you discover that, even if it was useful, it was meaningless… The only truths that are useful are instruments to be thrown away."

Umberto Eco[9]

"A theory is more impressive the greater the simplicity of its premises is, the more different kinds of things it relates, and the more extended its area of applicability. Therefore the deep impression which classical thermodynamics made upon me. It is the only physical theory of universal content concerning which I am convinced that, within the framework of the applicability of its basic concepts, it will never be overthrown."

Albert Einstein[10]

It is often emphasized by scientists that "there is only one physical reality, but it has many facets". In the caption of this part the semiotician Umberto Eco puts the purpose of scientific inquiry into the form of epic. The patterns we are looking for – to possess more control – do not represent and aim at providing "absolute truths" in the philosophical sense, but "ladders", tools that are built to "attain something" and to be thrown away if we find another tool or discover another aspect of reality. Being and working in the knowledge of this idea is an important influential factor for researchers. In case of complex adaptive

[9]Umberto Eco, writer, Nobel laurate, also an expert in semiotics, quote from *The Name of the Rose,* p.492. 1980.
[10] Albert Einstein (1879-1955)

systems it is especially important to realize this, since we can't even formulate the problem easily (as it may be a moving target). Our goal is to find a flexible, multifunctional "ladder".

Gaining knowledge includes scientific and metaphorical processes. In this chapter we find a collection of important scientific achievements and discover also the limitations they impose. Also, these scientific paradigms can serve as excellent metaphors, since the knowledge they "compress" is significant and has been used in many different contexts. So, in the next sections we first see a summary about scientific discovery, and a collection of scientific tools and candidate metaphors.

4.1 The Nature of Scientific Discovery

The subject and scope of this work requires a more detailed exploration of its limitations and philosophical background than it is usual in engineering theses. As it is apparent in this chapter, the new paradigms of computational undecidability, chaos and unpredictability imply us to seriously consider the means and limitations of our scientific understanding and explore the possibilities for the advancement of our field. The basic questions regarding the background of this thesis are concerned about the following issues:

- *The relation of scientific and rhetorical knowledge*
- *Objectivity and subjectivity of human knowledge*
- *Knowledge discovery through computational modelling*
- *Practical consequences of the above*

The treatise of these important questions were significantly influenced by the work of Aristotle, Karl R. Popper, Henry Poincaré, Michael Polányi, and recently summarized in the context of complexity science by Fortunato T. Arecchi [1], whose conclusions provide a strong influence on the argumentation in the next sections.

As early as in Aristotel's work it was acknowledged that to arrive at conclusions and gather knowledge about the world we have at least two basic methods: the scientific and the rhetorical one, as it is depicted in *Fig.4.1.* The scientific method has been clearly dominant in the European culture, and the rhetorical one has been typically neglected.

As Encyclopedia Britannica puts it, *science is any system of knowledge that is concerned with the physical world and its phenomena and that entails unbiased observations and systematic experimentation. In general, a science involves a pursuit of knowledge covering general truths or the operations of fundamental laws.* However, this pursuit can take many paths and use different languages.

The first attempt at creating a scientific language was introduced by Galileo, who tried to overcome the limitations of everyday languages, where one linguistic symbol may describe several different objects or events, and one object might have several names – a potential confusion. Instead of a one-to-one association of lingual symbols and objects/events he shifted the emphasis to their measured quantities (and avoided the vague battle to grasp the essence of reality by words). In his scheme of the language of physics, reality is described by physically measurable quantities.

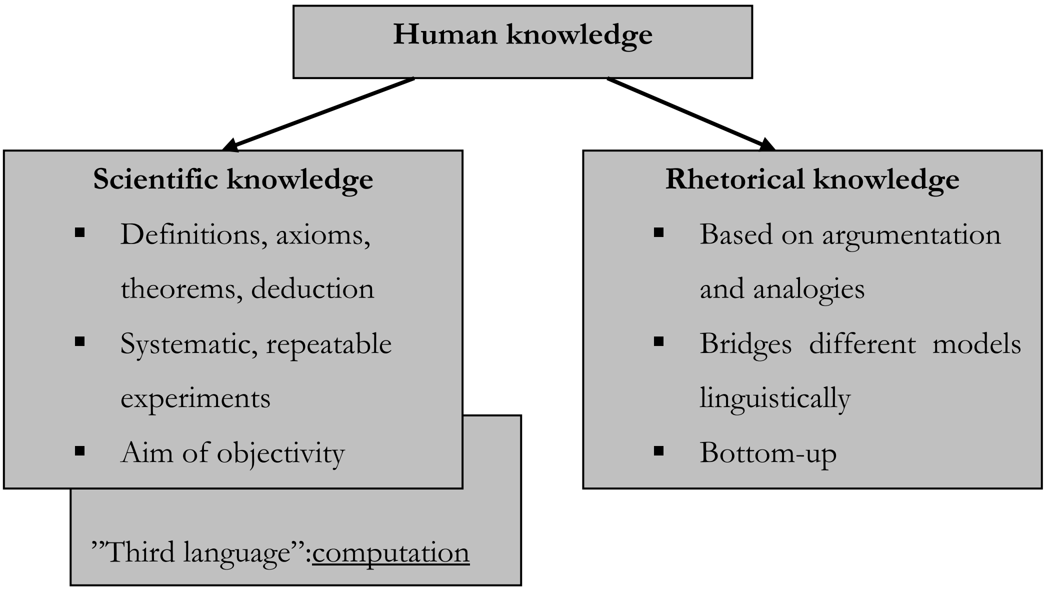

Figure 4-1 A classification of human knowledge

In any scientific endeavor definitions, axioms and theorems have a central role. It is often disputed how we arrive at these mathematical objects. The so-called "transcendental" explanation says that they are "given", and after deducing theorems from the definitions and axioms we may find some natural phenomena that "match" the mathematical structure[11]. It is easy to see why this is called "transcendental": *the first is the idea and reality is*

[11] This way of thinking is associated with german philosopher Immanuel Kant (1724-1804).

derived from it. However, it is common experience that this is not how things happen. Usually the observations and experience of the scientist motivate the measurements and the axioms, definitions are also motivated by them. After a set of axioms, definitions and theorems is set up, this system is used for prediction – the ultimate goal of the whole scientific procedure – and its result can be compared to observations – i.e. reality – and the whole system is due to skeptical criticism and revision. Arecchi calls this procedure as "critical realism"[1].

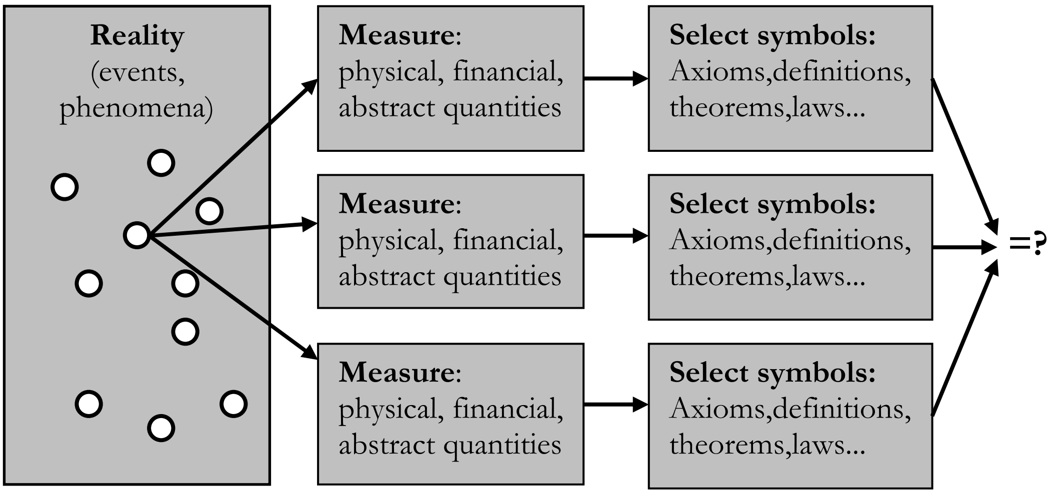

Figure 4-2 Development of different sciences of the same phenomenon by quantitative measurement and mathematization

An important step in the scientific method is the choice of axioms and definitions, the starting elements of the mathematical language (and its coevolution with observed reality), because this choice is subjective and depends largely on the decision maker's background and taste. This choice is also referred to as "common sense", warning us that even the most rigorous scientist is prone to make arbitrary decisions during the modelling process.

When looking at a problem (i.e. part of physical reality) we have some details and a global picture at hand. In the knowledge gathering process every involved individual has to find an optimum between the extremes of specificity and generality, and form their "own picture" of the problem (meaning that everybody will have a different optimum depending on their endogenous characteristics).

It is a generally accepted conclusion that follows from these shortcomings or limitations of the scientific process that there is no unique science, only different aspects of reality (*Fig. 4.2* and [12]).

The introduction of metaphors and analogies into the scientific language [1] aims at providing bridges between the different aspects of reality (the products on the output of *Fig.4.2*) grasped by different scientific tools. Such a bridge can only be heuristic, making use of its previous compression of knowledge[12] in another environment. In the same time a metaphor – due to its external origin – has the potential of enriching the method with a view of reality not yet captured. The use of metaphors/analogies is going to be explored further and applied in the methodology development.

4.2 The Metaphor of Evolution and its Application

The concepts of biology and evolution are often closely related to computation. Since optimization is inherent in many natural paradigms, their computational counterparts can often be applied in optimization problems, the most typical of which are evolutionary computation methods. However, it is not proven – and is not assumed by the evolutionary computation community – that simulated evolution applied to optimization problems works because it is essentially the same as biological evolution first observed by Darwin.

4.2.1 Neo-Darwinian Evolution

The paradigm of evolution that is the basis for some algorithmic optimization techniques is well described by D. Futuyuma:

"In the broadest sense, evolution is merely change, and so is all-pervasive; galaxies, languages, and political systems all evolve. Biological evolution ... is change in the properties of populations of organisms that transcend the lifetime of a single individual. The ontogeny of an individual is not considered evolution; individual organisms do not evolve. The changes in populations that are considered evolutionary are those that are inheritable via the genetic material from one generation to the next. Biological evolution may be slight or substantial; it embraces everything from slight changes in the proportion of different alleles within a

[12] These are the schemata of science (i.e. a complex adaptive system) according to Gell-Mann [16].

population (such as those determining blood types) to the successive alterations that led from the earliest protoorganism to snails, bees, giraffes, and dandelions." [14]

4.2.2 Evolutionary Computation

Evolutionary Computation (EC) or Evolutionary Algorithms (EA) is a general term for a number of simulated evolution strategies. Their general structure is represented as a pseudo–code below.

```
Start EA
          // start with an initial time
          t := 0;
          // initialize a usually random population of individuals
          initpopulation P (t);
          // evaluate fitness of all initial individuals in population
          evaluate P (t);
          // test for termination criterion (time, fitness, etc.)
          while not done do
               // increase the time counter
               t := t + 1;
               // select sub-population for offspring production
               P' := selectparents P (t);
               // recombine the "genes" of selected parents
               recombine P' (t);
               // perturb the mated population stochastically
               mutate P' (t);
               // evaluate its new fitness
               evaluate P' (t);
                 // select the survivors from actual fitness
               P := survive P,P' (t);
          loop
End EA
```

The most important variations of EAs are:

- Genetic algorithms (GA)
- Evolutionary programming (AP)
- Evolution strategies (ES)
- Classifier systems (CFS)
- Genetic programming (GP)

All these are very similar in the basic ideas, but their versatility indicates that in their design the engineer has a great variety of choices, and the success of the chosen and tested solution depends mainly on his intuition. First we will provide a detailed description of the most frequently applied GAs and after the last four variations of evolutionary algorithms will be discussed.

4.2.3 Genetic Algorithms (GA)

GAs have many variations, but all of them follow the sequence of activities represented by the following pseudo–code.

```
Start GA
          // start with an initial time
          t := 0;
          // initialise a usually random population of individuals
          initpopulation P (t);
          // evaluate fitness of all initial individuals of population
          evaluate P (t);
          // test for termination criterion (time, fitness, etc.)
          while not done do
               // increase the time counter
               t := t + 1;
               // select a sub-population for offspring production
               P' := selectparents P (t);
                 // recombine the "genes" of selected parents
               recombine P' (t);
               // perturb the mated population stochastically
               mutate P' (t);
               // evaluate its new fitness
               evaluate P' (t);
               // select the survivors from actual fitness
               P := survive P,P' (t);
          loop
End GA
```

It is the attribute of all evolutionary systems but especially the most widespread GAs that they can be – and have to be – "tailored" to the problem at hand. This is an advantage and a disadvantage in the same time. The termination of the algorithm is not clearly defined, most of the time it depends on the number of generations or the quality of the solution.

A simple and understandable definition is given in [49]: *"Genetic algorithms are a stochastic search method whose mechanisms are based upon simplifications of evolutionary processes observed in Nature".*

Genetic algorithms operate independent of the problem to which they are applied. The genetic operators are heuristics, but they are not working in the *solution–space* (that is called *phenotype–space*), but in the *representation–space* (this is called *genotype–space*).

The parameters to be optimized form the *phenotype space*. The genetic operators, however, work on abstract mathematical objects like binary strings, or any other representation, i.e. on the *genotype space*. The mapping between the two spaces is done by the h' decoding function as shown on *Fig. 4.3*. There are two main different approaches related to decoding. The first one is to design the representation as close as possible to the phenotype space to make decoding easier. The second one is to choose a standard algorithm that is already available and then design the decoding function. Each has advantages and disadvantages: the first one requires a very specific solution and the second one struggles with complex decoding functions, which can introduce additional mathematical complexity, nonlinearity to the algorithm.

However, if we want to produce a well performing algorithm then most probably the only way of building it is to make it very specific, significantly different from standard algorithms. A great advantage of using evolutionary heuristics is that the requirements for their application are very modest in comparison to most other search techniques[8]. In addition, genetic algorithms use other heuristics for determining which individuals will mate (selection), which will survive to the next generation (replacement), and how the evolution should progress.

Some genetic algorithms measure the similarity between solutions, because their common problem is premature convergence, when the solution space is not well explored, the genetic composition is the same for all.

Usually the best individual is carried to the new generation, that makes the algorithm to converge quicker. Since the entire population is replaced, everything depends on the crossover operator's performance. This means that if that operator has high performance, then the population will improve, but anyway the algorithm will be just like a random search. However, this technique is usually not a reasonable one.

The second variation of genetic algorithms is the steady-state algorithm with overlapping populations. A portion of the population is replaced by newly generated individuals. If only a few individuals are replaced the overlapping is close to 100%. However in the other extreme, if all of the individuals are replaced then we get the previously described simple genetic algorithm. This version has a quick convergence, because only the best individuals are selected to survive, and they are then replacing those that are not so good.

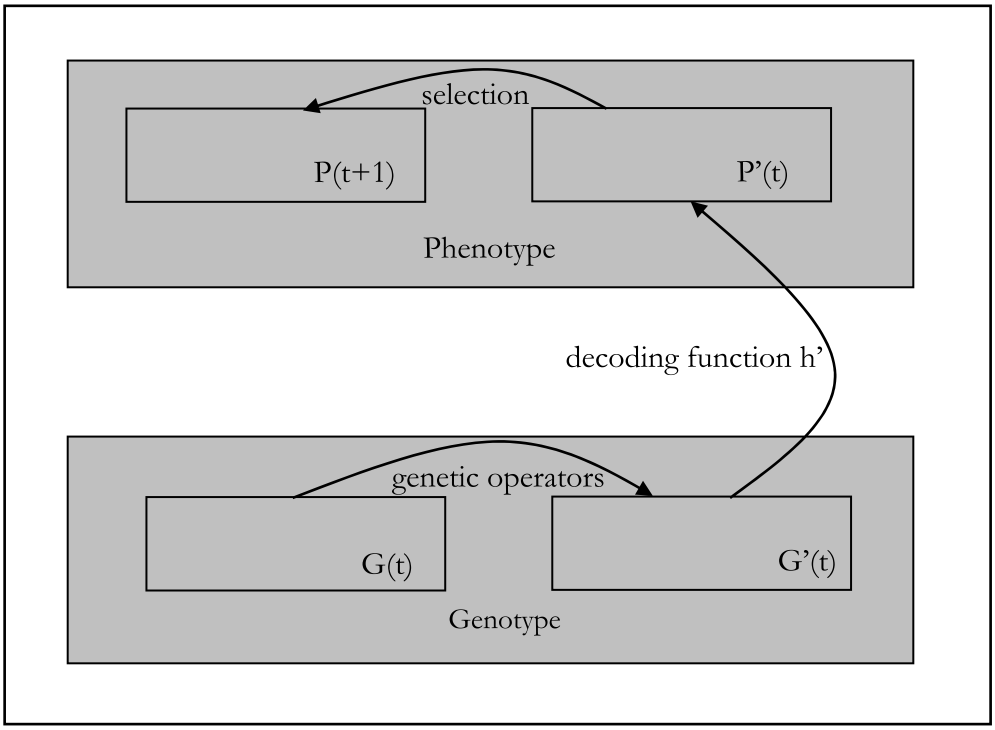

Figure 4-3 Abstraction between the geno- and phenotypes of a genetic algorithm

The simple and the steady-state algorithms will result in many copies of the same individual from the population. Once it happens, the only source of change is the mutation operator. The struggle algorithm maintains the problem of diversity the best, but as in every other case, the performance depends on the operators. If the mutation rate is too high, then the algorithm will perform just as a random search.

4.2.4 Evolutionary Programming (EP)

EP was originally invented by Lawrence J. Fogel in 1960, and the method is similar to that of genetic algorithms. It is more concerned about the behavioral link between the *parents* and the *offsprings,* and not the genetic operators. The process is as follows:

1. Generate an initial population of solutions. The size of this solution determines the quality and the speed of the convergence.
2. Execute *mutation* on all the members of the population. This mutation is different for all members, ranging from small to large mutating effects.
3. Determine the fitness of the members, and choose some of them to survive either by selecting them deterministically, or by stochastically.

It is important to note that there is no *crossover* as a genetic operator in this case.

Another important issue is that the representation is not constrained at all, there is no need for encoding, the representation follows from the problem. Also, the mutation operator is making minor changes only, and as the global optimum is likely to be approached it slows down.

The EP algorithm is represented in pseudo–code in [21] as follows:

```
Start EP
       // start with an initial time
       t := 0;
       // initialise a usually random population of individuals
       initpopulation P (t);
       // evaluate fitness of all initial individuals of population
       evaluate P (t);
       // test for termination criterion (time, fitness, etc.)
       while not done do
              // perturb the whole population stochastically
              P'(t) := mutate P (t);
              // evaluate its new fitness
              evaluate P' (t);
              // stochastically select the survivors from actual fitness
              P(t+1) := survive P(t),P'(t);
              // increase the time counter
              t := t + 1;
       loop
End EP
```

4.2.5 Evolution Strategy (ES)

ES is a concept born in 1963 at the Technical University of Berlin by *Rechenberg*, *Bienert*, and *Schwefel*. The main idea is usually applied for technical optimization problems, e.g. designing optimal shapes of airplane wings, turbine blades, etc. The mechanism of an ES is very similar to an EP, and despite of their long isolation – the first moves of the research groups were made in 1992 – they share many of the theoretical basics [33].

The steps of a simple – two membered – ES start with the generation of an initial population, then follows with slow mutation of the subsequent generations, until the child performs better than its ancestor, and is replacing it. This is such a simple structure, that even a few theoretical results were derived. Applying the more complex algorithm of *Schwefel* is multimembered, and is employing recombination with random mating, and mutation and selection.

However, the selection in case of an ES is rather deterministic, the worst individuals in the population are eventually removed, while in case of an EP the selection is more stochastic, and makes more use of probability theory and statistics. Another difference is that EP is an abstraction at the level of populations, and ES is at the level of the individuals. This means that recombination does not occur between different populations of an EP, but an ES may incorporate the self–adaptive information at the genetic level.

4.2.6 Classifier System (CFS)

Classifier Systems – also called *Evolutionary Reinforcement Learning* – employs similar concepts to other EC methods, especially GAs. The idea envisioned by Holland is to create a system that can sense its environment, and is able to react appropriately. The following elements build up the model [13]:

1. Environment
2. Receptors
3. Effectors

4. CFS (the system itself, a black box)

The simplest black box (let us say this is the CFS) is a computer. The computer (4) is able to transform the environment's (1) interactions with its input (2) and take the actions – according to the rules stored in it, e.g. IF-THEN rules – through its output (3). If we call the rules a classifier, and encode it – e.g. in the classical evolutionary way – as a binary string, then the set of rules becomes a *population*. Since we already know the basics of EC the rest is easy, zeros and ones are floating through the black box, this is called the *message list*. During the operation the inputs send a message list – the initial population – and in the black box it is decided which of the IF-THEN rules are activated to produce an output. The message list is emptied and the cycle is repeated.

This cycle is represented below in pseudo code:

```
Start CFS
        // start with an initial time
        t := 0;
        // an initially empty message list
        initMessageList ML (t);
        // and a randomly generated population of classifiers
        initClassifierPopulation P (t);
        // test for cycle termination criterion (time, fitness, etc.)
        while not done do
                // increase the time counter
                t := t + 1;
                // 1. detectors check whether input messages are present
                ML := readDetectors (t);
                // 2. compare ML to the classifiers and save matches
                ML' := matchClassifiers ML,P (t);
                // 3. process new messages through output interface
                ML := sendEffectors ML' (t);
        loop
End CFS
```

The general idea is to start with a randomly generated list, and then let the system learn the proper program itself by induction. However, CFSs is the area where there are the most problems coming up, and therefore its practical applications are not as widespread as most of the other EC methods, they are limited to research at the moment, but its unexplored nature makes it intriguing for experimentations.

4.2.7 Genetic Programming (GP)

GP is the extension of the genetic model of learning into the space of programs. The representation is a tree of solutions, and is very special, and the population is not a set of fixed–length characters, but is variable, it is a program that represents a candidate solution. In GP the *crossover* is implemented by selecting subtrees randomly and exchanging them. In the classical GP there is usually no *mutation*. Also, GP is an offspring of GA. [28]

This simple introduction to genetic algorithms described the main idea of genetic evolution, but the simple concepts do not mean that their implementation is simple, too. Publications about the topic emphasize that designing genetic algorithms is still an art, and no standard methods exist. That is why their application is not as widely accepted as that of other heuristics, novice engineers easily build algorithms which perform only as well as a random search.

4.2.8 Conclusions on Evolution

This chapter and the introduced evolutionary algorithms provided good examples for complex adaptive systems. At the same time they represent a metaphor-based knowledge transfer from biology to computing. It has not been proven that simulated evolution is the same as the biological one. However, simulated evolution has been very successful to solve real world problems.

4.3 Artificial Immune Systems

In the last decade there has been a growing interest in the use of the biological immune system as a source of inspiration to the development of computational systems. The immune system contains many useful information-processing abilities, including pattern recognition, learning, memory and inherent distributed parallel processing. For these and other reasons, the immune system has received a significant amount of interest to use as a metaphor within computing. This emerging field of research is known as Artificial Immune Systems (AIS) [11].

Essentially, AIS are the use of immune system components and processes as inspiration to construct computational systems. AIS is very much an emerging area of biologically inspired computation and has received a significant amount of interest from researchers and industrial sponsors in recent years. Applications of AIS include such areas as machine learning, fault diagnosis, computer security, scheduling, virus detection, and optimization. The field of AIS is showing great promise of being a powerful computing paradigm.

4.4 The Metaphor of Bacterial Chemotaxis

The way bacteria react to chemo attractants in concentration gradients provides another applicable paradigm for algorithmic optimization. In case of airfoil optimization it performs about as well as an evolutionary strategy [34].

4.5 Some Basic Ideas from Physics

The *Aristotelian*[13] meaning of Physics is *natural philosophy, a system of natural science* in the wider sense, but the application of the term has continually narrowed. It originally embraced the study of the whole of – organic and inorganic – nature, but further on, in the course of the XVIIIth century, its subject became limited to inorganic nature, and finally – by excluding chemistry – it has become *the scientific discipline that deals with matter and energy, and their interaction* [22].

In the previous two centuries this narrowing down continued in the separation of physical disciplines, also driven by the extensive support of engineering research. Instead of going into the details of listing fields of research, we favor the – somewhat simplistic and noniclusive, but very powerful – classification of Ilya Prigogine and Erich Jantsch. According to their view we can distinguish between three levels of inquiry in physics [24], *Classical* or *Newtonian dynamics* (4.5.1), *equilibrium thermodynamics* (4.5.2) and *dissipative structures* (4.5.3). It is characteristic of these "levels of inquiry" that the transition from one level to the other is taking place by so-called "symmetry breaks", i.e. the removal of an important "mathematical benefit" from the system.

In the following sections we introduce the basic concepts of the aforementioned three physical views and provide a summary of their potential applications (4.5.3).

4.5.1 Classical Newtonian Mechanics

13 Aristotel (BC 384-BC322)

Classical mechanics is the science upon which our belief in a deterministic, time-reversible description of nature is based [41]. The central figure of the first, classical formulation of physics was Isaac Newton[14], therefore we often refer to classical mechanics as *Newtonian.* The breakthrough hallmarked by his name found its establishment on the geometry of Euclid[15] and the experimental investigations and generalizations of their results of Galilei[16]. Euclidean geometry is the first rigorous, mature axiomatic system of high aesthetic standards, and – in the same time – of practical applicability. Galilei was the first one who not only carried out systematic physical experimentations but formulated those in the language of mathematics. However, Newton initiated and elaborated the "marriage" of these two, the result of which is classical mechanics in its Newtonian form. The central idea of this discipline is that the "history" of the elementary events happening to the physical bodies can be described as the sum of the individual stories of certain *elementary mass points* being the abstractions obtained from the observation of very small pieces of matter. The existence of the inertial systems of reference made it possible to consider these individual stories as plays taking place in front of the "metaphysical" scenes of space and time. It is this scenery that has its own internal symmetries described by the Galilei Transformations and that of the Euclidean Geometry. Matter is bound to behave according to the rules set by this "metaphysical background".

It was Newton who strongly suggested that the aim of physics is to seek "hidden" internal symmetries in the functional relations between huge sets of experimentally obtained and mathematically formulated data. This idea helped physics to gain great momentum. It made it evident that all kinds of mathematical tools helping us in representing physical quantities may be of great practical advantage in implementing this program. This way of thinking made it plausible to incorporate various concepts in the mathematical toolbox of physicists, such as:

- algebraic means as Hamilton's quaternions
- Grassmann's vectors and tensors

[14] Sir Isaac Newton (1642-1727)

[15] Eucleides (BC325-BC265)

[16] Galileo Galilei (1564-1642)

- Lie groups
- metric spaces
- Banach spaces
- Hilbert spaces
- differentiable manifolds and their tangent space, etc.

All these tools entered into the toolbox of physics through the door opened by Newton.

Other essential improvements in physics include the introduction of extremum principles, which remained in close harmony with the symmetry principles. In classical mechanics one of the most successful extremum principles was the Hamilton Principle (also called *the principle of least action*) stating that among the elements of the set of imaginable trajectories leaving point $\mathbf{q}_0$ at time instant t_0 and arriving at point $\mathbf{q}_1$ at time t_1 nature selects the realized one for which the integral S (Eqn. 1.) – i.e. also called "the action" – has its extremum value:

(Eqn. 1.) $$S = \int_{t_0}^{t_1} L(\mathbf{q}(t), \dot{\mathbf{q}}(t))dt$$

In this equation L is the Lagrangian of the physical system. This function can be constructed from the inertia, position, and velocity data of the points of the body measured with respect to an inertial system of reference. It serves as the link bridging the phenomenological and more abstract descriptions of classical mechanics. By the use of the means of variation calculus the above extremum principle yields a set of differential equations, the so-called Euler-Lagrange equations (Eqn. 2.), which can be solved in the knowledge of full information on the initial conditions of the solution. Since the system can have "s" degrees of freedom, we need as many *Lagrange's equations* to describe the system as there are degrees of freedom.

(Eqn. 2.) $$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \qquad (i=1,2,\ldots,s).$$

The simultaneous existence of integral form of the extremum principle and the differential equation requiring initial conditions generated a philosophical debate in connection with the principle of "causality". This concept seems to be clearly manifesting itself in the formulation using differential equations: the initial state of the system with the equations of motion (i.e. with the rule or physical law) determines the whole trajectory for the future. In contrast to that – as the integral form suggests – for each time instant t lying between t_0 and t_1, the past and future values have equally important effect on $\mathbf{q}(t)$. This seems to be in apparent contradiction with the interpretation based on the concept of causality.

Within the frames of Classical Physics these false contradictions are logically resolved by a special property of the equations of motion. Due to it, in the possession of the initial conditions the motion can be predicted for the future, but its history in the past can also be traced (calculated). In this way the integral form is not in contradiction with the principle of causality. Since the future is unambiguously determined by the past, the state in $t_1>t$ can determine the state in t since from it the state in t_0 preceding t_1 can be uniquely calculated, too.

4.5.2 Equilibrium Thermodynamics

On the basis of the classical mechanical formulation it is possible to describe a very large number of particles (e.g. gases) by the Euler–Lagrange equations, or – for better handling – by Hamilton's equations of motion (that we obtain directly from the Lagrangian by the Legendre transformation [29] p. 131). These "Hamiltonians" are functions of the generalized coordinates and impulses and we obtain equations of motion twice the number of the degrees of freedom of the system (while in case of the Lagrangian we have n degrees of freedom and n equations of motion, (Eqn. 3.); q is the generalized coordinate and p is the generalized momentum, H is the Hamiltonian).

(Eqn. 3.) $$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \dot{q}_i = \frac{\partial H}{\partial p_i}$$

In spite of the seemingly complete description of the system, in case of thermodynamic systems we give up the handling of exact mechanical information and concentrate on statistical averages, because the 2nd law of thermodynamics states that even if the movement of the gas molecules and the flow of heat (that is the result of molecular collisions that transfer energy) can be described by equations of motion, on the macroscopic scale irreversibility, i.e. "the arrow of time" appears. This seemingly contradictory issue was resolved by Boltzmann[17], who suggested a statistical formulation of entropy (Eqn. 4.), where S is the entropy of the system, k_B is the Boltzmann constant (k_B=1.381*10^{-23} J*K^{-1}).

(Eqn. 4.) $S=k_B \ln W$

The W means the number of microstates corresponding to the macrostate whose entropy is S (as Max Planck suggested W is called *thermodynamic probability*).

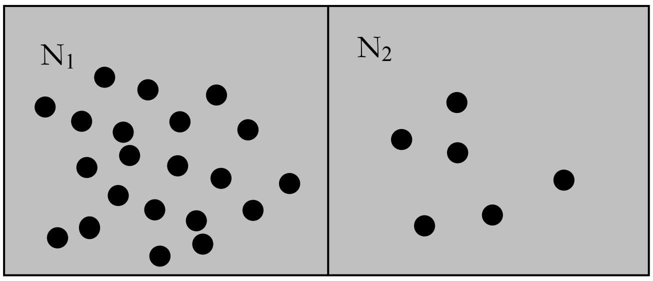

Figure 4-4 Distribution of gas molecules in a chamber

In Fig. 4.4 we can see the meaning of W. Considering the macrostate of a box containing a gas with N_1 molecules in one half of it and N_2 molecules in the other one, each molecule can be in any of the parts. The total number of ways in which the (N_1+N_2) molecules can be distributed between the two halves such that N_1 molecules are in one and N_2 are in the other is equal to W. The number of distinct "microstates" with N_1 molecules in one half and N_2 in the other is calculated in (Eqn. 5.) below.

[17] Ludwig Boltzmann, Austrian physisist (1844–1906)

$$W = \frac{(N_1 + N_2)!}{N_1! N_2!} \qquad \textit{(Eqn. 5.)}$$

According to Boltzmann, macrostates with larger W are more probable. The irreversible increase in entropy (stated by the 2nd law) then corresponds to evolution to states of higher probability. Equilibrium states are those for which W is maximum. In the above example W is maximum if $N_1=N_2$. It follows from the 2nd law is that the entropy of an isolated system reaching equilibrium also reaches its maximum. This is the *entropy maximum principle*. The 2nd law also implies that at constant entropy and volume every system evolves to a state of minimum energy ([27] p.124).

4.5.3 Nonequilibrium Thermodynamics

In classical thermodynamics stability is defined by the Gibbs–Duhem theory. However, the stability theory based on entropy production is much more general, and allows us to use it in nonequilibrium conditions, too. Hence the 2nd law we know, that the entropy change has two components, the entropy change due to irreversible processes and due to exchange of energy and matter:

$$\mathrm{d}S=\mathrm{d}_e S+\mathrm{d}_i S. \qquad \textit{(Eqn. 6.)}$$

For isolated systems:

$$\mathrm{d}_e S=0 \textit{ and } \mathrm{d}_i S \geq 0 \qquad \textit{(Eqn. 7.)}$$

For closed systems:

$$d_e S = \frac{dQ}{T} = \frac{dU + pdV}{T} \qquad \textit{and} \qquad d_i S \geq 0 \qquad \textit{(Eqn. 8.)}$$

For open systems that exchange both energy and matter (where $dU+pdV \neq dQ$):

(Eqn. 9.) $d_e S = \frac{dQ}{T} = \frac{dU + pdV}{T} + (d_e S)_{matter}$ *and* $d_i S \geq 0$

Since the Second Law of thermodynamics is a local law, we can divide the system into *r* parts:

(Eqn. 10.) $d_iS = d_iS^1 + d_iS^2 + \ldots + d_iS^r \geq 0$

in which diSk is the entropy production in the kth part. Also:

(Eqn. 11.) $d_iS^k \geq 0$ for every *k*

This statement is stronger than the Second Law, because it states that the *entropy production due to irreversible processes is positive in every part of any systems, regardless of boundary conditions.* Local entropy production is defined as:

(Eqn. 12.) $\sigma(x,t) \equiv \frac{d_i s}{dt} \geq 0$

(Eqn. 13.) $\frac{d_i s}{dt} = \int_V \sigma(x,t) dV$

The nonequilibrium regime of thermodynamics can be further distinguished into the near-equilibrium, or linear regime, and the far from equilibrium, or nonlinear regime. In case of the linear regime the entropy production per unit volume can be expressed in terms of the forces and flows, such as the heat flow:

(Eqn. 14.) $\sigma = \sum_k F_k J_k$

where F_k are forces and J_k are flows. The forces drive the flows, and they disappear when the system is in equilibrium. In far-from-equilibrium systems, in the nonlinear regime, the thermodynamic flows are no longer linear functions of the thermodynamic forces.

In the linear regime the stationary states are those where the total entropy production:

(*Eqn. 15.*) $$P = \int_V \sigma dV$$

reaches a minimum, and

(*Eqn. 16.*) $$\frac{d_F P}{dt} = \frac{d_J P}{dt}$$

Where $d_F P$ and $d_J P$ represent entropy production due to changes in forces and flows, respectively.

Outside of the linear regime:

(*Eqn. 17.*) $$\frac{d_F P}{dt} \leq 0$$

The fact that $d_F P$ can only decrease does not tell us how the state will evolve, but e.g. in case of chemical reactions we have some absolute measures based on affinities that describe the extent of unbalance. Irreversibility has a double role: destroying near-equilibrium and creation of order far from equilibrium. In these far from equilibrium systems there are no extremum principles that uniquely predict the future state of systems.

Opposite to equilibrium systems where states evolve by minimizing a free energy, nonequilibrium systems tend to evolve rather unpredictably, because for a set of nonequilibrium conditions it is possible to evolve to more than one state. This is symmetry breaking at its best. Since the new state attained is often an "ordered" state, this evolutionary process is referred to as "order through fluctuations". These systems typically maintain continuous entropy production and dissipate the accruing entropy, thus we call them "dissipative structures". A basic feature of these systems is that they are able to amplify small fluctuations under certain conditions, this makes the system stable. Then through the larger fluctuation the systems finds a new state of different organization. The two terms *order through fluctuations* and *dissipative structures* cover the main aspects of nonequilibrium thermodynamics.

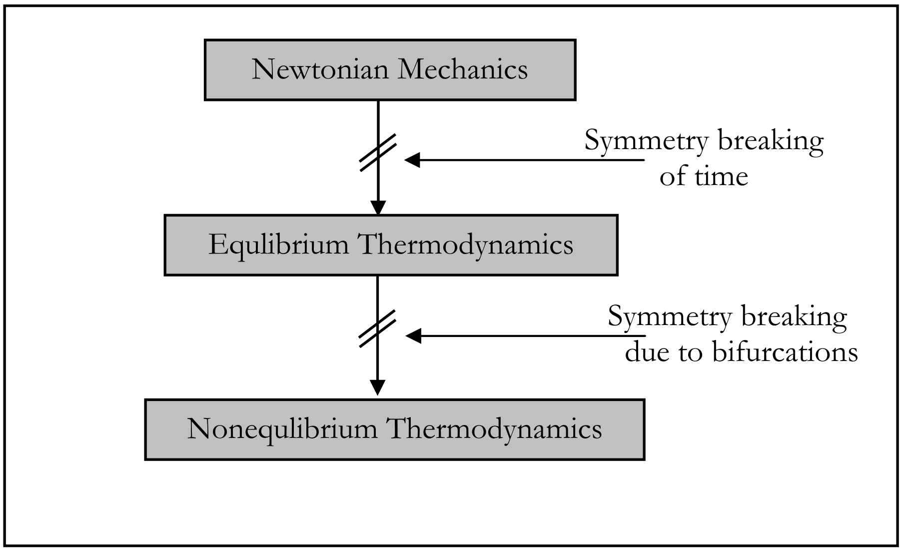

Figure 4-5 Symmetry breaking between the different levels of physical inquiry

These observations were made mainly in the realm of chemical reactions, however, they have far-reaching consequences in cosmology, materials engineering, various fields of physics and arguably in nearly all other disciplines including socioeconomic systems.

4.5.4 Summary of Dissipative Systems

In the first section of this chapter we have seen that Newtonian mechanics has had a strong influence on our thinking, enforcing the concept of time symmetry. This concept has various advantages, e.g. it is easy to calculate the state of a mechanical system at any time point, if we are in the knowledge of the initial conditions and the equations of motion. However, the study of thermodynamic systems revealed that the nature of these processes is not adequately described by Newtonian equations, and this realization resulted in useful abstractions, the introduction of Boltzmann's statistical entropy that gave up the desire for exact mechanical information about the particles of the system. This abstraction causes a symmetry break, as it is schematically explained in Fig. 4.5. Leaving the thermodynamic equilibrium behind we arrive at another symmetry break; far from equilibrium systems dissipate entropy and reorganize into new states not foreseeable before the appropriate bifurcation point.

4.6 Computation

The advent and widespread use of computers had a profound effect on the way we think about the word "computation". In common parlance a computer carries out computations, and these computations are the basis for everyday computer programs like word processing, spreadsheet calculations, games, internet browsers and electronic mail. These are all valid associations of computers with applications, but Computer Science has a lot more to tell about computation, computability, incomputability and models of computation that has lessons to teach for those attempting to use – digital, or any other type of – computers to extend the capabilities of the human mind and society. Computer Science is concerned about the techniques and methods of data processing, *not* computers.

4.6.1 Computers and Computation

Turning to the definition of the *computer* we may find that the emphasis on *electronic devices* is possibly a narrowing of scope. No one would argue that "the" computer is the boxy object sitting below our working desk, but going back to the original meaning of the word – i.e. a comput*er* is something or somebody that can compute – we may be able to abstract to many processes and organisms in nature which can carry out computation.

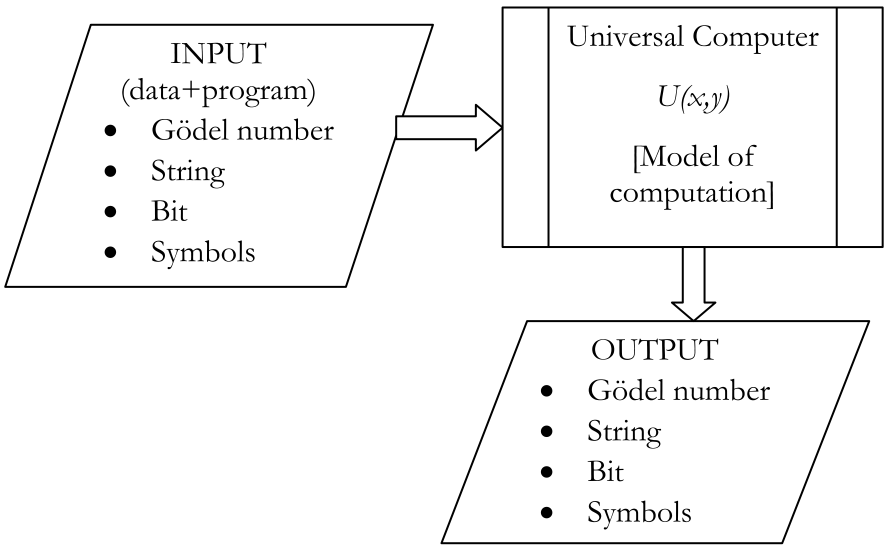

Figure 4-6 The concept of universal computation by Gödel numbers (and other media)

Such an overgeneralization of the computer might sound somewhat farfetched, however, such abstractions in general are often very powerful. As David Rogers put it: "*The most extensive computation known has been conducted over the last billion years on a planet-wide scale: it is the evolution of life.*"

To provide a more accurate definition of computation we have to turn to the so-called *Gödel numbers*. The *Gödelization* process is a mapping of several natural numbers into one single integer. The basis of this is the prime factorization of natural numbers. In case we would like to produce the Gödel number of the natural numbers x_1 and x_2, we have to use prime numbers p_1 and p_2 to carry them in their exponents, as shown with more general notations in the equation:

$$\text{(Eqn. 18.)} \quad \prod_{i=1}^{n} p_i^{x_i} = p_1^{x_1} p_2^{x_2} \ldots p_n^{x_n}$$

These numbers will be large, and therefore less practical, but we only use them for the purpose of definitions.

Definition The Universal Computer $U(x,y)$ is a medium for executing a program of Gödel number x on an input of Gödel number y and producing a result at the output. (See Figure 4.6).

As it is shown in Figure 4.6, the input and output can be anything from a bit or symbol to Gödel numbers, the Universal Computer will carry out the transformation of the input variable to the output according to its program.

4.6.2 Models of Computation

Based on the concept of the Universal Computer we can see that there is a mapping that is carried out by computations according to a program. This program can be an algorithm, a method, a series of instructions, but in any case it consists of elementary units, ingredients or rules. The characteristics of these ingredients determine the type of *model of computation* that describes how the mappings can be constructed.

One appealing feature of having numerous models of computation is what Church stated in 1941, i.e. *all computable functions can be computed by several equivalent models of computation* (actually he only speaks about Lambda Calculus, General Recursive Functions and Turing Machines)[51]. This is called the Church–Turing thesis, that is still not proven and impossible to do because of its fundamental nature. (A disproof would be a model of computation that can compute a function not possible to do by other models).

It is important to point out, that the equivalency of different models of computation provides us some freedom to choose the one most suitable for us. In the following subsections some of the models of computation will be shortly introduced, not so much for their immediate practicality in this work, but for the purpose of demonstrating their equivalence despite of their significant differences in representation. In Fig.4.7 the possible emulation of different models of computation by others is shown schematically.

Table 4-1The rules of General Recursive Functions

Zero	The *zero function* returns zero for any argument, e.g. $Z(x)=0$
Successor	The *successor function* adds one to its argument, e.g. $S(x)=x+1$
Projection	The *projection rule* allows that a general recursive function returns any of its arguments as the result, e.g. $P_i(x_1,\ldots x_n)=x_i$
Composition	The *composition rule* allows the construction of a new function by two or more functions. If $g(x)$ and $f(x)$ are general recursive functions, then so is $g(f(x))$
Recursion	*Recursion* allows recursive definitions. E.g. if $g(x)$ and $h(x)$ are general recursive, then so is $f(x,y)$ defined as $f(x,o)=g(x)$, for $y=0$, and $f(x,y+1)=f(h(x),y)$, for all other y.
Minimization	A general recursive function can be expressed as the minimization of another such function. E.g., if $g(x,y)$ is general recursive, then so is the function $f(x)=\mu y[g(x,y)=50]$ where μ is the minimization operator. (It is possible that there is no y that satisfies this equation).

4.6.3 General Recursive Functions

The rules found in Table 4.1 may be repeatedly applied to construct General Recursive Functions. The rules are either basic functions or rules for creating another function.

4.6.4 Turing Machines

A Turing machine is an abstract representation of a computing device. It consists of a read/write head that scans a possibly infinite one-dimensional (bi-directional) tape divided into squares, each of which is inscribed with a 0 or 1. Computation begins with the machine, in a given "state", scanning a square. It erases what it finds there, prints a 0 or 1, moves to an adjacent square, and goes into a new state. This behavior is completely determined by three parameters: (1) the state the machine is in, (2) the number on the square it is scanning, and (3) a table of instructions. The table of instructions specifies, for each state and binary input, what the machine should write, which direction it should move in, and which state it should go into. (E.g., "If in State 1 scanning a 0: print 1, move left, and go into State 3".) The table can list only finitely many states, each of which becomes implicitly defined by the role it plays in the table of instructions. These states are often referred to as the "functional states" of the machine [13].

A Turing machine, therefore, is more like a computer program (software) than a computer (hardware). Any given Turing machine can be realized or implemented on an infinite number of different physical computing devices. Computer scientists and logicians have shown that if conventional digital computers are considered in isolation from random external inputs (such as a bit stream generated by radioactive decay), then given enough time and tape, Turing machines can compute any function that any conventional digital computer can compute.

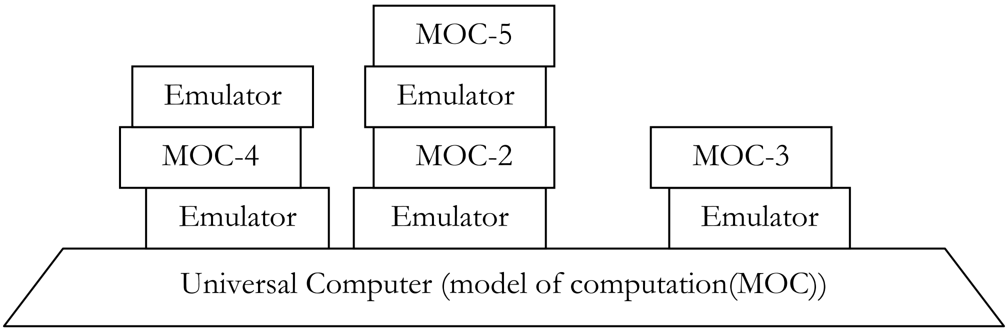

Figure 4-7 Emulation of different Models of Computation on a Universal Computer

Turing's interpretation of his ideas suggested that human intuition could correspond to incomputable steps in an argument.

Therefore the formulation of e.g. production networks in terms of a computational, information processing machinery offers significant advantages for the following reasons:

- easy and straightforward definition of the problem
- possibility for the exploration of computational limitations
- possibility for direct computer modelling
- application of the results of computer science
- direct connection to other methods and principles
- more possibilities for theoretical investigation of the model
- Related to the theory: because of the Gödel theorem any formalization of the systems in question are limited, so any of them describe only one part of reality.
- The Gödel limitation is overcome by the extended, metaphor-enriched science. [1] p.143.

The introduction of universal computation and models of computation lead us to the conclusion that the generalized concept of computation is a very good basis for the scientific investigation of large complex systems.

The benefits are:

- Universal computation is a mathematically clear concept
- The limits of computation is a thoroughly and rigorously studied and explored area

- The availability of relatively fast computers are an excellent media for carrying out experiments

Chapter 5 The ModNet Toolbox and Simulator

"Not only do we use instruments to give us fineness of detail inaccessible to direct sense perception, but we also use them to extend qualitatively the range of our senses into regions where our senses no longer operate"
P.W. Bridgman

The purpose of the *ModNet production network modelling toolbox and simulator* is to provide a Java class library for the Swarm ABM simulator that aids those who want to use the methodology of the next chapter, or just to create agentbased models of production networks, including machining equipment, orders, schedules, controllers, decision making entities (including humans) [40].[18]

5.1 The ModNet Toolbox

In Fig.5.1, 5.2 and 5.3 the main classes of the ModNet class hierarchy is shown. The Swarm classes are not indicated, the full Java documentation is available on the web[19].

The most important classes of the ModNet toolbox are the the following :Factory, Machine, Intelligence, Controller, Job, Operation, Workpiece, OrderStore, JobSchedule. This class hierarchy can be easily extended in many ways. Since the Intelligence class only includes basic heuristic rules (or "rules of thumb") it is convenient to use evolutionary computation to build new rules. A convenient way of doing it is to use a Java class library

[18] In the same research project this toolbox and simulator had been developed, a graphical Gantt charting tool and a problem generator was implemented [26]. However, in this thesis only the design of the basic class hierarchy of the toolbox and simulator is claimed to be an individual contribution.

[19] More extensive documentation and related material is continuously updated at available at http://www.pe.tut.fi/personal/patkai/software/modnet/

called Evolvica, that is freely available for research purposes[20]. This same tool can also be used to set up and optimize/evolve production network topology.

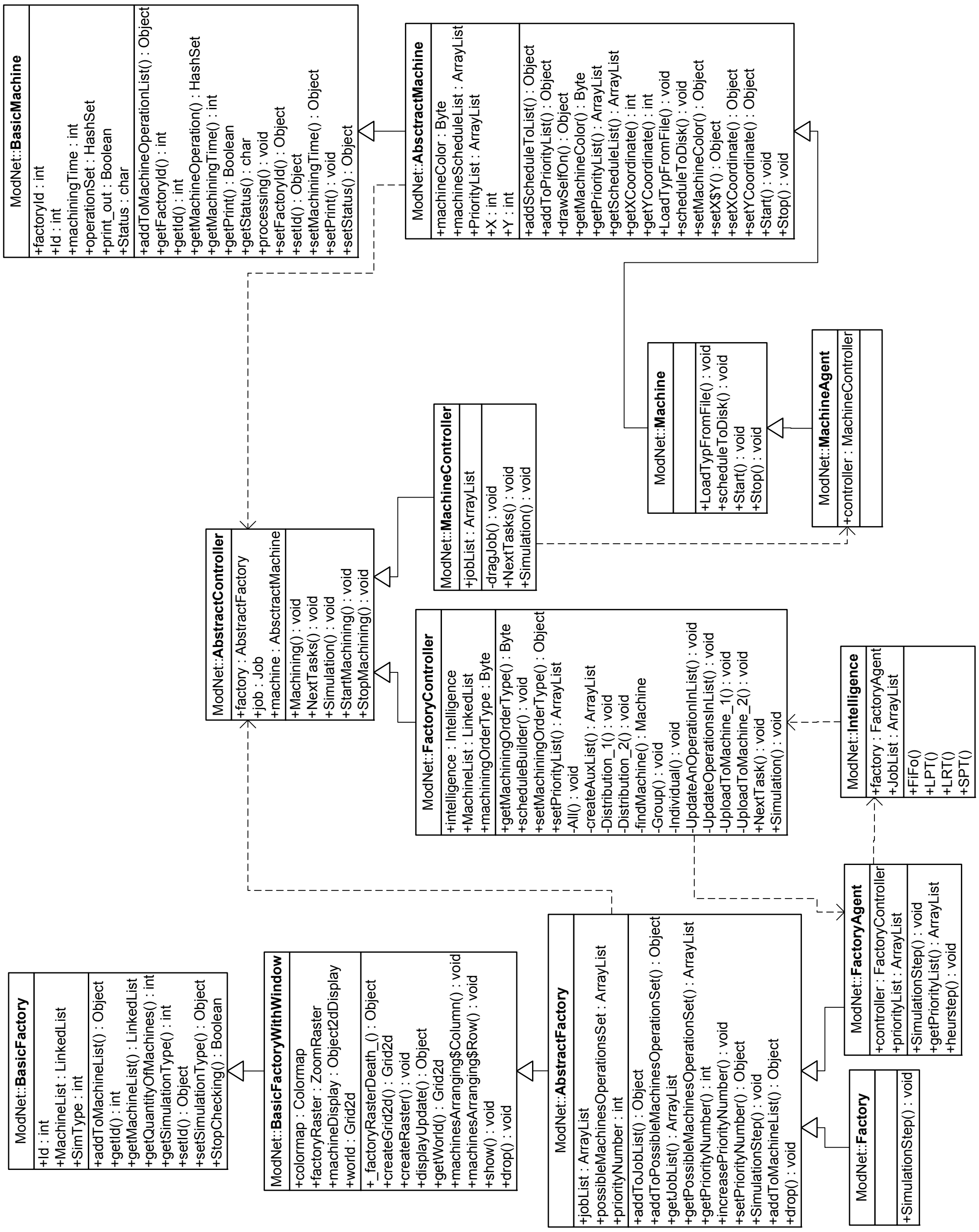

Figure 5-1 The class diagram of the core, interdependent classes in the ModNet toolbox

[20] Available at http://www.evolvica.org

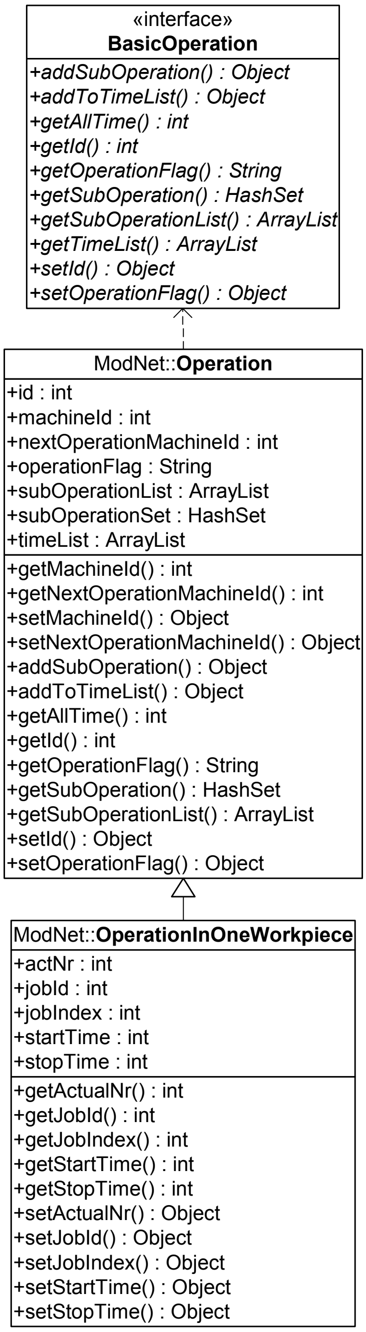

Figure 5-2 The class diagram of the Operation classes

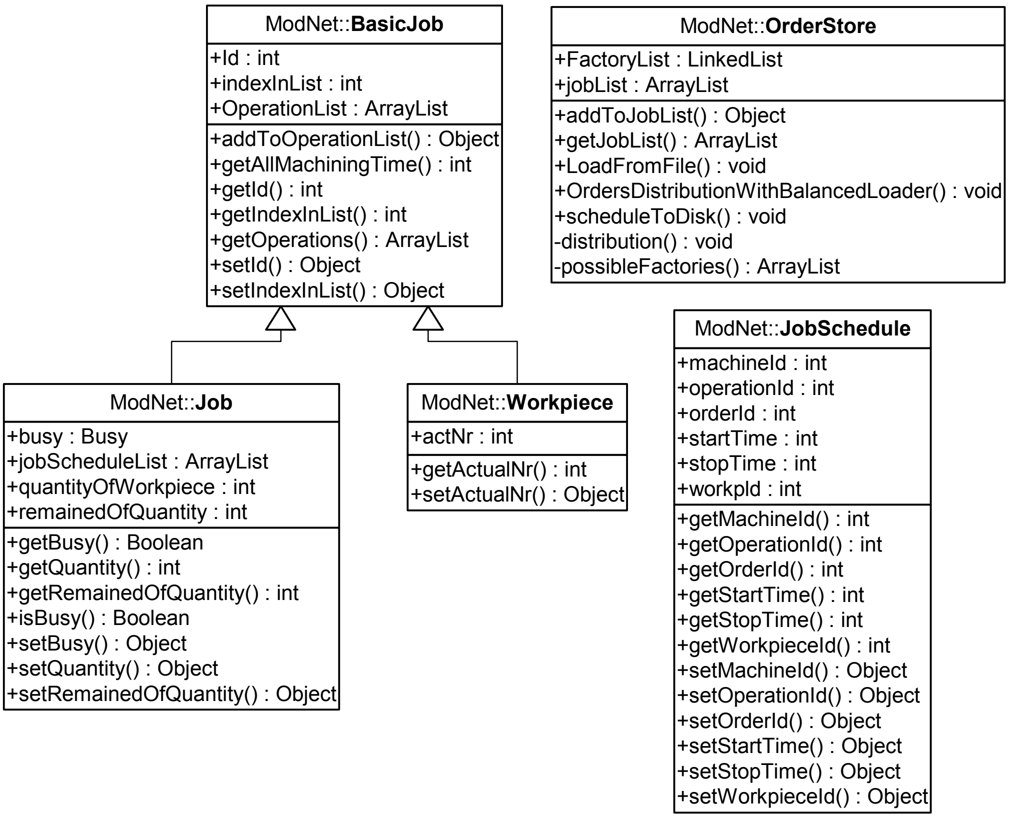

Figure 5-3 The class diagram of some other important classes in the ModNet toolbox

5.2 The ModNet PN Simulator

The ModNet toolbox in itself is only usable for those who can write programs in Java and in any case consumes time, requires debugging. To speed up and simplify the creation of agentbased production network simulations the toolbox is extended with some useful tools for programming-free setup. This method supports the definition of factories and production networks, machine types and capabilities, orders, basic process plans, several control options and operational modes. In the next subsections these are discussed.

5.2.1 Simulation Types

As it is depicted in Fig.5.4 , control in the simulated system can have at least three different kinds of arrangements. The following options are available (obviously by programming more options can be added):

- Centralized control

- Factory level control
- Distributed control

These three control methods resemble typical configurations that can be a subject of investigation by this simulator.

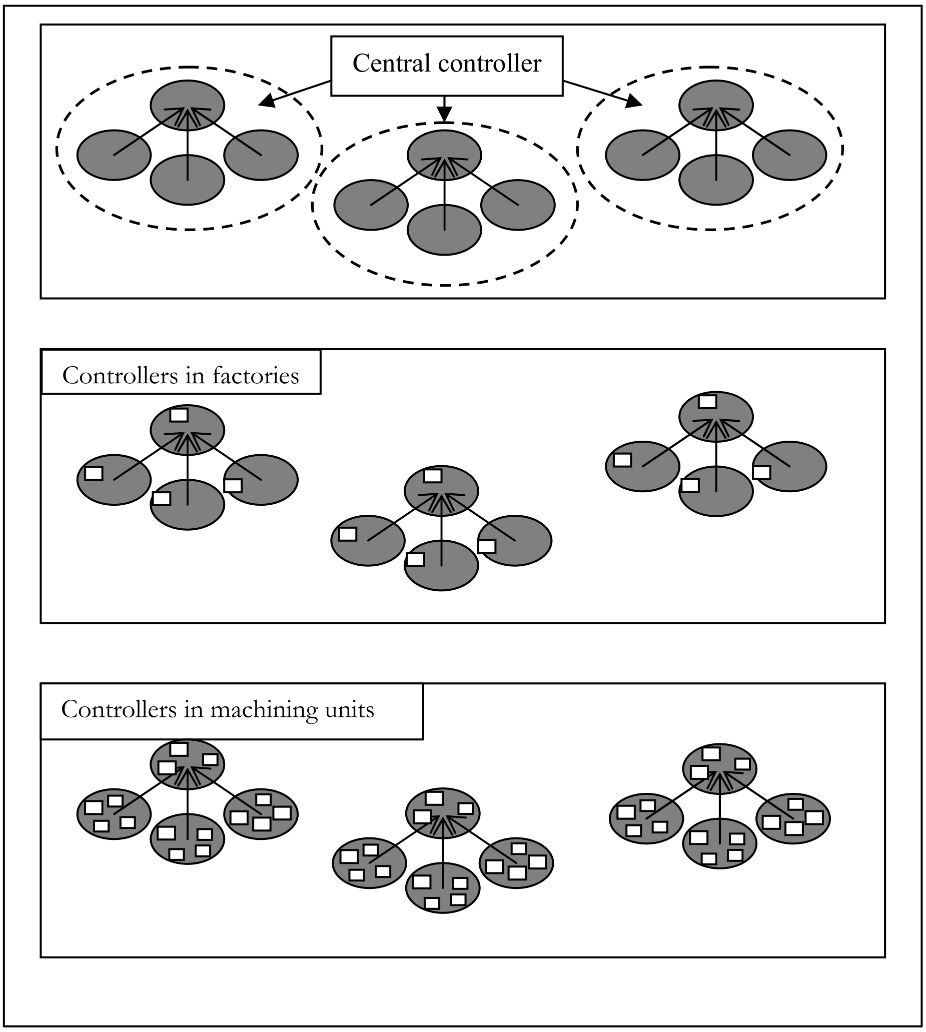

Figure 5-4 Preprogrammed options of centralized, decentralized and distributed control in production networks

5.2.2 Operational Modes

The operational mode is a constraint on the execution of jobs in the simulation. The following options can be set:

- Individual: *any workpiece can be processed whenever priority or physical constraints allow.*
- Group: *workpieces have to be entirely finished before the next one is processed.*

- Total: *in case there are multiple workpieces in a job, the next one can be started after all of these are completed.*

5.2.3 Configuration of the Simulator

The ModNet simulator is configured in three text files, and in case we want to use the automatic problem generator then a fourth one is used. All these files are .csv (i.e. comma separated), and their layout is shown in the following three tables.

The first step is to define the characteristics of the production network as shown in Table 5.1.

Table 5-1 Configuration of the production network layout

Simulation type Factory no.	Machine type code (MC)	Machine quantity (MQ)	MC	MQ	MC	MQ	MC	MQ
2								
1	1	5	2	10	3	4	4	10
2	3	10	4	10	2	6		
3	1	5	3	10				

In the layout table each row is one describes a factory. In the upper left corner there is a single code denoting the type of simulation (1-centralized, 2-factory level, 3-distributed). The odd numbered columns denote machine types and the even numbered columns following them denote the number of the machines from that type. This way we have a basic description of a set of factories and the method of their control. Despite calling the entities *machines* they are actually *actors,* and therefore they can denote any kind of equipment playing a role in production, even humans can have a code.

The definition of machine types and their capabilities is seen in Table 5.2. Along the machine (actor) ID we can see the code of the operations they can carry out. (Again, these can be non-machining, general operations, such as handling by a human. E.g. an actor/machine ID may be followed by human operation codes.)

Table 5-2 Machine description table

Machine ID	Cutting codes (capabilities of the machining centers)						
1	23	24	25	29	32	43	47
2	11	14	17	19	23	29	33
3	44	51	53				

In Fig. 5.3. the order definition file is depicted. This file is already preprocessed, in a real order database we don't immediately know the resource requirements, assigned machine types and machining/processing time units. However, from the point of view of our investigations this only makes the system simpler, without any practical loss.

Table 5-3 Order definition with resource requirements

Job quantity/code	Machining code (MC)	Machining time (MT)	MC	MT	MC	MT	MC	MT
1200								
12	10	23	20	25	10	32	20	29
11	12	21	4	27	20	28	30	29
23	10	19	20	29	30	33	70	44
28	10	29	30	31	20	41	50	51
3000								
23	20	12	10	32	20	25	10	29
21	4	11	12	28	30	27	20	29
19	20	23	10	33	70	29	30	44
29	30	28	10	41	50	31	20	51

After setting up a simulation – either by the simplified method or by programming – the Swarm machinery starts running and the orders are dispatched according to the chosen control strategy. This dispatching and scheduling is done in default by the simple dispatching rules, but new ones can be added or evolved by an evolutionary algorithm.

Also, the configuration of the production network can be changed dynamically. Depending on the goals of the investigation probes and metrics have to be introduced, especially in case of an evolutionary algorithm that needs an objective function.

Chapter 6 The Integrated Methodology

> *"Understanding what is going on around us is equivalent to building models and confronting them with observations. This statement may sound like a truism to a physicist or a chemist, but it goes far beyond physics and chemistry. At each moment our sensory systems scan the surroundings, the brain registers and compares the observations with respect to images already formed, and eventually reaches a preliminary conclusion. One of the basic steps in this procedure is the extensive use of analogies and archetypes. Physico-chemical systems giving rise to transition phenomena, long-range order, and symmetry breaking far from equilibrium can serve as an archetype for understanding other types of systems that show complex behavior – systems for which the evolution laws of the variables involved are not known to any comparable degree of detail. More important, in many of these systems the very choice of what should be a pertinent variable may well be part of the problem we try to solve."*
> *Ilya Prigogine*[21]

> *"It is a wonderful feeling to recognize the unity of a complex of phenomena that to direct observation appear to be quite separate things."*
> *Albert Einstein*[22]

After the attempt of the previous chapters to establish a scientific and metaphorical background for methodology development, in this chapter the synthesis of a *methodology* for complex system modelling is introduced. This methodology is integrating scientific and rhetoric knowledge by the systematic use of metaphors, analogies and agentbased models.

The methodology we describe is an emergent complex adaptive system itself. It is assumed that it benefits from the previously evolved schemata of different fields, and is due to further evolution and adaptation. One of the main intentions of this thesis is represented in the way we aim at providing an *integrating* framework, or a "meta-methodology" that can be the template for other, more refined methods. This is also emphasized in this chapter as it promotes general complex system modelling.

---

[21] Ilya Prigogine received the Nobel prize in chemistry for his discoveries related to non-equilibrium systems. Quote from [36].

[22] Written in a letter to Marcel Grossmann.

In the next sections the limitations of methodology development and the proposed methodology itself is presented.

6.1 Limitations & Challenges of System Models and Methodologies

Figure 6.1 shows a summary about the different number sets from the point of view of computability. This is also a summary about some of the limitations of computation as such.

The following sets are identified:

1) Recursive sets: *a set of numbers about which a program can decide if a number is a member or not.*
2) RE sets: *recursively enumerable, but not computable, or the "halting set".*
3) CO-RE sets: *computable recursively enumerable*
4) Not RE sets: *not recursively enumerable*
5) not CO-RE sets:*not computable recursively enumerable*

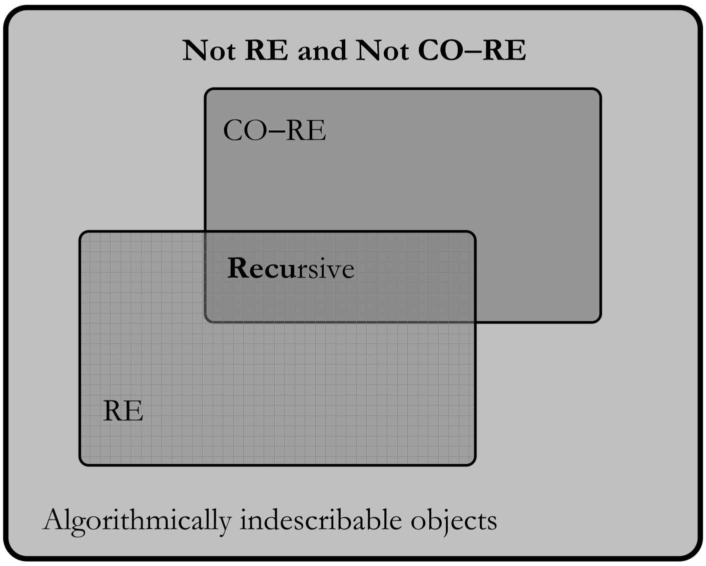

Figure 6-1 Classification of numbers according to their computability

Algorithmically indescribable objects are Not RE and Not CO-RE. These categories provide the basis for distinguishing between mathematical computations according to their usefulness. Here is the most serious limitation of computation, we assume that not all processes and phenomena can be described by computation. However, this is still a matter of debate amongst mathematicians. We also assume in this thesis that an associated methodology that integrates computation and *other* representations such as rhetoric has the potential of covering the whole problem.

Table 6-1The transition of basic paradigms and dogmas in contemporary science

Scientific dogma (common knowledge)	**Problem and time of appearance**	**New dogma**	**Consequence**	**Practical consequence**
Decidability	Gödel theorem, 1931 [19][20]	Undecidability	*Any formal language leads to paradoxes sometimes, we can only derive partial truths.*	We cannot develop a single theory for a reasonably difficult problem
Predictability	Chaos, Lorenz, 1963 [31][13][36]	Unpredictability	*A very small change in the input variables can mess up our predictions totally.*	Long term forecasting is mostly impossible
Explainability	Complex emergence, Kolmogorov, Chaitin, 1960's [13][36]	Unexplainability	*The components of the system do not explain its collective behavior.*	We can have difficulties understanding the situation, *common sense stops working*

Chapter 3 provided some insight into the nature and characteristics of "systemic" scientific endeavors, but falls short of the great expectations for practical applications. The reason for this is that handling (i.e. understanding, modelling, simulating, controlling) large, complex, heterogeneous systems is a challenge incomparable to problems modeled by causal mechanistic equations, and is even impossible, because the complexity of complex adaptive systems that include humans and machinery cannot be captured by a single exact model borrowed from the hard sciences. In *Table 6.1* we can see a summary of the key problems that strongly influence all the present and future solution methods trying to gain control over such systems.

The earliest of these is the Gödel theorem that has shaken mathematics and the belief in unifying formalisms that embrace and perfectly solve a problem area.

The actual consequences are crucially important, because rhetorically formulating it provides an explanation why in practice there have are always been competing theories even in specialized fields. In the next chapter the metaphoric bridging method provides a way to get around this problem, by acknowledging it first and filling in the gaps by rhetoric (despite the fact that rhetoric as a tool has a low respect amongst scientist and especially engineers, because it is very easy to abuse to cover the gaps instead of bridging them).

The observation of chaos in meteorological calculations by Lorenz has resulted in the unpleasant knowledge of another limitation, i.e. the impossibility of long term prediction of system states. This directly modifies the objectives of the research we do, and we are no longer aiming at exact, quantitative forecasts, but use our models in a less direct way to explore possibilities, to "harness complexity" as Axelrod puts it.

The so-called "reductionist" scientific approach is often criticized that translates observations to mechanistic equations. In spite of this fashionable criticism no-one denies that equation-based analytical models are more desirable than "soft" models involving significant uncertainty, fuzziness and the lack of causality. The corresponding methodology aiming at minimally an increase in the understanding of such systems. As in case of the previous two problems, this can only be achieved by a modification in goals and approach.

6.2 Knowledge Transfer Between Models by Metaphors

In Chapter 2 the importance of metaphors as the bridging material between different models was described, and in *Fig. 6.2* it is shown in more detail. The basic knowledge that any scientific or computational model can represent only a segment of reality – as reflected by measurements, observations and models – requires that we fill the gaps between them by our continuous mental– and the corresponding rhetorical models[23].

Using the scientific language and method observations and measurements are put into a mathematical model, assuming that the experiments are repeatable and reproducible by others. This satisfies the requirement of objectivity. Part of the process is the validation of the model through numerous experiments that are usually carried out by other parties ensuring that no subjective mistakes are generated during experiments and measurements.

The "third" language or symbol system of computation [37] is in many ways related to the language of mathematics, as it can represent equations and solve them with some limitations related to continuity and problem size. However, computation also became a very flexible and multi-purpose tool due to the ubiquitous availability of computing machinery. Various modelling approaches make it possible to create models that are not possible to describe by equations. One of these possibilities is agentbased modelling – mentioned in chapter 3 – that approaches modelling from a quite different aspect. Parunak [39] describes the advantages of agentbased modelling and points out that they represent a different aspect of reality, since equation based models represent the relation of observables and agentbased models represent the internal behavior of the modeled individual.

Following the process shown in *Fig. 6.2* we can model reality or natural phenomena (including manmade systems) and exert a varying amount of control over them, depending on the success of the model. Since we know that in case of complex problems none of our models will cover the whole and provide a grip on the controls we have to make use of different models and either integrate them or use a different one in different circumstances. The knowledge of the possibility about building bridges by rhetoric is not new at all, but its use is mostly unconscious or not systematic.

[23] This strictly means the gaps in between multiple theoretical and compuational tools, not in physical reality.

The method for transferring knowledge between different models and even has long been part of scientific discourse. The method, however, is described by e.g. Nicolis and Prigogine [36] and by others [1], in the realm of complex systems where such knowledge transfer becomes crucial, since we have systems that require us to be approached in a systematic and knowledge integrative manner. In *Fig. 6.2* this algorithm is shown in an area enclosed by dotted line.

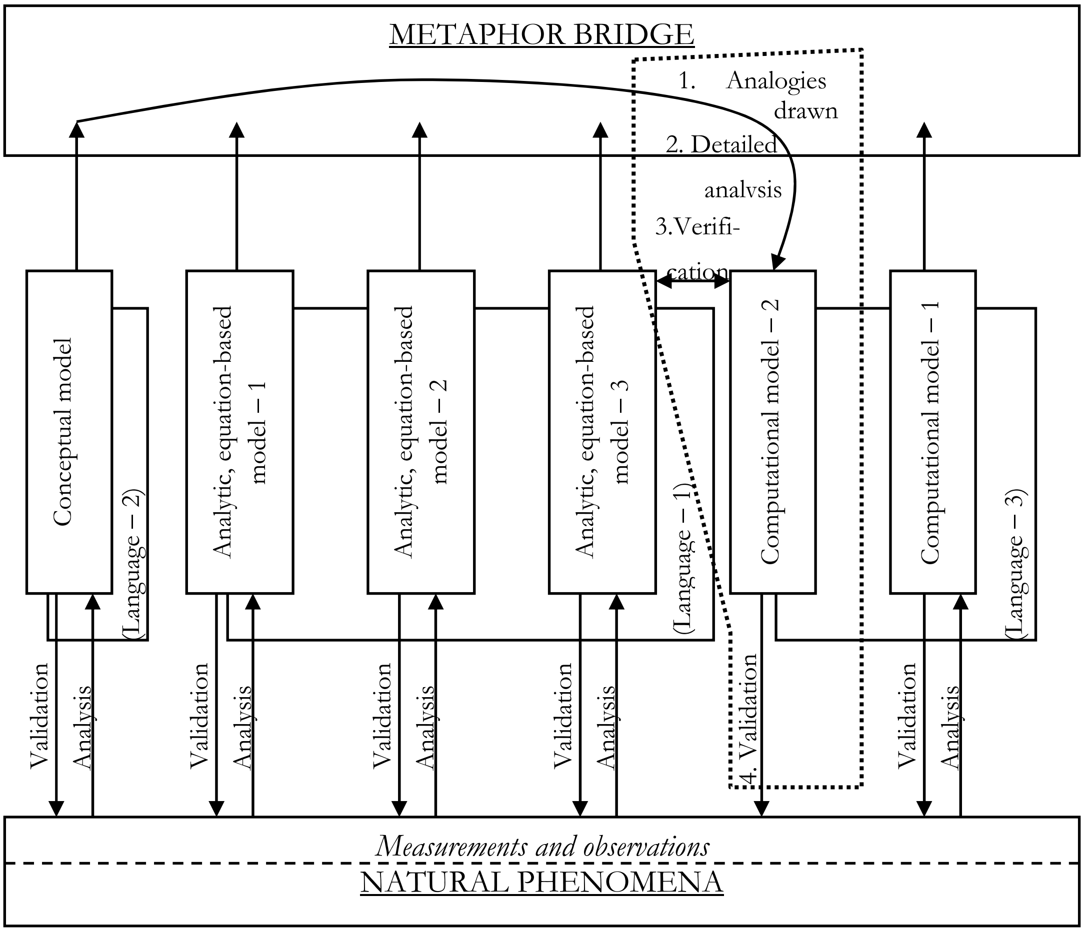

Figure 6-2 Transferring knowledge between models by drawing analogies

6.2.1 Knowledge Transfer Example – the Metaphor and Analogy of Evolution

A good example of this knowledge transfer is the evolutionary computation methods introduced in chapter 4. In case of the *"evolution metaphor"* the knowledge transfer process is as follows:

1. Darwin analyzed species and came to conclusions about the basic process of evolution

2. Microbiologists developed analytic methods down to the genetic level and showed how genetic inheritance works, defined such concepts as population genetics, crossover, gene mutation, selection pressure and process.
3. Computer scientists found that some of the problems they face cannot be solved by exact mathematical equations or algorithms, and realized that it is necessary to approach these problems heuristically.
4. Somebody had the idea that looking for better candidate solutions is metaphorically similar to the evolution of species.
5. The evolutionary concepts in (2) were given meaning in the realm of computing.
6. The evolutionary algorithm was assembled and it was observed that simulated evolution is able to develop individuals by selection, crossover, mutation, etc.

It is not sure, that such a knowledge transfer always works, but the general observation is that interdisciplinary cooperation is mostly very fruitful[6]. In the same time this emphasizes the importance of a methodology that directs us in making such experiments.

6.3 Synthetic System Modelling with Knowledge Transfer

Synthesis is often regarded as a secondary approach to analysis, not a complementary or equal alternative [39]. However, searching for new structures, new knowledge, imaginary-, nontrivial systems and in case we have no other way for the investigation of a mechanism (with possibly explanatory power) synthesis provides a viable alternative.

A characteristic example of synthesis is the modelling of cellular automaton, where we have no analytical knowledge about the emergent phenomena, at most heuristic rules, which were formed on the basis of extensive computational studies[24].

[24] Chris Langton introduced the λ parameter that hints whether the cellular automata will be chaotic, repeat a pattern or be random [13].

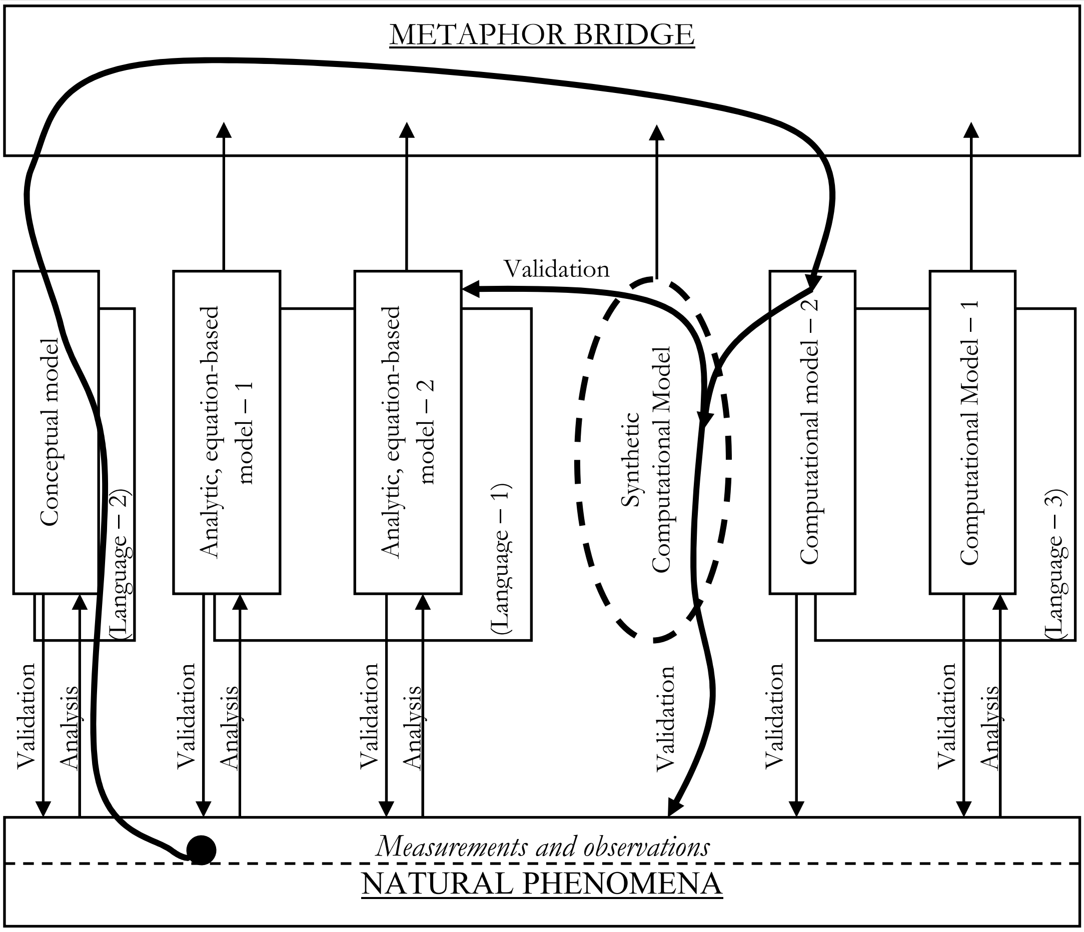

Figure 6-3 Computational Synthesis of System Models

6.3.1 Bottom-up System Modelling

In *Fig. 6.3* a computational synthesis method is shown and explains the idea how agentbased models can fit into the integrating scenario. Model building itself is not science without a methodology that insures the required objectivity, the repeatability of experiments, applicability and predictive power. By following the thick arrows in the figure we can see the following steps of the method:

1. Formulate conceptual entities by abstraction from measurements and observation of a phenomenon.
2. Extract a metaphor from the conceptual model.
3. "Translate" the metaphor into a computational component model, i.e. *agents*. (At this point there are many possibilities, there is plenty of space for creativity.)
4. Assemble a synthetic computational model, i.e. a simulation model. (Choosing the parameters and the simulation scenarios again provides many custom options.)

5. Run the simulation model and explore it with various configurations, parameters.
6. Validate it against other type of models.
7. Validate it against reality.

By this procedure we have integrated an empirical, synthetic (meaning bottom-up) modelling method in a scientific environment. This integration doesn't mean that we have a flawless scientific method, but it is not far from it, because it provides a systematic way of handling computer experiments and uses analytical and rhetorical knowledge to put the model into context, validate it and use it for gaining extra insight into systems.

6.3.2 Methodology

In *Fig. 6.4* a detailed view of the empirical methodology is shown from a different aspect.

This methodology helps avoiding some of the trivial problems of agentbased, synthetic, bottom-up modelling.

1. The first step is to put down in written form what we expect of the model, because building in expectations into a model is very easy. By making these intentions "transparent" we admit our skepticism concerning the model. This methodology leads to its applications also, because − as it can be expected from the scenario of *Fig. 6.2* − we do not expect that such a model will be omnipotent and able to replace the equation-based, mathematically sound models. Agentbased modelling and synthetic methods have the advantage that they can produce models that would not have been thought of by humans, and by running various open-ended evolutionary simulations many malfunctioning structures can be "explored" as well [31]. This is why the complementarity of such a modelling methodology is emphasized.
2. Document the problem and its environment − i.e. more than the problem − targeted *by* the model. This indicates that the description we are aiming at is usually more than the pure problem we know, therefore it is desirable to have a descriptive documentation about the environment, (e.g. what does the simple mathematical formulation of a job-shop scheduling problem tell about how the workers use the

system on the shop floor? Or what do the route and time scheduling of trucks tell about issues like speeding, alternative routes used, reversing from one-way streets?) A good documentation gives clues how much we can rely on the formal description of the problem.

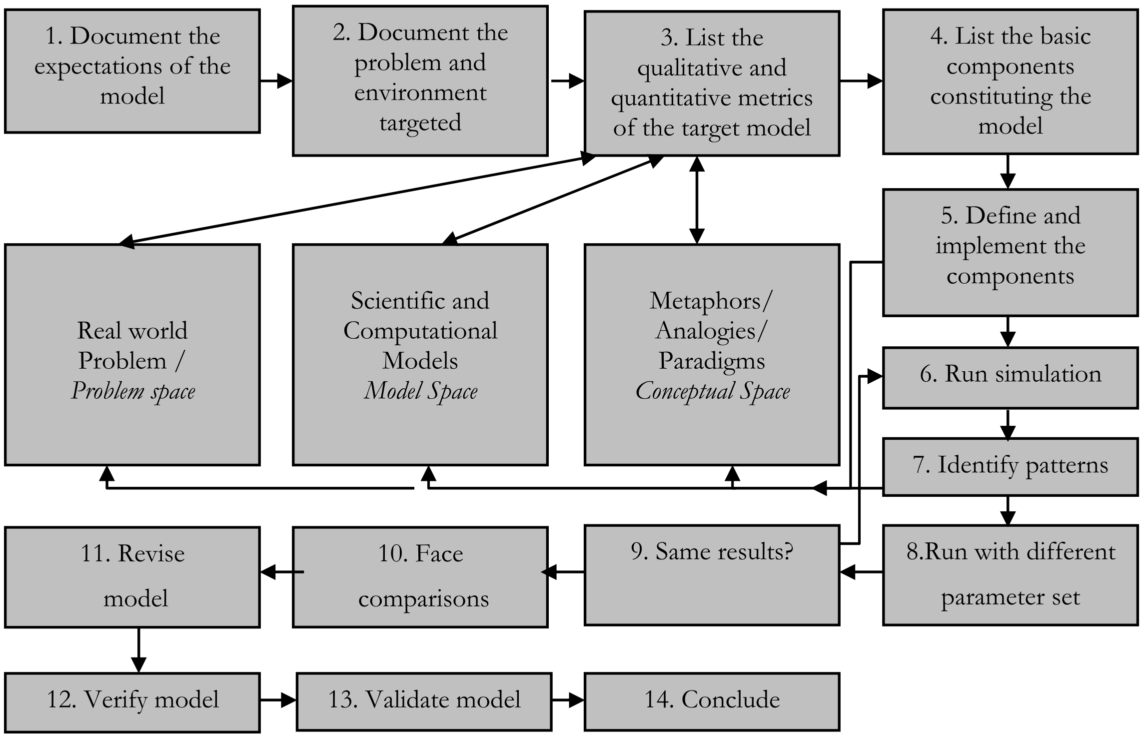

Figure 6-4The computational system model synthesis methodology

3. The fact that we are making models itself suggests that we have certain metrics that we are aiming at influencing. These have to be at least qualitative but more desirably quantitative. These metrics are often not trivial, e.g. when talking about complex adaptive systems the observation and quantitative description of *emergence* is a serious challenge.

4. From the previous figure we have seen that for a bottom-up model we need to find elementary components, agents, to be able to build a simulation. This selection process has some arbitrariness, but often a very similar selection is made by different people.

5. Define and implement the components.

6. Run the simulation with an initial setup, i.e. parameters, initial and boundary conditions.

7. Identify patterns that have the potential to tell something about the defined metrics in 3.

8. The computational exploration of the model with different parameters.

9. In case the model shows responsiveness to parameter tuning it has to be explored as well as possible.

10. Face comparisons: the results of the simulations can be simply compared by looking at the results, and visually deciding whether they follow a pattern.

11. Based on the results so far the model can be revised.

12. Verification of the model can be done e.g. by implementing it on another simulation platform. This step also depends on the software development methodology.

13. Validation can be done by comparing the results to equation based models or any quantitative results.

14. Concluding the work involves documentation, comparisons with the initial expectations.

This methodology based on the processes in Fig.6.2 and Fig. 6.3 integrates agentbased modelling (involving the "third language") rhetoric knowledge and validates against equation-based models therefore represents a universal tool for modelling and simulating complex adaptive systems.

6.4 Chapter Conclusions

- Some limitations of systems inquiry approaches have been drawn
- It is claimed that metaphor exchange between different analytical and computational models is a common, but formally and consciously rarely discussed method
- The abstract, generalized synthesis algorithm was introduced

- It was concluded that the introduced and formalized methodology has a looser connection with other models therefore its use and role is different and complementary
- Summarized the application possibilities of the integrated complex system model synthesis methodology

Chapter 7 Complex Adaptive Production Networks

"It is not so much important to be rigorous as to be right. "
A.N.Kolmogorov

In chapter 3, Fig.3.4 it was shown that to arrive at a scientific model a concretization of metaphors and analogies is necessary. In case we find a metaphor that seems to capture that essence of a problem we can refine it and transform it into a conceptual model. This model can be advanced even further, if it is put in a scientific, rigorous form of mathematics or logic. In case of complex systems very often the observer is stuck with a metaphor or a conceptual model, and the formation of a scientific model eludes him. Based on the experience of the complex systems research community – described in chapter 3 – it is quite usual that the "1)metaphor-2)conceptual model-3)scientific model" process becomes problematic in the last step and a lot of effort is consumed in building agentbased simulations that actually don't meet the scientific criteria. In the next sections an attempt is made to use the metaphor of complex adaptive systems and dissipative structures to form a basic conceptual model and show their analogies with production networks. Together with the ModNet toolbox and the methodology of the previous chapter this forms the original contributions of this thesis. The completion of the process would mean that we prove that dissipative structures are isomorphic with production network problems, but this step is not carried out because simulation is not an adequate tool to make such a proof. However, the analogy itself has some value and the suggested future work is likely to find some use of this conceptual mapping.

7.1 Production Network Modelling

One of the central issues related to this thesis is the computational modelling and synthesis of production networks. Computation and reproducible computer models provide the objective basis for discussion, and since it is very expensive – and directly impossible – to carry out scientific experiments with real enterprises we are challenged to find computational models.

Chapter 4 of this thesis has introduced some basic scientific paradigms aiming to show the possible metaphors and limitations, and we attempted to provide a computation-based framework that matches the basic requirements for a synthetic study.

This section is the application of the previous one for production networks synthesis, and therefore builds on the establishments of the integrated methodology.

7.1.1 Complex Adaptive Production Networks

It is always important to remember that modelling is an activity that is strongly influenced by our minds, the theories we know, the goals we and our society pursue. This is why it is so important to aim at objectivity, develop several scientific, engineering or management methodologies so that the many facets of reality are all represented. Also, it is important to note, that the computational models we build and talk about are only the result of the interaction of software and computing machinery, i.e. they rely on the basic assumption of scientific activity that our world is strongly interconnected and the "truths" found in a discipline have a place in another one, and as we put it in previous chapters: it is possible to experience the unity of knowledge through the use of metaphors. In common parlance it means that we *believe* that our observation at a certain time and place getting to know e.g. a simulation model's behavior we also learn about the modeled system that it represents.

In chapter 2 we talked about production networks and throughout the thesis complex adaptive systems and their computational models were often mentioned. Because of their similarities and mutual interests it is plausible to connect the two and propose a definition of such production networks that strongly resemble complex adaptive systems. In *Table 7.1* we can see some of the parallels of complex adaptive systems and production networks.

Table 7-1 Similarities of Complex Adaptive Systems and Production Networks

Complex Adaptive Systems	Production Networks
Constructed of a great number of independent units, i.e. agentbased	Consist of a hierarchic system of supply chains, factories, manufacturing cells, machining centers and equipment with their own operators, including different levels of scheduling
Agents are heterogeneous	All production units from the bottom up are different in capabilities, decision making rules, size, quality, efficiency
Agents adapt to their environment	Production units can be reorganized, personnel changed, equipment modified, reorganized
Dynamism – Chaos	Because of virtuality, fluctuating demand, cultural, political, transportation problems the bullwhip effect creates a very dynamic, often chaotic system (by chaotic we mean very small change causing very large effect)
Emergence of complex patterns	Though rules are simple at the micro level, the complexity of the whole can be large. Many of the features of a system are not planned, they merely *occur,* i.e. emerge. These features can be desirable or undesirable, too.
Positive feedback – pressure to reorganize	Erroneous policies or business environmental factors can accelerate unwanted change and break the system – and then it reorganizes after bankruptcy, ownership changes, selling some production units, etc.

The main reasons for constructing a definition of complex adaptive production networks are:

- it is an aim of this thesis to bring the vocabulary of complex systems closer to that of production networks
- in either modeled or real production networks complexity and adaptation have a very significant role
- such a name distinguishes and emphasizing the kind of research carried out in relation to it
- also emphasizes that we assume that production networks are complex and adaptive, in cases where it is possible to simplify them or complexity is not an issue we are not interested in them

Definition: *A "complex adaptive production network" – CAPNET – is an abstraction of networked production systems that can be a real organization or a computerized production network model exhibiting the characteristics of complex adaptive systems.*

With this definition and concept we direct the focus of our research of production networks to complexity and adaptation.

This view is already represented in the methodology and provides a tool for achieving the goal described in [31] by D. Levy as:

"By understanding industries as complex systems, managers can improve decision making and search for innovative solutions. …Chaos [complexity] theory is a promising framework that accounts for the dynamic evolution of industries and the complex interactions among industry actors. By conceptualizing industries as chaotic systems, a number of managerial implications can be developed. Long-term forecasting is almost impossible for chaotic systems, and dramatic change can occur unexpectedly; as a result, flexibility and adaptiveness are essential for organizations to survive. Nevertheless, chaotic systems exhibit a degree of order, enabling short-term forecasting to be undertaken and underlying patterns can be discerned. Chaos [complexity] theory also points to the importance of developing guidelines and decision rules to cope with complexity, and of searching for non-obvious and indirect means to achieving goals."

7.1.2 Dissipative Structures

In addition to the complex adaptive systems paradigm, the dissipative structures introduced in chapter 4 show promising characteristics, which hint that drawing analogies between them and developing the ModNet toolbox further in that direction is potentially useful.

The characteristics of dissipative structures are the following:

1) The systems considered are open
2) Positive feedback dominates in the reorganization of the state
3) They maintain continuous entropy production and dissipate the accruing entropy
4) They emerge from dissipative selforganization
5) Autocatalitic
6) Autopoietic: self-referential, the function of the system is to renew itself
7) The dissipative structure finds and maintains its form and size independent from the environment
8) Dynamic nonequilibrium structures never achieve absolute stability
9) Fluctuations continuously "test" the stability of the structure
10) The environment can enforce stability even when the subsystem would be unstable
11) The farther we go from thermodynamic equilibrium the more choices become available to find a metastable dissipative structure
12) The principle of maximum entropy production: the system is procuring high entropy close to its instability, where it forms a new structure, but production of entropy is low close to an autopoietic stable state

The process of dissipative structure formation and evolution is the following:

1) Shift starts from near equilibrium conditions due to openness, entropy production is low
2) System enters the linear nonequilibrium regime, thermodynamic forces and flows are in balance
3) System enters the nonlinear nonequlibrium regime, entropy production is high(!)
4) System reaches a bifurcation point, goes beyond and finds a new state
5) Entropy production is lower, the system manages the state or goes for another reorganization

7.1.3 Metaphor Transfer Possibilities to CAPNs

As one of the major themes of this thesis is the integration of knowledge gathering tools, in *Table 7.2* a collection is provided about the possible metaphor transfers from concepts to computational models.

Table 7-2 Metaphors and their application in CAPNs

Metaphor	Implementation
Dissipative structures	Not yet implemented. Strong bounds with complexity is promising.
Evolution	Implemented, available standard tools[25]. The ModNet simulator can be extended with evolution, agents can be evolved (removed/added to population). By genetic programming we can also evolve decision rules easily in e.g. Scheme or Lisp.
Classifier system	Indirectly implemented. A ModNet simulation – in case it is evolvable – naturally behaves as a CS, however, the schemata are not easy to analyze and follow in this architecture. A transparent method of handling schema evolution would be desirable.

Based on information in chapter 4 it is apparent that evolution is a very well researched metaphoric knowledge transfer method while dissipative structures were only recently proposed and have not received proper attention [36], and especially no application in the field of production networks. The reason for this is most probably the straightforward application possibilities of evolutionary search methods for optimization problems, helped by the inherent abstraction of these algorithms in the separation of the geno- and the phenotype. Also, evolutionary computation methods started to gain attention in the 1960's, while dissipative structures were only suggested for such a use and became known in general only a few years ago.

In case we would like to model production networks as dissipative structures a series of concepts would have to be translated into other terms, and the appropriate ("modelable") aspect of CAPNs to be found. Since dissipative structures dissipate entropy, information

[25] A good example is found at www.evolvica.org an advanced Java toolbox for Evolutionary Computation.

entropy could be an applicable idea, and all the actual processes and transformation of matter, energy and information could be handled as information. However, this is only speculation about application, in the framework of this thesis these ideas are not worked out in detail.

7.1.4 Chapter Conclusions

This chapter has:

- Listed the analogies between complex adaptive systems and production networks
- Defined complex adaptive production networks
- Listed some potential and partially implemented or at least initiated metaphor transfer methods to production network modelling

Chapter 8 Summary, Conclusions and Outlook

"Vladimir Nabokov's conviction rings true: "What can be controlled is never completely real; what is real can never be completely controlled"".[26]
Ilya Prigogine

"Plasticity is a double-edged sword: the more flexible an organism is the greater the variety of maladaptive, as well as adaptive, behaviors it can develop; the more teachable it is the more fully it can profit from the experiences of its ancestors and associates and the more it risks being exploited by its ancestors and associates."
Donald Symons[27]

"All models are wrong, some are useful."
R. Sargent

The motivation behind this thesis has been to face the challenge of newly appearing degrees of freedom in the globalizing arena of production networks. Because of the complexity and fuzziness of the problem set related to complex systems/production networks we considered the problem itself and solution methods complex adaptive systems. In the following sections the original objectives of chapter 1 are listed and some evaluating comments added.

8.1 Comments about Meeting the Objectives

1) **Establish the epistemological background for the research, including the critical view of science and rhetoric**

   The basic Aristotelian scenario of scientific and rhetorical knowledge was presented, also the so-called "third language", i.e. computation was added with the remark that computation is not essentially different from scientific knowledge, it has strong links to it, they are intertwined. It was claimed that the integration of these methods is rarely addressed in scientific papers and that conscious integration should provide an advantage in systems investigations.

---

[26] Ilya Prigogine: The end of certainty, p.154. [41]

[27] Gary Flake,: *The Computational Beauty of Nature,* pp.361. [13]

2) **Define a new problem class based on the observations made related to complex adaptive systems and production networks**

   Several problems and problem classes were introduced in narrative. Complex adaptive production networks as an abstract problem class have been defined. This definition puts proper emphasis on production networks as complex adaptive systems and focuses attention on complexity and adaptation, two crucial questions in case of the management of large interconnected systems.

3) **Survey state-of-the-art methods and tools related to the identified problem, formulate this from an original point of view to make it a minor original contribution**

   A short introduction and list of software, management and analytical methods was provided. It was emphasized that in the area of complexity and adaptation investigation there are only a limited number of tools available, and especially in case of agentbased modelling there are no properly worked out and tested methodologies aiming at generality (ABM is an "art"). Commercially available tools are typically limited to controlling, information storage and analysis, and sometimes provide what-if analysis features, but open ended evolution and systematic synthetic simulations are not supported, except in the jESevol package that is at testing stage.

4) **Identify and explore scientific tools showing potential for successful application either as exact methods or as applicable metaphors/analogies**

   A multidisciplinary study has been carried out in the fields of physics, computer science, biology and "systems and complexity". The studied areas are all rich in well understood paradigms, potential analogies – all of them in the fields of basic research of natural sciences waiting for intuitive knowledge transfer methods. There are certainly many more scientific results that could have been mentioned and used.

5) **Identify the limitations of the methodology with special attention to computational and philosophical issues**

Computers as the media of simulational investigations have strict limitations, even though the available computing power on a chip is doubling about every 18 months for several decades now and networking offers an even greater power for distributed computing. The real limitations – however – are not in the clock speed of microprocessors, but in computability. It is of major concern whether a computer can represent all the solutions for a problem and how it can find the good ones. It can be concluded that computation by computers is a valuable resource that is by far not exhausted yet, and in spite of its limitations there is plenty of potential in computational methods. On the philosophical side the transfer of knowledge by analogies is a good possibility and the proposed methodology builds on it.

6) **Formulate an integrated methodology for buttom−up complex system modelling**

   The integration of the two main types of knowledge has been proposed in a computational framework by incorporating rhetorical, verbally formulated methods in various components of the simulation model. The components of this methodology are not new inventions, but their integration is shifting the unconsciously and unknowingly used methods into an organized one that can be a subject of further research itself. There are many aspects of this methodology − especially applications and refinement − that are not completely worked out and belong to future plans.

7) **Design an agentbased modelling software library that enables researchers to run comparable and reproducible simulations of distributed production systems with different distribution of control**

   The ModNet agentbased production network simulator was introduced in chapter 5. This software library has components/agents and collections that represent production network components and make it easy to build agentbased models.

8) **Describe an analogy transfer in detail, showing how knowledge is transferred through a scientific metaphor into a model**

There are several analogy transfer examples in the manuscript. The evolutionary is described in chapter 6, and the complex adaptive systems and dissipative structures analogies introduced in chapter 7.

8.2 Summary of Arguments in the Thesis

The list of arguments below is a summary of the claims made in this manuscript. They represent and summarize the key issues, opinions and realizations related to the topic of this work. Not all of them in this list are original, but together they represent the originality of this dissertation.

I. **Complex Adaptive Production Networks:** During their recent evolution – driven by fast transportation, communication, technological possibilities, political and economical stability – production networks became nonlinearly coupled distributed systems best described as complex adaptive systems, shifting the emphasis of their scientific investigation from optimization to complexity and adaptation.

II. **Complex adaptive solution systems:** The different languages of inquiry can be integrated by a flexible rhetorical establishment. Specialization has created large gaps between disciplines, computational models are disintegrated, therefore the bridging is only possible by a cognitive-conceptual model, where rhetoric and metaphors are the glue. The key to a good solution is successful integration of science, computational models and rhetoric. Scientific models cannot fully describe social- and business systems, but they can be (1) part of the model and (2) provide metaphoric knowledge through analogies.

III. **Exchange of knowledge between models by metaphors:** Metaphors provide an ideal medium for exchanging knowledge between models. The method of metaphoric knowledge transfer is used in practice since the birth of science, but it is usually not formally integrated into scientific methodologies.

IV. **Agentbased Modelling is the most suitable paradigm for the implementation of the presented methodology:** The most suitable programming/modelling/simulation paradigm for the investigation of complex adaptive systems is ABM. This should be part of this system of solutions. ABM is *not* the goal, the most important or central tool of complex adaptive systems research, it can only be part of the exogenous utility network we use to gather knowledge and support decisions.

V. **Agentbased Modelling in itself is not valuable:** By such a flexible computational method it is mostly possible to prove and disprove the same claims. ABM needs a methodology that embeds it and gives meaning to the simulations.

VI. **Agentbased Modelling is just as limited computationally as other programming paradigms:** Programming an ABM produces code that could be created by other programming paradigms, in theory the resulting machine code is a bitstring of data a control code just like in case of any other software. The only advantage, the *value* of the method is that this paradigm forces us to think and create a model in a certain way that we would most probably not do with other paradigms.

VII. **Agentbased Modelling has to remain an art:** The existence of a closed form method would mean that we are not modelling a complex system. The formation of a model, especially an ABM, requires significant human input that is necessarily subjective. Therefore the model is only complete *for* the modeler and *with* the modelers state of mind.

8.3 Conclusions

The modelling and simulation of complex systems is not without traps and pitfalls. First of all we know that chaotic systems are extremely hard and mostly impossible to predict, also in case of deterministic chaos. Aiming at the modelling of such aspects of large production networks is a questionable attempt, but necessary, at least for the identification of chaotic regimes and the gathering of knowledge by simulation. Complex systems are even more problematic than just chaotic ones, because of the great number of interacting parts where we are not satisfied with modelling and observing the global dynamics, but are interested in the internal representation of the components.

One interesting aspect of the work carried out in this thesis was that the research was drawn towards – without such intentions – the philosophy of science. It was necessary to face some very basic questions related to human knowledge, undecidability, incompleteness and unexplainability. Complex systems cannot be handled with the ease of more traditionally formulated ones, especially since formal systems have been proven to be incomplete and contradicting themselves.

Production systems of significant size we are talking about *are* complex adaptive systems – or more precisely they do have such an aspect – even if we don't model or consider them in that fashion. This is a key issue. However, even if the managerial search for best practices is an evolutionary search for better schemata, *formalized, self-aware methods are always superior* – as close to scientific standards as possible – and this is the direction this thesis is aiming to progress. Even if we don't model strategies, rules, or policies, firms are subject to extinction – and they do get extinct. Extending the area of interest to modelling these aspects complicates our task and makes us face many limitations, but avoiding them do not make us wiser either.

Already in the 1940's John von Neumann promoted that many systems such as enterprises should be modeled based on individuals. Due to the availability of cheap computing power his challenge can be pursued and the area of computational modelling and complex systems is receiving more and more attention. The biggest problem is that system theories fashionable in the XXth century have quickly reached their limitations and did not go much further from observations and pointing out some interesting aspects of systems. This is very well understandable if we consider the combinatorial explosion that the complication and detail of description can cause in an interconnected system. This is what most probably inspires the pursuers of systemic understanding to postpone the development of system theories (and its applications in domains subject to hype-and-disappointment) nowadays and concentrate on details, gathering lots of knowledge at the micro level. All this vast amount of generated data – when time comes – can provide an establishment for formulating a theory.

To sum up the concerns mentioned so far: it is dangerous to talk about systems and complexity, because there are no standard methodologies developed for them yet, and many who tried made only very modest claims [32], or had to step back and modify goals and research methods. In the same time it is somewhat paradox to even talk about a "standard methodology", because we have just made it a point that only an abstract methodology captures complexity at its best. This work is therefore relatively abstract and does not make claims at the quantitative level, and reaches back to basic knowledge about the philosophy of science and possible development concerning the integration of computers into the knowledge gathering process.

8.4 Future Plans and Opportunities

The wide range of scientific disciplines explored and mentioned in this work naturally implies that they couldn't have been and were not exploited fully. The level of detail isn't low either, therefore plans are already on the way to work it out by e.g. reviewing more potential metaphors, work out their transfer in detail, provide ample software to represent them computationally and most importantly refine a methodology for the investigation and application of such systems.

The ultimate vision associated with the proposed methodology is that it will eventually become a day-to-day tool, supporting modelling for the assessment of opportunities, false scenarios and potential changes integrated with the rest of the analytic tools, methods, software currently used in the business of production networks.

References

[1] F.T. **Arecchi**, *Complexity in Science: Models and Metaphores,* in The Emergence of Complexity in Mathematics, Physics, Chemistry and Biology, Pontifica Academia Scientiarvm, 27-31 October, 1992. Princeton University Press. 1992.

[2] Ross W. **Ashby**, *An Introduction to Cybernetics,* Chapman and Hall Ltd., London, 1956.

[3] Robert **Axelrod**, *The Complexity of Cooperation – Agent-Based Models of Competition and Collaboration,* Princeton University Press, 1997.

[4] Robert **Axelrod**, Michael D. Cohen, *Harnessing Complexity – Organizational Implications of a Scientific Frontier,* Basic Books, 2000.

[5] Robert **Axtell**, *The Emergence of Firms in a Population of Agents: Local Increasing Returns, Unstable Nash Equilibria, And Power Law Size Distributions,* Working paper No. 3., 1999.

[6] Stafford **Beer**, *Decision and Control – The meaning of Operational Research and Mangement Cybernetics,* Wiley, 1966.

[7] Suzanne **Berger**, Constanze Kurz, Timothy Sturgeon, Ulrich Voskamp, Volker Wittke: *Globalization, Production Networks and National Models of Capitalism – On the Possibilities of New Productive Systems and Institutional Diversity in an Enlarging Europe,* Soziologisches Forschungsinstitute Göttingen e.V., www.sofi-goettingen.de, 2003.

[8] T. **Bäck**, U. Hammel, H.P. Schwefel, *Evolutionary Computation: Comments on the History and State,* IEEE Transactions on Evolutionary Computation, Vol. 1, No.1, April 1997.

[9] Ludwig von **Bertalanffy**, *General System Theory, Foundations, Development, Applications,* George Braziller, New York, 1968.

[10]G.A. **Britton**, S.B. Tor and W.Y. Zhang, *Techniques in Knowledge–Based Expert Systems for the Design of Engineering Systems.* Book chapter in Business and Technology of the New Millenium, ed. C.T. Leondes. To be published 2004.

[11]Dipankar **Dasgupta,** *Special Issue on Artificial Immune Systems,* IEEE Transaction on Evolutionary Computation, Vol.6., No.3., June 2002.

[12]Gregory N. **Derry**, *What Science is and How it Works*, Princeton University Press, 1999.

[13]Gary W. **Flake**, *The Computational Beauty of Nature – Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation,* MIT Press, 1998.

[14]Douglas J. **Futuyma**, *Evolutionary Biology,* Sinauer Associates 1986.

[15]Ram **Ganeshan**, Terry P. Harrison. *An Introduction to Supply Chain Management.* Penn State University. URL:http://silmaril.smeal.psu.edu/misc/supply_chain_intro.html

[16]Murray **Gell-Mann**, *The Quark and the Jaguar – Adventures in the Simple and the Complex,* Abacus, Little, Brown and Co., UK, 1995.

[17]F.H. **George**, *Automation, Cybernetics and Society,* Leonard Hill Books Ltd., London, 1959.

[18]Robert **Gibbons**, *Why Organizations Are Such a Mess (and What an Economist Might Do About It),* draft paper, MIT Sloan School, 2000.

[19]Kurt **Gödel**, *Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme, I.* Monatshefte für Math. u. Physik 38, 173-198, 1931.

[20]Kurt **Gödel**, *On Formally Undecidable Propositions of Principia Mathematica and Related Systems.* New York: Dover, 1992.

[21]Jörg **Heitkötter** and David Beasley, eds. *"The Hitch–Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)",* USENET: comp.ai.genetic. http://www.cs.bham.ac.uk/Mirrors/ftp.de.uu.net/EC/clife/www/. 2000.

[22]A.S. **Hornby**, S. Wehmeier, *Oxford Advanced Learners Dictionary*, Oxford University Press, 1989.

[23]Ricki G. **Ingalls**, *The Value of Simulation in Modeling Supply Chains,* Proceedings of the 1998 Winter Simulation Conference, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

[24]Erich **Jantsch**, *The Self-Organizing Universe – Scientific and Human Implications of the Emerging Paradigm of Evolution,* Pergamon Press, 1989.

[25] Stuart **Kauffman**, *At Home in the Universe,* Oxford University Press. 1995.

[26] Janne **Keskinarkaus**, Béla Pátkai, Seppo Torvinen, *Generating and Solving Resource-Constrained Project Scheduling Problems Emulating Virtual Production Networks,* ICPR'03, The 17th

International Conference on Production Research, August 3-7, 2003, Blacksburg, Virginia, USA.

[27]Dilip **Kondepudi**, Ilya Prigogine, *Modern Thermodynamics – From Heat Engines to Dissipative Structures,* John Wiley & Sons Ltd, 1998.

[28]John R. **Koza**, *Genetic Programming II., Automatic Discovery of Reusable Programs (Complex Adaptive Systems),* Bradford Books, 1994.

[29]L.D. **Landau**, E.M. Lifshitz, *Mechanics, Volume 1 of Course of Theoretical Physics,* Pergamon Press, 1969.

[30]Christopher L. **Langton**, *Artificial Life: The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems,* September, 1987, Los Alamos (Proceedings Volume in the Santa Fe Institute Studies in the Sciences of comPlexity, V. 6.)

[31]D. **Levy**, *Chaos Theory and Strategy, Theory, Application and Managerial Implications,* Strategic Management Journal, 15 (Summer), 167–78. 1994.

[32]Francesco **Luna**, Benedikt Stefansson, *Economic Simulations Swarm: Agent-Based Modelling and Object Oriented Programming,* Kluwer Academic Publishers, 2000.

[33]K. **Miettinen**, M.M. Mäkelä, P.Neittaanmäki, J. Périaux, *Evolutionary Algorithms in Engineering and Computer Science, Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications,* Wiley, 1999.

[34]Sibylle D. **Müller,** Jarno Marchetto, Stephano Airaghi, Petros Koumoutsakos, *Optimization Based on Bacterial Chemotaxis,* IEEE Transactions on Evolutionary Computation, Vol.6., No.1., February 2002.

[35] John von **Neumann**, Oskar Morgenstern, *Theory of Games and Economic Behavior,* Wiley, 1967.

[36]Grégoire **Nicolis**, Ilya Prigogine, *Exploring Complexity – An Introduction,* R. Piper GmbH and Co. KG Verlag, Munich, 1989.

[37]T. **Ostrom**, *Computer Simulation: the third symbol system.* Journal of Experimental Social Psychology, vol.24., pp.381–392. 1998.

[38]John F. **Padgett**, Doowan Lee and Nick Collier, *Economic Production as Chemistry,* Industrial and Corporate Change, 2003.

[39]H.Van Dyke **Parunak**, Robert Savit, Rick L. Riolo, *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and User's Guide,* Proceedings of Multi-Agent Systems and Agent-Based Simulation, MABS'98, 10-25, Springer, LNAI 1534. 1998.

[40]Béla **Pátkai**, J. Keskinarkaus, S.Torvinen, *An Agentbased Computational Framework for Production Network Modelling,* International Conference on Computational Cybernetics, ICCC'2003, Siófok, Hungary, August 29-31, 2003.

[41]Ilya **Prigogine**, *The End of Certainty – Time, Chaos, and the New Laws of Nature,* The Free Press, 1997.

[42]Robert D. **Putnam**, *Bowling Alone: The collapse and revival of American community,* Simon and Schuster, 2000.

[43]Nicholas **Roegen (Georgescu-)**, *The Entropy Law and the Economic Process,* Harvard University Press, 1999.

[44]Timothy J. **Sturgeon**, *How Do We Define Value Chains and Production Networks?*, Bellagio Value Chains Workshop, Sep.25-Oct.1, 2000.

[45] Timothy J. **Sturgeon**, *Turn-key Production Networks: A New American Model of Industrial Organization?,* University of California at Berkeley. Berkeley Roundtable on the International Economy. Working paper 92A, August 1997.

[46] Timothy **Sturgeon**, *Turn-key Production Networks: The Organizational Delinking of Production from Innovation,* In: Ulrich Juergens (ed.), New Product Development and Production Networks. Global Industry Experience, Springer Verlag, Berlin, New York. 1999.

[47]József Kázmér **Tar**, *Novel Adaptive Control of Roughly Modelled Non-linear Systems Exemplified in Robotics and Chemical Reactions*, lecture delivered at the 2nd International Symposiuum on Mechatronics Bridging Over Some Fields of Science, Budapest, November 15, 2002.

[48] Pietro **Terna**, *How to use the Java Enterprise Simulator (jES) Program?,* Dipartimento di Scienze economiche e finanziarie G. Prato, Universita di Torino, Italia. Nov. 2003.

[49]M.B. **Wall**, *A Genetic Algorithm for Resource–Constrained Scheduling,* PhD Thesis, Massachusetts Institute of Technology 1996.

[50]Gerald M. **Weinberg**, *An Introduction to General Systems Thinking,* Wiley-Interscience, 1975.

[51]Eric W. **Weisstein** et al. *Church-Turing Thesis* From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Church-TuringThesis.html 2004.

[52]Norbert **Wiener**, *Cybernetics or Control and Communication in the Animal and the Machine,* MIT, 1971.

[53]M. **Wooldridge**, N.R. Jennings, D. Kinny, *The Gaia Methodology for Agent-Oriented Analysis and Design.*

[54]H. **Wörn**, D. Frey, J. Keitel, *Digital Factory Planning and running enterprises of the future,* IECON'2000, The IEEE International Conference on Industrial Electronics Control and Instrumentation, Nagoya, Japan. October 22-28. 2000.

Bibliography

[55]Emile **Aarts**, Jan Korst, *Simulated Annealing and Boltzmann Machines,* Wiley, 1989.

[56]Ibrahim **Al-Harkan**,*On merging sequencing and scheduling theory with genetic algorithms to solve stochastic job-shops,* PhD thesis, University of Oklahoma, 1997.

[57]**APICS–1998**, Finite Capacity Scheduling Software Survey

http://lionhrtpub.com/apics/surveys/fcs/fcs97.shtml

[58]A. **Bauer**, R. Bowden, J. Browne, J. Duggan and G. Lyons, *Shop Floor Control Systems,* Chapman and Hall 1994.

[59]Christian **Bierwirth**, Dirk C. Mattfeld, *Production Scheduling and Rescheduling with Genetic Algorithms,* MIT Press, Evolutionary Computation, Volume 7, Number 1, 1999.

[60]Agostino **Bruzzone** and Robert Signorile, *Simulation and Genetic Algorithms for Planning and Shipyard Layout,* Simulation, Special Issue on Harbour and Maritime Simulation, Volume 71, Number 2, August 1998.

[61]S.J. **Buckley** and S.S.Murthy, IEEE Expert, pp. 23., January–February, 1997.

[62]R. E. **Burkard** and E. Çela, *An Asymptotical Study of Combinatorial Optimization Problems by Means of Statistical Mechanics*. SFB Report No. 133, Institute of Mathematics B, Technical University Graz, July 1998.

[63]H. **Crowder**, M. Padberg, *Solving large-scale symmetric salesman problems to optimality*, Management Science, vol. 26, pp. 495–509, 1980.

[64]P. **Cunningham**, J. Browne, *A LISP–based heuristic scheduler for automatic insertion in electronics assembly,* Int. Production Research, 24 (6), 1398–1408. 1986.

[65]G. **Dagtzig**, D. Fulkerson, S. Johnson, *Solution of a large scale travelling salesman problems*, Operations Research, vol. 2., pp. 393–410. 1954.

[66]Charles **Darwin**, *The Origins of Species,* 1859.

[67]P. **Esquirol**, P. Lopez, *Constraint–Oriented Cooperative Scheduling for Aircraft Manufacturing,* IEEE Expert, pp. 32-39., January–February, 1997.

[68]W.J. **Fabricky**, P.M. Ghare, P.E. Torgersen, *Industrial Operations Research,* Prentice–Hall 1972.

[69]**FlexGA,** *Evolutionary and Genetic Algorithms User's Guide.* 1999.

[70]D. **Fyffe**, W. Hines, N. Lee, *System reliability allocation and a computational algorithm*, IEEE Transactions on Reliability, vol. R–17, pp.64–69, 1968.

[71]Mitsuo **Gen**, Runwei Cheng, *Genetic Algorithms and Engineering Design*, Wiley–Interscience Publication, 1997.

[72]J. **Giarratano**, G. Riley, *Expert Systems–Principles and Programming,* PWS–KENT Publishing, 1989.

[73]R. **Graham**, P. Hell, *On the history of the minimum spanning tree problem*, Annals of the History of Computing, vol. 7., pp. 43–57, 1985.

[74]M. **Grötschel**, O. Holland, *On the symmetric traveling salesman problem: solution of a 120-city problem*, Mathematical Programming Studies, vol. 12, pp. 61–77, 1980.

[75] M. **Grötschel**, O. Holland, *Solution of large–scale symmetric traveling salesman problems*, Mathematical Programming, vol. 51, pp. 141–202, 1991.

[76]G. **Halevi**, *The Role of Computers in Manufacturing Processes,* Wiley and Sons, 1980.

[77]E. **Hart**, P. Ross, J. Nelson, *Producing robust schedules via an artificial immune system*. In Proceedings of the IEEE World Congress on Computational Intelligence, 1997.

[78]Emma **Hart**, David Corne, *The State of the Art in Evolutionary Approaches to Timetabling and Scheduling,* EvoStim: The EVONET Working Group on Evolutionary Scheduling and Timetabling, 1999.

[79]D. **Hildum**, *Flexibility in a Knowledge–based System for Solving Dynamic Resource–Constrained Scheduling Problems,* Umass CMPSCI Technical Report 94-77, University of Massachusetts, Amherst, 1994.

[80]Paul **Hoffman**, *The Man Who Loved Only Numbers, The Story of Paul Erdős and the Search for Mathematical Truth*, Little Brown & Company, 1999.

[81]J.H. **Holland**, *Outline for a logical theory of adaptive systems,* J. Assoc. Comput. Mach., Vol. 3, pp. 297–314, 1962.

[82]S. **Johnson**, *Optimal two–and–three stage production schedules with setup times included*, Naval Research Logistics Quarterly, vol. 1, pp. 61–68, 1954.

[83]S. **Kirkpatrick**, C.D. Gelatt Jr., M.P. Vecchi, *Optimization by simulated annealing*, IBM Research Report RC 9355. 1982.

[84]J. **Kruskal** Jr., *On the shortest spanning subtree of graph and the traveling salesman problem*, Proc. ACM, vol. 7, no. 1., pp. 48–50, 1956.

[85]Mika **Kuhmonen**, *The Effect of Operational Disturbances on Reliability and Operation Time Distribution of NC–machine Tools in FMS,* Lappeenranta University of Technology, Finland. Doctor of Technology Thesis, 1997.

[86]Cornelius **Lanczos**, Space Through the Ages, 1974.

[87]Hau L. **Lee**, Corey Billington, *The Evolution of Supply-Chain-Management Models and Practice at Hewlett-Packard.* Interfaces 25 (pp.42-63): 5 September-October, 1995.

[88]B.L. **Maccarthy**, Jiyin Liu, *Addressing the gap in scheduling research: a review of optimization and heruistic methods in production scheduling,* International Journal of Production Research, Vol. 31., No.1., 1993.

[89]Ernst **Mach**, *Popular Scientific Lectures,* Chicago, 1895.

[90]Abbe **Mowshowitz**, *Virtual Organization: The New Feudalism*, COMPUTER, pp. 110–112. April 2001.

[91]M. **Numao**, S. Morishita, *Cooperative Scheduling and its Application to Steelmaking Processes,* IEEE Trans. Industrial Electronics, vol. 38, no. 2, pp. 150-155. 1991.

[92]M. **Padberg**, G. Rinaldi, *Optimization of a 532-city symmetric traveling salesman problem by branch and cut,* Operations Research Letters, vol. 6, pp. 1–7, 1987.

[93]M. **Padberg**, G. Rinaldi, *A branch and cut algorithm for the resolution of large scalesymmetric traveling salesman problems,* SIAM Review, vol. 33, pp. 60–100, 1991.

[94]S.S. **Panwalker**, W. Iskander, *A survey of scheduling rules,* Operations Research, 1977.

[95]Judea **Pearl**, *Heuristics – Intelligent Search Strategies for Computer Problem Solving,* Addison–Wesley 1984.

[96]Jens **Rasmussen**, *Information Processing and Human–Machine Interaction,* North–Holland, 1986.

[97]Herbert **Simon**, *Administrative behavior, 4th Edition,* Free Press, April 1997.

[98]Arno **Sprecher**, *Scheduling Resource–Constrained Projects Competitively at Modest Memory Requirements,* Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1997.

[99]Janyashankar M. **Swaminathan**, Stephen F. Smith, Norman M. Sadeh, *A Multi Agent Framework for Modeling Supply Chain Dynamics.* Technical Report, The Robotics Institute, Carnegie Mellon University, 1996.

[100] Lester C. **Thurow**, *Building Wealth – the New Rules for Individuals,Companies in a Knowledge–Based Economy,* MIT press, 1999.

[101] F. **Tillman**, C. Hwang, W. Kuo, *Optimization of Systems Reliability,* Marcel Dekker, New York, 1988.

[102] Seppo **Torvinen**, *Model Factory Concept, Company Networking and Subcontracting Collaboration.* Seminar organised by the Finnish Federation of Metal, Engineering and Electrotechnical Industries, Hanasaari Culture Centre, Espoo, Finland, 22.4.1998.

[103] Gordon **Walker**, Bruce Kogut, Weijian Shan, *Social Capital, Structural Holes and the Formation of an Industry Network,* Organizational Science, (JSTOR) Vol.8.Issue 2, 1997.

[104] Gerhard **Weiss**, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* The MIT Press, 2000.

[105] D.H. **Wolpert**, W.G. Macready, *No Free Luch Theorems for Optimization,* IEEE Transactions on Evolutionary Computation, Vol. 1, No.1, April 1997.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland