

Igor Nkolić

# Co-Evolutionary Method For Modelling Large Scale Socio-Technical Systems Evolution

26



NEXT  
GENERATION  
INFRASTRUCTURES  
FOUNDATION

Co-Evolutionary Method For Modelling Large Scale Socio-Technical Systems  
Evolution

PROEFSCHRIFT

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen op 18 november 2009 om 10.00 uur  
door

Igor NIKOLIĆ

scheikundig ingenieur  
geboren te Novi Sad, Joegoslavië.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M.P.C. Weijnen

Copromotor: dr. ir. G.P.J. Dijkema

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. M.P.C. Weijnen	Technische Universiteit Delft, promotor
Dr. ir. G.P.J. Dijkema	Technische Universiteit Delft, copromotor
Prof. dr. ir. A. Verbraeck	Technische Universiteit Delft
Prof. dr. ir. P. A. Kroes	Technische Universiteit Delft
Prof. dr. C.M. Jonker	Technische Universiteit Delft
Prof. dr. B. R. Allenby	Arizona State University
Prof. dr. S. E. Page	University of Michigan

ISBN 978-90-79787-07-4

*Published and distributed by:*

Next Generation Infrastructures Foundation

P.O. Box 5015

2600 GA Delft

The Netherlands

T: +31 15 278 2564

F: +31 15 278 2563

E: info@nginfra.nl

I: www.nginfra.nl

This research was funded by the Next Generation Infrastructures Foundation programme and the TU Delft.

Copyright © 2009 by I. Nikolic

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Author's e-mail: nikolic.igor@gmail.com. Typeset by the author with L<sup>A</sup>T<sub>E</sub>X. Title page illustration by The Perry Bible Fellowship, <http://pbfbcomics.com/>, used with permission.

<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Socio-technical System Earth . . . . .	1
1.2 Global challenge . . . . .	3
1.3 Discovering Patterns . . . . .	7
<b>I Theory: Towards a co-evolutionary modeling method</b>	<b>9</b>
<b>2 Research Theme and Question</b>	<b>11</b>
2.1 Perspective . . . . .	11
2.2 Thesis Focus . . . . .	18
2.3 Research Question . . . . .	20
2.4 Readers' Guide . . . . .	21
<b>3 Theoretical Foundations</b>	<b>25</b>
3.1 Introduction . . . . .	25
3.2 Formalisms . . . . .	27
3.3 Complex Adaptive Systems . . . . .	28
3.4 Evolution . . . . .	34
3.5 Complexity Framework . . . . .	41
<b>4 Modelling Foundations</b>	<b>45</b>
4.1 Introduction . . . . .	45
4.2 Industrial Clusters . . . . .	46
4.3 Modelling Complex Adaptive Systems . . . . .	48
4.4 Literature Review . . . . .	54
4.5 Operationalizing the Agent . . . . .	56
4.6 Next Steps . . . . .	59

<b>5</b>	<b>Modelling method and requirements</b>	<b>61</b>
5.1	Introduction . . . . .	61
5.2	Core Concepts . . . . .	62
5.3	Guiding Principles . . . . .	69
5.4	Requirements . . . . .	72
5.5	Method description . . . . .	76
<b>II</b>	<b>Practice: Co-evolutionary method in action</b>	<b>83</b>
<b>6</b>	<b>Learning: Case Studies and Knowledge Engineering</b>	<b>85</b>
6.1	Introduction . . . . .	85
6.2	Flow-based Evolution . . . . .	87
6.3	Combination of Infrastructures . . . . .	99
6.4	Chocolate Game Model . . . . .	111
6.5	System Decomposition Method . . . . .	126
6.6	Conclusion . . . . .	141
<b>7</b>	<b>Full-scale Case Study: CostaDue</b>	<b>143</b>
7.1	Focus Point . . . . .	144
7.2	Hypothesis . . . . .	144
7.3	Case Description . . . . .	145
7.4	Model Details . . . . .	146
7.5	Experiment Details . . . . .	153
7.6	Case Study Results . . . . .	155
7.7	Domain-Specific Insights . . . . .	165
7.8	Method Development Conclusions . . . . .	166
7.9	Conclusion . . . . .	170
<b>8</b>	<b>Method verification: evolution of three case studies</b>	<b>173</b>
8.1	Introduction . . . . .	173
8.2	Bulk Biochemicals . . . . .	174
8.3	Metals Production Network . . . . .	186
8.4	Bioelectricity . . . . .	200
<b>III</b>	<b>Insights</b>	<b>215</b>
<b>9</b>	<b>Results and Discussion</b>	<b>217</b>
9.1	Results and Discussion . . . . .	217
<b>10</b>	<b>Conclusions</b>	<b>239</b>
<b>11</b>	<b>Reflections and Future Work</b>	<b>243</b>
11.1	Reflections . . . . .	243
11.2	Outlook and Future Work . . . . .	249
	<b>Bibliography</b>	<b>252</b>

<b>IV</b>	<b>Appendices</b>	<b>285</b>
<b>A</b>	<b>System Levels Detailed</b>	<b>287</b>
A.1	Agent . . . . .	287
A.2	Network . . . . .	290
A.3	System . . . . .	294
<b>B</b>	<b>CVI Ontology</b>	<b>299</b>
<b>C</b>	<b>CVI Infrastructures</b>	<b>301</b>
<b>D</b>	<b>Simulation engine</b>	<b>303</b>
D.1	Hardware and Operating System . . . . .	303
D.2	Knowledge Management . . . . .	305
D.3	Simulation Software . . . . .	308
<b>E</b>	<b>LHS vs Random sampling</b>	<b>311</b>
<b>F</b>	<b>Requirements Table</b>	<b>313</b>
<b>G</b>	<b>Overview of the Reviewed Literature</b>	<b>317</b>
	<b>Summary</b>	<b>329</b>
	<b>Samenvatting</b>	<b>335</b>
	<b>Curriculum Vitae</b>	<b>343</b>



## ACKNOWLEDGEMENTS

First and foremost I must acknowledge the support of my co-promotor and daily supervisor, Dr. Ir. G.P.J. Dijkema. His ability to let a PhD candidate do his thing, encouraging him to find his own way and yet be able to step in and steer when necessary is legendary. Without his efforts, this thesis would have never reached the level it has. I am also very grateful to my promotor, Prof. dr. ir. M.P.C. Weijnen who, despite having an agenda from hell, always managed to patiently and carefully read, and with uncannily sharp wit, comment on my work. The sometimes heated debate that followed has each and every time lead to further improvement of this thesis. She managed to channel my stubborn and headstrong writing style into a balanced thesis.

Science is not something one can do alone. Without the intense collaboration with my colleagues and students, this work would be much poorer. For collaboration that directly made it into the thesis I would like to thank: P.J. Beers, G. Beekebreede, S. Blokker, K.H. van Dam, C.B. Davis, P.W. Heijnen, R. van Krevelen, T. E van de Lei and K. Sakamornsnguan.

Science should also be fun. Chris, Koen, Emile, Rob, Laurens, Angelique, Connie and all the other colleagues at the E&I group, thanks. You all made it endlessly more fun. After all, where would we be without regular sessions of magnet penguin bondage?

Finally, I have to acknowledge the support of my family. Without complaining they suffered a stressed, busy and sometimes grumpy dad and husband. Without their love and support, this thesis would never exist. Pieter, Rutger and Carola, this is for you...

Igor Nikolić  
Delft, October 2009



# CHAPTER 1

## INTRODUCTION

There is a Chinese curse which says: “May he live in interesting times.” Like it or not, we live in interesting times...

*Robert F. Kennedy, 1966*

Our planet is facing the greatest problems it has ever faced. Ever. So whatever you do, don't be bored. This is absolutely the most exciting time we could possibly hope to be alive in. Things are just starting...

*The Kleptones, Careless or Dead, 24h, 2006*

## 1.1 Socio-technical System Earth

**Prehistory to 19<sup>th</sup> century** Since man is around <sup>1</sup>, production and trade have existed. As early as 3000 years ago the Silk Road started connecting Asia, Europe and Africa, over land and water, transporting goods, technology and ideas between the continents. Trade continued and intensified as the cycle of the rise and ruin of civilizations continued. With the development of sailing ships, all continents were connected, large-scale trade began, and the first multinational companies emerged. The invention of the steam engine in the 18<sup>th</sup> century sparked the Industrial Revolution. On the upside, this enabled a tremendous increase in the speed of production and transport. On the downside, masses of workers were laid off and fell into extreme poverty, and for the remainder of the employed the working conditions in the factories were appalling. Industrialization included the building of railroad networks that enabled further national industrial development, and international steam shipping led to a further increase in international trade and colonialism. The poor working conditions incited the founding of labor unions, which then increasingly affected politics and the way businesses were organized. At the same time, an extensive telegraph network developed, enabling reliable global communication. Some of the wealth generated started to trickle down to ordinary citizens, driving consumption.

---

<sup>1</sup>This introduction is not meant to be a complete historic account of the human history. It is a personal view used to set the stage.

**Rise of engineering** In the 2<sup>nd</sup> half of the 19<sup>th</sup> century industrial skills and practices rapidly increased. The formation of engineering societies fostered the translation of scientific underpinning and codification of industrial operations into formal R&D procedures (Dijkema, 2004). By World War I, major engineering disciplines such as electrical engineering, chemical engineering, mechanical engineering and civil engineering had become well-established.

**Twentieth century** During the interbellum and in the years following WW II, the world economy became increasingly networked. Decisions made in one part of the world began to have direct impact across the globe. The achievement of large-scale electric power generation, and especially the introduction of the incandescent bulb in the late 19<sup>th</sup> and early 20<sup>th</sup> centuries, sparked the development of a new physically networked infrastructure that quickly spanned entire continents. An explosive increase in telecommunications infrastructure, first through telegraph, later the telephone and the Internet allowed the financial networks to emerge. These were able to fund these large-scale technical systems already linked the greater part of the globe, further increasing the speed of their development. They affected the world not only economically and politically, but also in terms of real physical and social impact. The 1929 stock market crash and the subsequent global recession is a case in point.

**Post WW II** WW II not only changed the international scene; it also spurred technological development, laying the foundations of post-war industrial development, e.g. in petrochemicals and polymers. The discovery of the transistor and, later, the computer chip started the information revolution. The discovery of DNA led to advancements in biotechnology and medicine. In the military, notably in aircraft and aerospace engineering, a new engineering discipline was in the making: systems engineering (Blanchard and Fabrycky, 1998; Dijkema, 2004). Public multinational organizations such as the UN, the WTO and the IMF entered the global stage, while large multinational companies with their fast and massive capital flows dominated the global economy. During this period the global distribution of income became increasingly skewed (Melchior et al., 2000), creating a massive disparity between rich and poor. Despite these excesses, the general level of welfare increased overall, especially in the developing world.

**Twenty-first century** The last decade of the 20<sup>th</sup> century saw the emergence of the Internet and the World Wide Web, inspiring a new revolutionary era in network-based information systems (Castells, 2000). The micro-electronics industry supporting them mushroomed, and the resulting dramatic price drop and performance increase allowed for further developments. Molecular biology and the Human Genome project sparked a biological information revolution in the life sciences. Global supply networks, the extensive offshoring of manufacturing and services and the global financial networks enabled by the Internet have transformed the planet into a globally connected village (Friedman, 2005). The size of the World Wide Web in 2008 was estimated to be between 27 and 60 x 10<sup>9</sup> pages<sup>2</sup>, and the number of users is currently growing at 300% per year<sup>3</sup>. Global society has become a highly interconnected networked system.

---

<sup>2</sup><http://www.worldwidewebsite.com/> Accessed August 19, 2008

<sup>3</sup><http://www.internetworldstats.com/stats.htm> Accessed August 19, 2008

**Box 1: Vocabulary** This thesis draws from many different scientific fields. This has resulted in a multidisciplinary vocabulary being used in this thesis. Therefore, not all concepts may therefore be familiar to the reader. Key concepts are briefly defined below and are further elaborated in Chapters 3 and 4.

**system** A regularly interacting or interdependent group of components forming a unified whole.

**intractability** A process is intractable when the fastest way to determine its outcome is to perform the process, i.e., the outcome can not be determined in advance by any computational device.

**evolution** The process of variation, reproduction and selection of individuals over time. When applied to non-biological systems it describes the process of change in the structure and the function of a system.

**co-evolution** As entities never evolves alone, but act and react with other entities, evolution is often referred to as co-evolution.

**attractor** A set of states to which a system evolves. That is, once a system state evolves to the point that it is relatively stable (the attractor point), it will remain close to it even if the system is disturbed.

**complex adaptive system** A dynamic network of many components acting in parallel, constantly acting and reacting to what the other components are doing.

**emergence** Stable macroscopic patterns arising from the local interactions of system components.

**actor** Any entity (person or organization) in the real world.

**agent** The abstraction of an actor, being the smallest component in a complex adaptive system.

**formalism** A well-defined language describing a certain knowledge domain, such as mathematics, economics or psychology.

## 1.2 Global challenge

**World problems** The societal development described in the previous section has come at a high environmental price. As The Kleptones point out, our planet is facing the greatest problems it has ever faced. Population is growing, as is material and energy consumption, resulting in an ever-increasing environmental impact. The IPAT equation (Impact-Population-Affluence-Technology) (Allenby, 1999; Graedel and Allenby, 1995) (see Eq. 1.1) offers insight into the situation.

$$Environmental\ Impact = Population \times \frac{GDP}{Person} \times \frac{Environmental\ Impact}{Unit\ of\ per\ capita\ GDP} \quad (1.1)$$

By examining the components of this equation we can gain a sense of its scale of impact.

*Environmental Impact* Current atmospheric CO<sub>2</sub> concentrations are the highest in 400,000 years (Petit et al., 1999). Global warming is evident (KNMI, 2008; Parry et al., 2007) and is already adversely influencing the human population. Deforestation and desertification are destroying ecosystems across the globe (Skole and Tucker, 1993). This habitat loss is leading to a record number of species going extinct (Barbault and Sas-trapradja, 1995; Pimm and Raven, 2000).

*Population* In the last 50 years the earth's population has more than doubled (UN, 2006).

$\frac{GDP}{Person}$  World industrial production has increased 100 fold over the past 100 years (Allenby, 1999), the global GDP has increased by a factor of 50 (DeLong, 2002), and global energy consumption has increased by a factor of 100.

$\frac{Environmental\ Impact}{Unit\ of\ per\ capita\ GDP}$  Economies are requiring more and more energy and materials per unit of production (Bartelmus, 2003). For example, in China the direct material input per unit of GDP doubled between 1989 and 1996. "Nowhere ... do we find a case of 'absolute delinking' in the sense of absolute reductions of material input occurring while the economy continues to grow" (Fischer-Kowalski and Amann, 2001), and thus nowhere is a reduction in the environmental impact per unit of GDP observable.

**Wicked problems** While these aforementioned problems seem unrelated, their causes and effects are deeply intertwined. They are sometimes described as *wicked* problems (Rittel and Webber, 1984, 1973). Solutions to wicked problems are often difficult to recognize as such because of the maze of interdependencies. They have contradictory and changing requirements; and while attempting to solve them, the solution may reveal or create other, even more complex problems. More and more of these wicked problems are emerging at the same time and on the same planet. While it is possible to view them in isolation from each other, such a reductionist approach has proven to be inadequate in solving these issues.

**System collapse** The world is indeed facing its greatest problems to date. Starting with Rachel Carson's "Silent Spring" (Carson, 1962) and the "Limits to Growth" report (Meadows et al., 1972), a variety of 'system thinkers' have postulated societal collapse (Diamond, 2005). This collapse would be caused by environmental problems. The human presence does not just affect the earth's ecosystems; it dominates them (van der Voet, 2001). We are exceeding the planet's carrying capacity. At the same time, the human species is fully dependent on the world's ecosystems for its survival. Spaceship Earth (Fuller, 1969) is a large, networked system of which each and every component has some effect on all other components, no matter how small. In order to gain insight in to the interrelatedness of the world's problems, and perhaps attempt to solve them, it is necessary to have a broad, systems-based perspective.

**Box 2: Systems perspective** Systems perspective is aptly defined by Bar-Yam (Bar-Yam, 2003, 2008) :

Taking into account all of the behaviors of a system as a whole in the context of its environment is the systems perspective. While the concept of system itself is a general notion that indicates separation of part of the universe from the rest, the idea of a systems perspective is to use a non-reductionist approach in the task of describing the properties of the system itself.

In the systems perspective, once one has identified the system as a separate part of the universe, one is not allowed to progressively decompose the system into isolated parts. Instead, one is obligated to describe the system as a whole. If in describing system properties one separates them into parts, this produces an incomplete description of the behavior of the whole. Any description of the whole must include an explanation of the relationships between these parts and any additional information needed to describe the behavior of the entire system.

**Multiple levels and perspectives** Given the scale and complexity of the whole earth system, it is impossible to understand it in its totality. In order to systematically study it, we need to deconstruct it into smaller, interrelated components. These components can be aggregated into three system levels: micro, meso and macro, which generally correspond respectively to the local, regional and global scales. The interactions of many local components create regional components, and many regional components together create the global systems state.

The identity of any component consists of two parts: the first part being the nature of the component *itself*, and the second part being the interactions it has with other components. Because of this dual nature, the description of any component is dependent on the *perspective* from which it is observed. In other words, whether a given interaction is taken into consideration or not determines the way the component is viewed by a particular observer.

Each component in the system can and must be viewed from a variety of perspectives, as no component can be fully defined and understood from a single perspective (Mikulecky, 2001; Nietzsche et al., 1961). Obviously, a system comprising of these components will also require a multiple-perspective approach. In the context of understanding socio-technical components, these perspectives can be understood as scientific disciplines. Each discipline, be it engineering, economics or psychology, views system components differently. For example, from the engineering perspective, the chair you are sitting on is a stable structure, carefully designed using geometric principles to hold certain forces in the vertical direction. From an economic perspective, it is a product of the global economic system, its price resulting from the market equilibrium of supply and demand for office furniture. From a psychological perspective, the chair is an extension of its user. Whether you are sitting on a cheap Ikea chair or in a luxurious leather armchair projects a certain image of you to others.

**Avoiding system collapse** In order to avoid a system-wide collapse, we need to change the system's components, whose identity and interactions give rise to the aforementioned problems. Whenever a change is made to the way a system component is and interacts, the change immediately affects the entire system. The moment the change has taken place, the system is effectively changed. The observable effects of this change, however, may not be apparent right

away. Observable, system-level changes in a systems state may happen over very different time scales than the actions that cause the change to happen. Sometimes these changes are immediate: for example, banning cars from driving through Beijing during the 2008 Olympics directly improved the air quality in the city. However, because the global system operates on biogeochemical time scales of millennia and distances of thousands of kilometers (Lovelock, 2000), while the human perception horizon is measured in a few kilometers and six months (Allenby, 1999), the effects are either too large or happen too slowly for humans to directly perceive. Furthermore, every change in a system component causes many other system components to react, further changing the system. The system thus evolves over time, through visible changes to its state and through changes in its internal interaction structure.

**Problem of prediction and design** The challenge is thus not only to change the system components, but also to be able to anticipate and evaluate the long-term effects of our actions, since they can easily turn out to be undesirable. This means that we need to learn how to more accurately predict the possible future outcomes of evolving systems, both when the systems are managed and when they evolve autonomously.

The following situation is an example of such an effect. Cooling food is an important measure in prolonging its usability and allowing it to be stored for lean times. The first mechanical cooling systems, invented in 1756 and developed in the following decades, used either very flammable (diethyl ether) or very toxic (ammonia) substances in their operation. In the 1920s a safe, non-toxic and non-flammable solution was finally found in the form of chlorofluorocarbon compounds (CFCs). These compounds were used on a massive scale, and based on their toxicological and physical safety, and because of the structure of the production system, no recycling was performed. Only in the 1980s was it realized that CFC emissions were responsible for the rapid depletion of the earth's ozone layer. It seemed a simple case of single cause, single effect; and CFCs were relatively quickly banned by the Montreal protocol (Ozone Secretariat of the United Nations Environment Programme, 2006) in 1987. The treaty was hailed as a victory of global environmental policy. However, HFC-23, the main replacement for CFCs, was found to have a global warming potential (GWP) 12,000 times greater than that of CO<sub>2</sub>. Since global climate responds even much more slowly to emissions than the ozone layer does, the full effect of HFC emissions is today barely observable but will slowly reveal itself as the earth's atmosphere transports them to the stratosphere in the coming hundred years (Sand et al., 1997). As previously mentioned, such problems are wicked. There is no central coordinator that can 'solve' or 'optimize' the system. Even worse, there is no single solution, only the possibility to change one or more of the system components, creating new problems which we can not predict in advance.

**Addressing global problems** Given the interrelatedness of global problems and the overwhelming complexity of the systems perspective with its multiple components, how does one go about solving these problems? A new approach is needed.

The first thing we need to realize is that there is no 'quick fix'. There is, by definition, no single-perspective solution possible to a multi-perspective problem. We must be prepared to constantly shift perspectives and choose system levels based on what is appropriate at the time, sometimes even considering multiple perspectives and levels in concert. We must make decisions and perform actions that we know will affect the entire system, across all components, levels and perspectives.

**Self vs. interaction** The approach to solving these problems is two-tiered. First, we must change the identity of the components from which the system is built, either through changing the components *themselves* or by changing the *interactions* between them. Second, since in changing the system's components we change the entire system, we must understand how the system will react to a change in any one of its components. We must therefore learn how the entire system changes as well, and then shape its evolution towards a more sustainable state.

**Challenge summarized** The world of today is a fabric of interconnected, co-evolving, networked socio-technical systems that have large environmental impacts. In explicitly recognizing this interconnectedness, we realize that global problems must be understood from an integrated perspective: a whole system view across multiple levels (local, regional and global) and across multiple perspectives (multidisciplinary and multicultural). The solution to these global problems must be sought in interventions in multiple networks (socio-political, economic, knowledge, energy, mass etc.). In view of the evident public interests involved, the central problem this thesis attempts to answer can be defined as: *How can we shape the evolution of large-scale socio-technical systems towards a more sustainable state?*

## 1.3 Discovering Patterns

**The pattern of evolution** Some 2500 years ago Heraclitus (Fowler, 1977) said “Πάντα Ρεῖ”. Everything flows; the only constant is change. This still holds true today, and yet the speed of change has dramatically increased (Garreau, 2006). The constant change, or more specifically, constant evolution of systems, is the subject of this thesis.

**Knowing the future** It has been suggested that the universe can be viewed as a giant computational device (Bostrom, 2003; Fontana, 2005; Hayes, 1999), in which the computation is performed by the local interactions between system components (Wolfram and Gad-el Hak, 2003). The algorithm running in the universe, with the human species being one of its many components, is intractable (Dennet, 1996). Any attempt to predict the 'result', or future state of the universe, in advance would require the presence of something larger than the universe itself, containing a perfect description of the universe within it. For the creatures running around inside the universe who are themselves part of the computation (Adams, 2002), it is clearly impossible to predict the future state of the universe around them.

**The problem to be addressed** The problem this thesis addresses is that, while the exact prediction of the future state of the earth system is impossible, we need to have insights into the future development of the earth system as a result of our actions in order to sustainably develop further. The reasons for this inability to predict are that there are a myriad of different (types of) things interacting at the same time, and that we ourselves are part of the equation. Much of this interaction is path-dependent, and we have no perfect record of how these interactions have taken place in the past. Furthermore, humans have a limited time and space horizon when gathering information and are limited in their ability to process the available information. Finally, the environment is full of random and unpredictable events that further increase the difficulty of prediction.

**Recognizing patterns** If we cannot get an exact prediction of the future, the second best we can do is to try to grasp a sense of general direction. However, even this is difficult, since analyzing as many of the different interactions as possible and logically working out their consequences is something we humans do relatively poorly. In order to solve that deficiency, computers have been designed to do just that: systematically process lots of information. Humans, on the other hand, are very good in recognizing patterns, something computers have great difficulty with. Recognizing things that look like things we have seen before and basing our actions on them is something we do in our daily lives. The challenge is thus to use the combination of the systematic processing ability of machines and the creative pattern-analyzing ability of humans to gain insight into possible future behavioral patterns of systems and peer forward into the mist.

**The solution is in the patterns** This thesis aims to develop a testable and repeatable method for combining different types of knowledge from many different *people or domains* to create *computer simulations* of how the world changes. The models created by this method allow us to examine the way the simulated systems change over time under different conditions and reveal patterns to help us predict possible system futures. These patterns can then be used to make better informed decisions about which actions we should take so as to positively shape the future of our world. The method is based on using experts to collect and formalize knowledge, thereby allowing computers to systematically create patterns from this knowledge, and letting decision makers learn from these patterns and thus start understanding the systems of which we are a part.

## Part I

# Theory: Towards a co-evolutionary modeling method



## 2.1 Perspective

This thesis opened by a brief illustration of the development of socio-technical systems. The deep interconnectedness of social and technical systems was addressed. The global 'whole earth system' can be seen as an evolving entity wherein the social and technical components are co-evolving at an ever-increasing rate. As a consequence, the earth's human population is facing a global sustainability challenge, which requires timely and effective actions in order to address the interconnected, multiperspective and multilevel systemic problems. Using a systems perspective, the ability (or lack thereof) to quantitatively predict system-wide evolution subject to such actions was explored. It was concluded that instead of focusing on the exact prediction of a single evolutionary path, multiple potential system-wide evolutionary pathways must be generated when analyzing emerging patterns in response to certain actions.

This chapter introduces the concept of large-scale socio-technical systems, central to this thesis. Within this concept the perspective of sustainability as an emergent property of the global evolving system is discussed. Finally, the core theme of this thesis, 'What models to use to generate evolutionary patterns and how to develop them,' is elaborated into research questions.

### 2.1.1 Large-Scale Socio-Technical Systems

**$\lambda$ -systems** Large-Scale Socio-Technical Systems is a term used in Thomas Hughes's system theory (Bijker et al., 1987). To avoid having to spell out the lengthy term, these systems will be referred to as  $\lambda$ -systems throughout this work.  $\lambda$ -systems are a class of systems that span technical artifacts embedded in a social network, by which a large-scale, complex socio-technical artifact emerges. Examples of  $\lambda$ -systems include organizations and institutions that develop around and sustain a particular industrial system, be it a single plant, an industrial complex, a set of interconnected supply chains or an entire global enterprise. They consist of a large number of diverse technical artifacts, such as machines, factories, pipelines and wires. They also consist of social components, such as policies, organizations and institutions that shape the technical components and at the same time are shaped by them. Regional industrial clusters, interconnected power grids, multimodal transport networks and telecommunication

networks are examples of such systems.  $\lambda$ -systems thus consist of interwoven physical and social networks.

**Socio-technical paradigm** In the classic engineering view, individual technologies are placed in a physical system context, isolated from issues such as economics, policy or regulation. However, in a social context, activities such as R&D, investment and consumption only take place as a result of *decisions* made by individuals or organizations. Individuals either affect the technology directly as consumers or indirectly when organized in groups. Groups such as companies or governments generally reflect the desires and behavior of individuals, but usually have much more constrained behavior patterns (Luhmann, 1995). These groups create institutions, rules, regulations, policies and habits. Technology is developed under, and is subject to, these rules and regulations. Its use is driven by desires, emotions and the availability of knowledge (Williamson, 1987). Given this, the idea of socio-technical systems emerges as a wider systems perspective (Herder et al., 2008).

Our industrial society can be viewed as a collection of interconnected large-scale socio-technical systems. Connections between these systems are of multidimensional nature; system content, structure and boundaries shift and evolve. At the global level there is no central *coordinator*, but order and structure emerge from widely distributed bottom-up interactions of subsystems, some of which do have local centralized control, and some of which are fully distributed.

**Physical networks** Physical networks consist of interconnected physical entities or things. Connections and interactions between things are governed by laws of nature, and their design tends to be relatively fixed in time, as they often involve large material and financial investments. For example, chemical processing plants consist of unit operations (mixers, reactors, heat exchangers, etc.) that exchange mass and energy (Coulson and Richardson, 1999). Processing plants are in turn connected to global supply networks that produce a bewildering range of goods. Other examples of physical networks include the network industries of electric power, natural gas, water, etc., as well as road and rail infrastructures. In physical networks interactions are mainly *causal*.

**Social networks** When dealing with social networks we explicitly consider the social science perspective. Social networks are seen to consist of interconnected actors, be it organizations or individuals. In these networks, the interactions are *intentional*. While human beings are physical entities and have physical interactions with each other and the world around them, we will consider only the social connections between people (Watts and Strogatz, 1998).

Connections between people are governed by social laws, rules and conventions and take many forms. These connections are as real as physical ones and vary in duration from very short and informal (e.g. interaction with a clerk in store, a passing acquaintance) to very long and fully formalized (e.g. governments, religious organizations, etc.) (Williamson, 1987).

Given that humans are both physical and social beings, they naturally interconnect the social and physical networks. Information flows between people can have large effects on physical realities. Flows of money between actors affect how mass and energy are allocated and used. Social networks are limited by laws of nature in the ways they shape physical networks. The flows of mass and energy through physical networks are governed by contracts, which are honored because of the legal structures of which they are a part, and which are formed and broken

in response to market dynamics. The structure and size of a physical network is determined by the private and public actors that decide and influence the investment in physical assets. It is the co-evolutionary interaction between these networks that causes the global sustainability problems discussed earlier. At the same time, the solution also lies in them.

### 2.1.2 Sustainability

**Modern society** Industrial society has become dependent on highly advanced and complex socio-technical networks (Castells, 2000). Due to global demographic, ecological, economic and geopolitical trends, these systems are being strained in their capacity (especially transport and energy networks) (Kraay, 1996), they are actively being threatened by political unrest (Farrell et al., 2004) and have an unacceptable and ever-growing environmental impact. These problems of the 21<sup>st</sup> century require a change in the way infrastructure networks, industrial networks, regional clusters and entire economies are used, managed and developed (Rotmans, 1998; Sterman, 1994). At the same time, the inter-dependencies between  $\lambda$ -systems and their surroundings, as well as the rapid changes in the internal components of  $\lambda$ -systems, make it more difficult than ever to understand them well enough to allow successful intervention (de Bruijne and van Eeten, 2007).

**Sustainable development** What, then, is sustainability? In the now famous words of the Brundtland (1987) Commission:

Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.

The basic idea behind this classic definition is the maintenance of a comfortable human existence on the planet. While this is something that everyone can agree with, the definition has a major oversight: it does not consider the world outside the human species at all. It is a purely anthropocentric perspective on sustainability (Achterberg, 1994). As such, it does not address multiperspective and multidimensional nature of the global system. A definition rooted in the systems perspective is needed.

**Emergent sustainability** The great difficulty with sustainability is that it cannot be defined at the individual level, nor at a particular time. An individual or a technology cannot be sustainable in itself. System-wide sustainability is an *emergent property* of the networked individual components in parallel action (Kauffman, 2000). Therefore, a new definition, based on the work of Allenby (2006) will be used in this thesis:

Sustainability is an emergent property of the global evolving multidimensional (social, technical and biogeochemical) networked system to indefinitely continue existing while retaining its ability to evolve.

**Evolving sustainability** Managing the evolution of sustainable  $\lambda$ -systems involves shaping the desired emergent behavior and avoiding the unsustainable emergent properties. This is somewhat like driving a race car through a minefield in a thick mist with 6 billion people all pulling at the steering wheel at the same time.

A sustainable state of the world cannot emerge by blindly doing things and hoping that they will work out for the best. Emergent behavior is a collective property of all system elements and their interactions. It is chaotic, as small changes in individual elements can potentially have large effects. It is path dependent, as past decisions will affect the availability of options for future decisions. Predicting which local interventions will lead to a sustainable state over time is difficult. As [Ehrenfeld \(1997b\)](#) argues, true sustainability will require a paradigmatic change in the structure of  $\lambda$ -systems that support human society. This means that we need to find a radically different way to address the design and management of  $\lambda$ -systems and to deal with their complexity.

### 2.1.3 Modelling the evolution of $\lambda$ -systems

This section will present some of the theories and concepts necessary to model the evolution of  $\lambda$ -systems. It is a short overview used to set the scene. For a detailed discussion of the theoretical backgrounds, please see [Chapters 3 and 4](#).

In order to start the discussion on how to understand and model the evolution of  $\lambda$ -systems we need to examine the three traditional approaches relevant to systems modelling. These approaches, described in the following paragraphs, are: Computable General Equilibrium modelling, System Dynamics and Process Systems Engineering.

**Computable General Equilibrium and System Dynamics** Static equilibria in the form of Computable General Equilibrium (CGE) models ([Jones, 1965](#); [Leontief, 1998](#)) have gained widespread attention, and the already large body of literature continues to grow. In this domain of neoclassical economics the emergent properties of  $\lambda$ -systems are the subject of analysis. Due to the nature of these models, they have little to offer in terms of steering  $\lambda$ -systems. These continue to be aggregate, static models that only implicitly include the generalized, aggregate behavior of consumers, producers and others.

In System Dynamics models, a system's structure is described and its response over time to a variety of inputs is examined ([Forrester and Wright, 1961](#); [Meadows et al., 1972](#)). While these models do not assume the world to be static, the system structure modelled is necessarily static. System dynamics models are typically set up to construct models of global system dynamics. System aggregation is applied as a rule, as the scope and goal of many models is to get a feeling of a system's overall dynamic response.

General equilibria models and system dynamics models have drawbacks for the modelling of  $\lambda$ -systems: they are top-down and employ system aggregation. Decision-making is not explicitly modelled. To mimic the evolution of socio-technical systems, we need models that account for the fact that in evolution the 'devil is in the detail'. It may be a single individual and his action that causes the overall emerging system behavior to change dramatically. Furthermore, it is not only system utilization, operation and behavior that changes with time but also system structure and content. Yet in neither of the above two classical approaches to modelling can such changes be included.

**Process Systems Engineering** The traditional engineering approach is based on designing system components, products and processes that are independent of each other ([DeLaurentis](#)

and Crossley, 2005). This approach serves us well as long as the component's interconnect-  
edness is minor and the social, economic and environmental impact is limited. Our thinking  
appears to be largely siloed in monodisciplinary thinking based on clearly delineated physi-  
cal systems in particular domains, e.g. applying chemical engineering in the chemical industry,  
aerospace engineering for aircraft design and mechanical engineering in automobile manufacture  
(DeLaurentis and Crossley, 2005).

The need for a different dimension of engineering knowledge in order to be able to design at  
a higher, integrated system level has been articulated by Westerberg et al. (1997). Subsequent  
progress in *Process Systems Engineering* (PSE) has led to more integrative approaches, through  
the use of superstructures, non-linear models, etc. (Dijkema, 2004; Grossmann and Westerberg,  
2000; Westerberg et al., 1997). This progress has mainly <sup>1</sup> focused on improving current proven  
technology and design at the processing plant level <sup>2</sup>. While this has paved the way towards  
more integrative systems thinking, the level is not sufficient for understanding and shaping  
 $\lambda$ -systems.

What we effectively do in the design of technical components of  $\lambda$ -systems is linearize these  
systems around a known point and then extrapolate to the point solution we seek. In other  
words, we assume a closed system boundary and a limited set of components that may be part  
of our solution. We then optimize somehow to find a 'best' solution - closed, fixed end-point  
problem solving (DeLaurentis and Crossley, 2005; Dijkema, 2004). However, in order to get  
to that 'best' system state, a variety of actors must decide on the system components. As a  
consequence, it often happens that by the time the design has been created and implemented,  
the system's internal structure has evolved and changed from the initial assumption. Mean-  
while, the conditions external to the physical system, including the stakeholders' preferences  
and objectives, may also have changed.

**Engineering  $\lambda$ -systems** Given the nature of sustainability as an emergent property of the  
global system, we must focus on the way  $\lambda$ -systems evolve.  $\lambda$ -systems are dynamic, multidimen-  
sional networks whose internal structures and functions change over time. While the technical  
sub-components of  $\lambda$ -systems *can* be engineered, their social networks and the emergent system  
structures currently *cannot*, even though through laws, regulations and customs we attempt to  
manage them. Their current states have evolved as the result of a series of discrete decisions  
made on the basis of the actions, interests and influences of the involved stakeholders. Their  
actions are driven by pressures exerted from the  $\lambda$ -systems' external environments and from the  
limitations of their internal structures. Global markets, national and international rules and  
regulations, availability of and access to capital, knowledge and skilled labor are examples of  
external pressures. Internal pressures are caused by, e.g., changes in the composition and pref-  
erences of the working populations, or by replacement of old assets with new, more advanced  
ones.

**Shaping evolution** The exact state of an evolving system is intractable (Dennet, 1996).<sup>3</sup> A  
system's being intractable means that no model or simulation can be predictive of the exact  
details of the evolutionary development of any particular  $\lambda$ -system in response to a given change.

---

<sup>1</sup>Notable exceptions are supply chain models by Perea-López et al. (2003), regional production planning  
models by Sung and Maravelias (2007) and a functional modelling approach by Dijkema et al. (2003).

<sup>2</sup>For examples of plant level integrative design, see (Ingram et al., 2004) and (Henrion et al., 2001).

<sup>3</sup>For an extensive discussion of evolution and intractability, please refer to Section 3.4 and Box 1.

Any evolving system is chaotic, i.e., potentially sensitive to small parameter value changes, plus it receives random input from the environment. However, evolving systems are path dependent and they have a robust, characteristic attractor structure (see Section 3.3). This means that evolving systems come *from* somewhere and that they preferentially 'want' to go *to* somewhere. Every evolving system has a stable *pattern of possible regions* (attractors) it can be in. Exactly which state the system will take is impossible to know in advance, but we might be able to estimate to which attractor (region) it will head.

By carefully changing a system's parameters, namely the identity of its components, their interactions or the system's environment, we can steer or at least alter the path of the evolutionary process. If we steer correctly, we *might* prevent the system from going towards an undesirable attractor, even if we do not have control over its exact state.

### 2.1.4 Generativist paradigm

**Generative Science approach** In the traditional scientific paradigm used since Bacon, in order to understand a system one has to break it down into small components to be tested and experimented upon in isolation. The assumption is that when each component has been understood in isolation, then the entire system has been understood as well. Such a reductionist paradigm results in models that remove the emergent and evolutionary properties of  $\lambda$ -systems, exactly the behavior that is of interest to researchers and useful for managers and planners. In order to understand the emergent properties, we need to think about describing the world in entirely new terms. The new holistic paradigm is called Generative Science (Epstein, 1999) and describes complex behaviors as generative processes. The central principle is that phenomena can be described in terms of interconnected networks of (relatively) simple units. In this approach deterministic and finite rules and parameters of natural phenomena interact with each other to generate complex behavior. Epstein (1999) poses the generativist's question as: "How could the decentralized local interactions of heterogeneous autonomous agents generate the given regularity?" The experiment that answers this question is: "Situate an initial population of autonomous heterogeneous agents in a relevant spatial environment; allow them to interact according to simple local rules, and thereby generate - or 'grow' - the macroscopic regularity from the bottom up."

**Reductionism vs. Holism** The generative science approach begs the question, how is this different from the traditional reductionist approach to science? To start, Dennet (1996) distinguishes acceptable and useful reductionism from *greedy reductionism*. Greedy reductionism, often arises when reductionism is applied too far, claiming that everything can and *must be* explained by the smallest possible parts. For example, a greedy reductionist approach would claim that consciousness must be explained solely in terms of electron movement. Holism on the other side is the idea that a system, be it physical, social or biological cannot be determined or explained by its component parts alone. Extreme application of holism states that it makes no sense to examine the components, and that we should only study the system in its entirety. Both greedy reductionism and extreme holism can not be sustained, especially when trying to understand  $\lambda$ -systems evolution.

The middle, as is often the case, is a more fertile ground. Reductionism in its milder form allows for new types of phenomena, or *epiphenomena*, to be caused by interactions of components. Reductionism implies that these epiphenomena exert no causality on the fundamental

phenomena that explain and cause them. However, this is where reductionism breaks, as almost any example of a complex system will possess the property of "reflexive downward causation". It is the property of a system as a whole to causally influence the state of its own constituents, which in turn determine the causal powers of the whole system (Kroes, 2009). This property comes straight from the holistic corner.

So, as reductionism denies that there is nothing new when speaking about epiphenomena, and holism says that they are real and cause downward causation, we need to examine the notion of emergence in more detail in order to be able to position this work.

**Role of emergence** According to Kroes (2009):

The following general characterization of emergence is our point of departure: emergent features in (complex) systems are 1) novel, qualitatively different features in comparison to the features of the system's parts, which 2) cannot be reduced to the features of those parts and their relations. Of course, we must explicate in more precise terms what the meaning of the notions of "novel", "qualitatively different" and "reduced". Following Van Gulick (2001: 16 sq) we distinguish between a metaphysical/ontological and epistemic reading of this characterization of emergence; the former concerns emergence with regard to real world items, the later with our representation of the world

The notion of emergence that will be used in this thesis is of an epistemic nature, not ontological. Ontological understanding of emergence leans to extreme holism, as all emergent properties are embedded in the "essence" of the system, and can not be examined by any reduction. The epistemic notion allows for a degree of (useful) reductionism, meaning that there is no "magic" involved in emergent properties, but that they can be understood as novel and surprising properties, arising from lower level element interactions. Furthermore, Kroes (2009) states that :

In an epistemic reading, notions such as "novel", "qualitatively different" and "reduced" are to be interpreted in terms of relations between knowledge of emergent features and knowledge of the features of the emergence base. This implies that relations of epistemic emergence "turn crucially on our abilities or inabilities to comprehend or explicate the nature of the links or dependencies among real-world items [the emergent features and the features of the emergence base]" (Van Gulick 2001: 16)

Emergent properties, according to the epistemic notion are not "greedily reducible", as emergent properties at the system level can and do influence the elements that they arise from. This leads us to the notion of simplification, that any model of a complex system must do.

**Simplification** Creating any model means taking a mental conceptualization of reality and formally codifying it in a model. In order for a model to be useful, it has to be simpler than the reality it is aiming to describe, while retaining all the aspects relevant to understanding the system. While the requirement of being simpler than the reality is fairly straightforward, being relevant is much more problematic. Relevance is a subjective property that is determined by the modeller. Depending on who is trying to understand the system, many different, and equally

valid models are possible. This property of Complex Adaptive Systems known as observer dependency will be discussed in more detail in section 3.5. Because of observer dependency there is no objective manner to determine what is a correct simplification, and it is impossible to remove the modeller from the model. In cases where an objectively measurable physical reality is being modelled, the task of simplifying is much more straightforward and less observer dependent than in our case, where real world, socially constructed aspects of  $\lambda$ -systems need to be encoded as well. As they are socially constructed, they can not be fully objectively determined. This means that there is a certain degree of arbitrariness in the models created. While this does not stand in the way of good scientific work, it is something modellers and model users must be aware of.

**Modelling approach** The main modelling approach used in this thesis is a generativist one, allowing for a certain degree of reduction, as after all, we are breaking down complex phenomena into lower system level interactions, and at the same time it allows a degree of holism, as it allows for reflexive downward causation as a natural part of the process of emergence. We aim to acquire an understanding of the evolution of  $\lambda$ -systems by describing the socio-technical aspects of the smallest system elements and their behavior. By using computers to systematically simulate the potential interactions and behaviors of these components, possible future system states are allowed to emerge. These emergent system states will be examined and inferences made about which actions are likely to lead to desirable, sustainable states in the systems under study. These insights can then be used in steering the evolution of real-world  $\lambda$ -systems.

## 2.2 Thesis Focus

This section will start with a recapitulation of the argumentation given so far. Next, the system level at which the work will focus is identified. Finally, the problem owner and the thesis domain focus are delineated.

**Recapitulation** So far the following argumentation has been presented:

- 1 The system Earth is an interconnected multidimensional fabric of  $\lambda$ -system networks.
- 2 These networks are continually evolving, due to both their internal dynamics and the changes in their environment.
- 3 Exact prediction of the direction of their evolution is impossible.
- 4 In order to explore patterns in a  $\lambda$ -system's evolution, a model based on the system's perspective and a generative approach are needed.

**Hypothesis** Models created in this way will enable better insights into the evolution of  $\lambda$ -systems and aid decision makers in shaping them.

These arguments are generic to any type of  $\lambda$ -system. It is therefore important to further specify which decision maker will be served, and which specific  $\lambda$ -system will be studied. First, the system level at which this work will focus will be discussed. Based on the choice of system level, the problem owner and the system under study will become apparent.

**System levels** It is possible to discern three system levels at which relevant  $\lambda$ -system problems and solutions can be placed. Please note that many different conceptualizations of system levels are possible, depending on what is conceptualized and who is doing the conceptualization. We present them as they will be used in this thesis.

**Macro level** The highest conceivable level of study is in this case the global (planetary) level. At this level the system behavior is dominated by the emergent properties caused by interactions from elements at lower levels. Examples of system level properties observable at this level are global warming, global population dynamics, world trade, global telecommunications, etc. While many urgent problems manifest themselves at the global level, it is difficult to directly affect them by acting at this level because, as already mentioned, their sources are rooted in lower levels.

**Meso level** At the meso level one can discern countries, regions, governments, etc. At this level the aggregate interactions of the smallest elements form intermediate entities that themselves interact to form the macro level. The meso level includes properties such as the eutrophication of river basins, regional energy networks and supply chains, and concentrations of various population groups. Problems at this level are somewhat easier to influence, as national and local governments have jurisdiction. Measures at this level can have large effects on global emergent properties. For example, the Kyoto Protocol was signed by parties at meso level, and if implemented fully could have the power to lower global CO<sub>2</sub> levels.

**Micro level** The smallest system elements are located at the micro level. Examples of elements at this level are individual persons, firms or pieces of equipment. The interactions and aggregations of micro-level elements form meso elements, which in turn interact to form the macro level. Examples of micro-level properties are the efficiency of a gas-burning central heating system in a household, the annual disposable income of a person or the profits of a firm. The properties of individual elements at this level can be directly influenced; for instance, the efficiency of a boiler can be improved, and income can be raised or lowered by taxes. However, changes in the observable behaviors of single individual elements often have very little or no effect on higher levels of aggregation<sup>4</sup>, and many micro elements must be influenced at once if one is to create emergent behavior at the macro level. For example, simply increasing the efficiency of the central heating system in my home will not solve the problem of global warming, but increasing the efficiency of all heaters installed in households in Europe will have a noticeable effect.

**Relations between levels** It is important to emphasize that the system levels presented above are just conceptual divisions, helpful when thinking in terms of systems. They are not real distinctions, as there is only one complex, networked system. The global economy defined at the macro level is created by the interrelations between countries, economies, etc. For example, the economy of The Netherlands is open for 55%. For every euro a Dutch citizen spends, 55 cents leaves the country and influences the economy elsewhere (Oosterwijk, 2006). System components do not interact only through money; energy and mass flows interconnect elements

---

<sup>4</sup>Obviously, Complex Adaptive Systems are chaotic, and a change in the behavior of a single element could potentially have very large effects. This is, however, very rare.

and levels, too. An average apple sold in a European shop can have up to 30,000 kilometers of transport embedded in it (Jones, 2002; Nielsen et al., 2003). In the systems perspective used in this thesis the system levels are interconnected. The macro level is the emergent property of the meso level and the meso level emerges from the micro level.

**Level of focus** The focus of this thesis is to examine the behavior of agents at the micro level through modelling and simulation, and to observe the effects on the meso level. The problem owner is defined at the meso level, as will be discussed below.

**Problem owner** The main problem owner considered in this thesis is the Regional Development Agency (RDA). The RDA is a typical actor at the meso level, shaping industrial networks in an evolving world over which it has largely no control. An RDA is a public or private organization whose primary role is to drive and sustain the socio-economic and ecological prosperity of the region under its control. RDAs do this by actively attracting investment, organizing infrastructure and creating suitable social networks. In some cases RDAs own land which they sell or rent to companies wishing to settle in the region. For many RDAs this represents a formidable task, responsibility and challenge. Not only must RDAs adapt their policies in a timely manner to a changing world, but the very economic structure of their region is shaped by decisions at the macro level over which the RDA has no direct control. Examples of effects outside RDA's span of control include companies investing in facilities to produce goods or services outside RDA's region, the EU setting new directives, or a national government constructing major infrastructures in competing regions. Depending on the type of activities, the relevant stakeholders may be part of a regional social network (e.g. a business park for life sciences start-ups) or one that effectively spans the globe (e.g. the chemical industry).

In the Netherlands, examples of RDAs are the port authorities like the Rotterdam Port Authority, Groningen Seaports and the Havenbedrijf Amsterdam or the Provincial development agencies, like the Provinciale Ontwikkelingsmaatschappij Limburg or the Investerings- en Ontwikkelingsmaatschappij voor Noord-Nederland.

**Focus on industrial networks** This thesis will focus on industrial and infrastructural networks, which are  $\lambda$ -systems. They consist of many physical artifacts and numerous social actors that design, maintain and control them. They have very long life spans: 15 to 30 years in the case of industrial plants, while some infrastructures have been in use for centuries. They have initial investment costs in the billions of euros and usually involve private-public funding and decision-making. They are deeply embedded in the social, economic and spatial structure and often have significant environmental impacts. Industrial networks are difficult to change because of path dependency caused by past investments and their large physical size. Society, however, depends on them to quickly adapt to new needs. RDAs often manage industrial regions that are full of industrial processes and infrastructures that support them. My own domain expertise as a chemical process engineer matches this.

## 2.3 Research Question

In this section the thesis hypothesis, the research objectives, the central research question and three sub-questions are defined.

### 2.3.1 Research Objectives

**Hypothesis** The main hypothesis of this thesis is that the use of models of simulated  $\lambda$ -system evolution can improve decision-making about industrial cluster development. By examining patterns of evolution across different simulation scenarios, we can reduce decision makers' uncertainty by answering 'what if' questions about  $\lambda$ -system evolution.

**Objectives** The objective of the work is to increase our knowledge of  $\lambda$ -system evolution patterns through simulation of the co-evolution of physical and social networks. The ultimate goal is to provide decision-making support for those involved in shaping the development of industrial clusters. The objectives can be stated as follows:

**Gain insight** in to the social, economic and technological aspects of the co-evolution of  $\lambda$ -systems, and more specifically, of regional industrial systems.

**Create a method** for compiling models that suitably represent the social and technical realities of industrial networks. These must comply with the laws of conservation of energy and mass and must enable exploration of the design space of sustainable industrial network evolution.

**Support decision makers** by creating a scientifically sound tool that could be used to support critical actors, notably the RDAs, in decision-making processes regarding regional industrial development.

### 2.3.2 Research Questions

Given the research objectives and the previous discussion, the central research question can be formulated as:

**How can we create a model for exploring the evolutionary patterns of  $\lambda$ -systems?**

Three subquestions can be derived from the central research question:

**RQ 1: How can a generativist Complex Adaptive Systems perspective be operationalized in models that capture  $\lambda$ -systems evolution?**

**RQ 2: What are the content specifications of such models in terms of the relevant formalisms (knowledge domains)?**

**RQ 3: What are the specifications for a method that would create such models?**

## 2.4 Readers' Guide

In the final section of this chapter, the intended audience is briefly discussed and the structure of the thesis is presented.

**Intended audience** In general, this thesis is meant for anybody who is dealing with  $\lambda$ -systems and is trying to better understand the behavior of these systems over time. As the thesis will cover both methodological and practical issues in operationalizing the evolution of  $\lambda$ -systems, it is interesting for both researchers and practitioners.

**Practitioners** Ideally, this thesis will be read by development managers seeking to understand the processes of evolution happening in their regions. It will provide the reader with ideas for how to build models of their local  $\lambda$ -system and a sense of which types of results to expect. A practitioner who is not interested in the theoretical background but is eager to find out about this study's practical implications is advised to read part II.

**Scientific** A researcher who is interested in understanding and modelling  $\lambda$ -systems will be well served by this thesis. Researchers from such diverse fields as industrial ecology, regional geography, economics and infrastructures policy will find the presented approach useful. The thesis centers around creating and underpinning a social process that effectively formalizes relevant knowledge which will lead to the creation of working models of  $\lambda$ -systems' evolutionary paths. The underlying concepts of complex adaptive systems evolution, of model development and of implementation and application are reported. The thesis will present many theoretical and practical insights into the social processes leading to model formalization. A series of case studies explains the technical details of simulation and points out the ways such models can be used, their limitations and their strengths.

The thesis is organized into three parts and eleven chapters. The structure is presented below in Figure 2.1.

**Chapter 1: Introduction** In this chapter the background is sketched and the research problem and goals are stated. The approach to solving the problem is presented.

**Part I Theory: Towards a co-evolutionary modelling method** In this part the theoretical background and the design of the modelling method are explored.

**Chapter 2: Research Theme and Question** In this chapter the systems perspective used in solving the proposed problem is presented. The focus and object of the study are delineated. The research goals and questions are presented.

**Chapter 3: Theoretical Foundations** In this chapter the first research question is explored. The results of a literature review are compiled into a single framework for understanding the evolution of Complex Adaptive Systems.

**Chapter 4: Modelling Foundations** In this chapter the second research question is explored. Through a literature review, the elements necessary for building evolutionary models of  $\lambda$ -systems are identified. Industrial Systems and Agent-Based Modelling are discussed.

**Chapter 5: Modelling method and requirements** In this chapter the third research question is explored. The method of co-evolutionary model development is presented. The requirements shaping the co-evolution between the social process for knowledge collection, technical design, collected knowledge and collected facts are addressed. The method steps are described.

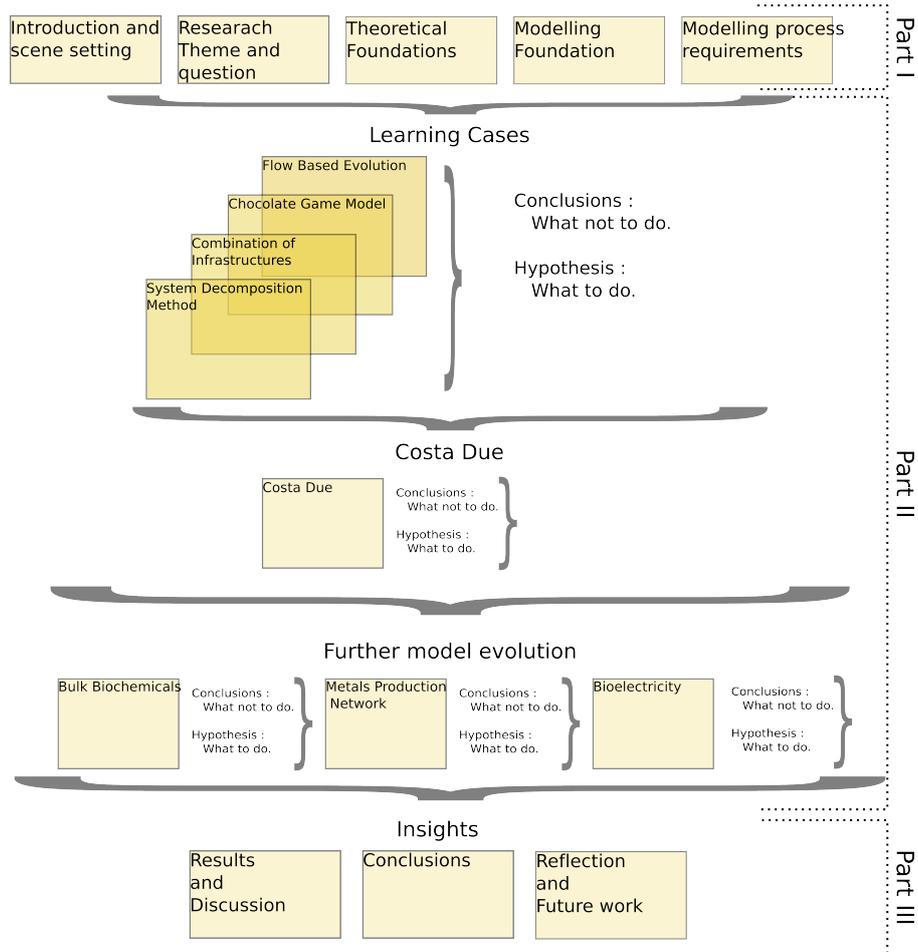


Figure 2.1: Structure of the thesis

**Part II Practice: Co-evolutionary method in action** In this part the development and demonstration of the method is reported in seven case studies.

**Chapter 6: Learning: Case Studies and Knowledge Engineering** This chapter presents the first three case studies, each having several important learning points. The Flow-Based Evolution model is the first network growth model presented, based on consuming and producing entities. The Chocolate Game models a  $\lambda$ -system based on the flow of discrete goods. The Combination of Infrastructures model examines the combinability of infrastructures and the knowledge representation needed to implement it. The main result of the learning case studies is the System Decomposition Method used for system formalization.

**Chapter 7: Full-scale Case Study: CostaDue** This chapter reports an extensive case study wherein a  $\lambda$ -system simulation is created of an existing industrial network in order to explore the options that will allow it to evolve towards a different state.

**Chapter 8: Method verification: evolution of three case studies** In this chapter three advanced models are presented, each with successive learning points. The Bulk Biochemicals case models a cluster exploring its economics environment. The Metals Network

model explores a different knowledge domain and simulates a global metals production network. Bioelectricity case is the most complex model presented, where all the lessons learned are brought together and a complete new formalism is added.

**Part III Insights** In this part the results and insights obtained are presented.

**Chapter 9: Results and Discussion** Here the results of the study are presented, some of the shortcomings are discussed.

**Chapter 10. Conclusions** In this chapter the thesis conclusions are presented.

**Chapter 11: Reflection and Outlook** The last chapter reflects on the modelling method and model outcomes, and provides directions for future work.

If you didn't grow it, you didn't explain its emergence...

(Epstein, 1999)

### 3.1 Introduction

**Goal and approach** The goal of this chapter <sup>1</sup> is to answer the first research question: How can a generativist Complex Adaptive Systems perspective be operationalized in models capturing  $\lambda$ -system evolutions? In other words, how can we systematically understand  $\lambda$ -systems evolution as a generative process based on the interactions of low-level components? The approach to answering RQ1 will consist of a literature review exploring views on complex systems from literature across different domains and disciplinary perspectives. The review will deliver the necessary building blocks for creating a unified conceptual framework that will enable us to systematically think about complex evolving  $\lambda$ -systems.

**Connecting formalisms** What this chapter attempts to do is connect different knowledge domains, or formalisms, that deal with similar issues. As already discussed in Chapter 1, Complex Adaptive Systems, such as  $\lambda$ -systems require multiple different formalisms to be fully described. The main assumption in this chapter is that a generativist systems perspective on  $\lambda$ -systems will ease the integration of multiple formalisms into a single, shared language. Creating such a language is not a trivial exercise; it requires significant effort.

**Shared language** Why is such a shared language important? As already discussed, when attempting to understand and describe  $\lambda$ -systems, multiple formalisms are required. If we understand multiple formalisms as multiple knowledge domains or fields, and if we assume that most people master only one field, this means that many different researchers and stakeholders need to communicate and understand each other if we are to understand  $\lambda$ -systems. In order for this understanding to happen, a shared language is needed.

---

<sup>1</sup>Parts of this chapter are based on

Even though English has evolved as the *lingua franca* of science, different disciplines actually do not speak the same language. The use of specialized jargons and terminologies by specific domains only slows the communication (NAS et al., 2005) across domain boundaries. Furthermore, academics may not question the meaning of words that they already know (Bracken and Oughton, 2006) and often assume that they are generally understood. This may even happen with very simple words like “infrastructure”, “network” and “system”, that can hold different meanings for different people. For example, where management scientists talk about *group think* to describe groups that have aligned their behavior in a social system, and economists talk about *lock-in effects* to describe the impossibility of change in organizations in the economic system, while physicists talk about *path dependent trajectories* and their *attractors*. The fact that terms that have different meanings for different people is an important source of miscommunication (Heemskerk et al., 2003).

This problem directly affects us when we attempt to understand  $\lambda$ -systems. Hansman et al. (2006) underscore the necessity for interdisciplinary research on  $\lambda$ -systems. In situations where multiple formalisms and knowledge domains need to be brought together, this issue of a shared language must be resolved.

**Unifying languages** Different multidisciplinary fields have attempted to deal with the problem of unifying languages in order to enable collaboration. A short literature review will examine the available options and their suitability for understanding and modelling  $\lambda$ -systems.

In the field of software design and engineering, the ‘Unified Modelling Language’ (UML) is the most notable example (Booch et al., 1999). UML is a standard language for specifying, visualizing and constructing software systems, designed by the Object Management Group <sup>2</sup>. It represents software constructs as visual elements and enables the direct translation from a software concept to a software implementation. The main problems with UML are that it is designed to deal only with software concepts and not to represent physical or social concepts. In the field of medicine biology, the ‘Unified Medical Language’ was created (Bodenreider, 2004; Lindberg et al., 1993) by the US National Library of Medicine. While spectacularly successful, the Unified Medical Language is domain-specific and thus unsuitable for  $\lambda$ -systems modelling. The field of learning and collaboration theory, decision support and common ground negotiations (Approach, 2007; Beers et al., 2007; Kirschner et al., 2008) has extensively studied the processes of creating shared languages between multidisciplinary groups. The work in these fields does not provide practical multidisciplinary languages. However, the theoretical concepts are useful when creating shared languages, as will be demonstrated in section 6.5.

In the Artificial Intelligence community, ontologies have been developed as a useful means of knowledge representation. Ontologies are formal descriptions of entities and their properties, relationships, constraints and behaviors that are not only machine-readable but also machine-understandable. When two agents are communicating about certain concepts, we want to be sure that they interpret these concepts in the same way. Therefore, it is of utmost importance to unambiguously specify each concept and its meaning, i.e., create a standard interface by defining a common language (McGuire et al., 1993; Mizoguchi et al., 1997). The meaning of each concept is stored not only in the subclass relationship *is*, e.g. apple is a fruit, red is a color, but also in the property relationship *has a*, e.g. an apple has a red color. In other words, an ontology contains explicit formal specifications of the terms within a domain and the relations

---

<sup>2</sup><http://www.omg.org/>

among them (Gruber, 1993).

Ontologies, together with the concepts defined by collaboration theory, provide us with a basis for exploring formalisms.

## 3.2 Formalisms

**What is a formalism?** A formalism can be seen as a pre-defined representational knowledge format. We define it as a set of concepts and their relations (ontology) that is used to represent knowledge, plus a set of accompanying rules for using the formalism. The concepts and relations in a formalism are often specific to a particular task. For instance, Suthers (2001) developed a formalism called Belvedere to enhance scientific discourse.

**Why use them?** The benefits of using formalisms are twofold. First, they can add task rationality by representing a task conceptually. For instance, specific formalisms have been developed for: the support of collaborative design tasks (the Questions-Options-Criteria approach of Buckingham Shum (1997)), negotiation of common ground (the Negotiation Tool of Beers et al. (2006)), and argumentation (the Issue-Based Information System of Conklin and Begeman (1987)). All of these are based on an analysis of the task they aim to support. Second, formalisms provide a common interface for collaboration between researchers. In this way existing differences among disciplinary jargons are alleviated by adding a common component to the discussion that is understood and used by all collaborators.

**How to use formalisms?** The art of using formalisms is to use one that is fit for the task it is meant to support (Van Bruggen et al., 2002). When creating models, one should use a formalism that is in line with the type of model one is constructing. It is of little use, for instance, to use a linear programming formalism when one is constructing an agent-based model. If, on the other hand, one is busy making a system-dynamics model, it helps if the domain experts reconceptualize their own knowledge in terms of stocks, flows, feedbacks and delays (Senge et al., 1994; Sterman, 1994). Likewise, for agent-based modelling, using a tailor-made agent formalism would assist modellers and domain experts in their collaboration.

**Creating formalisms** To effectively make a formalism, one should first analyze the task at hand in order to identify the basic building blocks that constitute it (task primitives). Next, rules need to be defined that describe how these building blocks are to be used, so that the task can be effectively carried out. The formalism thus consists of the task primitives and the rules for using them. While using a formalism, one should make sure that all collaborators have a shared understanding of it and act accordingly. This thesis will present the System Decomposition Method (SDM) in section 6.5, where the task to be structured is the agent-based modelling (ABM) of  $\lambda$ -systems. The SDM is based on an analysis of what ABMs are and how they are designed, with the aim of supporting multidisciplinary modelling teams and their exercises by delivering a standardized modelling procedure.

**Need for ontologies** In the case of understanding industrial networks, Allenby (2006) makes the following argument in support of using ontologies:

One of the interesting aspects of dealing with complex systems is that the boundaries of the appropriate system are determined by the query which one poses to the system (Allenby, 2005). Thus, for example, if I ask what the police arrest rate is for New York City, I have implied by my query the existing political boundaries of the City. If, however, I ask what the water supply infrastructure is for New York City, I have included at least a third of the State of New York, which has been legally structured, and engineered, to provide water supplies through a complex infrastructure to the City (Gandy, 2002). Following Sarewitz (2004), therefore, one can make the observation that any industrial ecology study is equivalent to querying a complex adaptive system (the economic and industrial system) and thus implicitly defining the relevant boundaries for the inquiry. This necessarily involves at least two inseparable but very different ontologies: the personal and cultural normative ontology the individual researcher brings to the query, which defines a set of boundaries that, from the perspective of the overall systems, are essentially arbitrary, and the “scientific” and more objective process the researcher applies to the particular study.

Allenby identifies the need for ontologies and highlights an important problem. As already discussed, multiple formalisms are at the base of any  $\lambda$ -system. Therefore, many ontologies can be created for any one  $\lambda$ -system, which might be incompatible with each other. In this thesis I hope to offer a practical method for resolving this problem by creating an ontology that is explicitly meant to be continually added to by the participants through collaborative research (see Chapter 7).

**Formalizing Complex Adaptive Systems** Having defined what formalisms are and how they are used, it is time to describe the main formalism used in this thesis, namely the formalism of Complex Adaptive Systems. In the following section Complex Adaptive Systems thinking and literature is reviewed. The insights gained are used to compile a single framework to be used in communicating knowledge across disciplines.

## 3.3 Complex Adaptive Systems

This section explores complexity and Complex Adaptive Systems from many different fields and identifies shared concepts. The complexity definition used in this work is presented and discussed, as is a three-layer approach to systemizing Complex Adaptive Systems thinking from a generativist systems perspective.

### 3.3.1 Complexity Across Disciplines

**Across disciplines** There are so many scientific fields that have relations to complexity that a complete review is not feasible. The literature review performed here will therefore be limited to fields that relate to  $\lambda$ -systems and their modelling, design and management. Complex systems and complexity have gained increasing attention in diverse scientific domains, such as ecology (Ehrenfeld, 2000), management science (de Bruijn and ten Heuvelhof, 2000; Koppenjan and Klijn, 2004), thermodynamics (Prigogine and Stengers, 1984), economics (van den Bergh and Janssen, 2005), sociology (Wasserman and Faust, 1994, reprint edition 2005), system dynamics

(Forrester, 1958; Senge et al., 1994; Sterman, 2000), Complex Adaptive Systems (Bertalanffy, 1968; Holland, 1997, 1996; Kauffman, 1993), Chaos Theory (Gleick, 1988) and educational psychology (U. and M., March 1999). Examples of complex systems described in these fields are ecosystems, multinationals, the Internet, the human body and social systems.

**Different names** Ubiquitous as Complex Adaptive Systems are, many different names have been given to them in different fields. Examples relevant to  $\lambda$ -systems are terms like: complex socio-technical systems (Bonen, 1981), socio-technical systems (Geels, 2004), large technical systems (Bijker et al., 1987), complex innovation systems (Katz, 2006), complex engineering systems (Ottens et al., 2006) and system of systems engineering (DeLaurentis and Crossley, 2005). We present a range of definitions used in different fields, namely artificial intelligence, management sciences, economics, systems analysis and agent-based modelling.

**Complex Adaptive Systems** Below is a definition from the field of Complex Adaptive Systems by Gandolfi (1999), adapted by Bertuglia and Vaio (2005):

An adaptive complex system is an open system made up of numerous components that interact with one another in a nonlinear way and constitute a single, organized and dynamic entity, able to evolve and adapt to the environment.

John H. Holland (Waldorp, 1992) defines a Complex Adaptive System as:

...a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a CAS tends to be highly dispersed and decentralized. If there is to be any coherent behavior in the system, it has to arise from competition and cooperation among the agents themselves. The overall behavior of the system is the result of a huge number of decisions made every moment by many individual agents.

The definitions of Holland and Gandolfi indicate that complex adaptive systems are characterized by diverse agents that interact in a dynamic, open network. This interaction results in an overall system that evolves and adapts to its environment.

The above definitions are from the field of complex adaptive systems. Let us also explore some other definitions of complex systems, used in other fields, that are relevant to understanding  $\lambda$ -systems. Please note the variety, breadth and depth of these definitions.

**Operations Research** In the Operations Research literature, complex systems are described as follows (Miser and Quade, 1985):

...structures that combine people and the natural environment with various artifacts of man and his technology.

**Policy Analysis** In the related field of Policy Analysis, a complex system is defined as follows (Walker, 2000):

A [complex] system includes people, social structures, portions of nature, equipment and organizations; the system being studied contains so many variables, feedback loops and interactions that it is difficult to project the consequences of a policy change. Also, the alternatives are often numerous, involving mixtures of different technologies and management policies and producing multiple consequences that are difficult to anticipate, let alone predict.

These descriptions are more domain-specific than the definitions from the field of CAS. The above definition given by Walker is far more focused on solutions and intervention. However, the definitions do acknowledge the diversity of components and interactions. They also emphasize that the behaviors of complex systems are hard to predict or anticipate because of the interactions and feedback of the components.

In General Systems Theory and related fields, the following definitions can be found.

**System Dynamics** In the System Dynamics field, a complex system is described as (Maani and Cavana, 2000):

...a collection of parts that interact with one another to function as a whole. However, a system is more than the sum of its parts; it is the product of their interaction. A system subsumes its parts and can itself be part of a larger system.

**Organizational Learning** The Organizational Learning field describes a complex system as follows (Senge et al., 1994):

A system is a perceived whole in which the components “hang together” because they continually affect each other over time and operate toward a common approach.

These definitions betray a top-down orientation in these fields. A system is considered foremost as a single entity. However, like in CAS, these definitions also acknowledge that system behavior arises from component interaction.

**Educational Psychology** In educational psychology, complex systems are described as follows (Hmelo-Silver and Azevedo, 2006; U. and M., March 1999):

Complex systems have a hierarchical nature and have multiple interacting levels.

This definition mainly discusses the importance of hierarchy and interaction between system levels, adding the concept of levels to the composite definition we are building.

**Public Administration** In mainly social fields the term ‘network’ is used to denote a complex system; below we provide examples from public administration. The public administration field describes a system as a network consisting of (de Bruijn and ten Heuvelhof, 2000):

...a dynamic entirety of actors, mutually dependent, experiencing variety, and relatively closed vis-a-vis to one another.

Similarly, the concept of policy networks is defined as ([Kickert et al., 1997](#)):

...(more or less) stable patterns of social relations between interdependent actors, which take shape around policy problems and/or policy programs.

In the definitions above, networks are viewed as collections of parts which are bound to a certain problem or program and which interact with one another to function as a whole. These definitions, like those in other fields, acknowledge that interactions between individuals lead to certain structures and behaviors.

**Insights** Examining the literature and example definitions provides us with important insights. The necessary components for a holistic, integrative and generativist understanding of Complex Adaptive Systems are present but not yet integrated. Each definition presented above, originating in its respective field, contributes a piece that can be used in a new, unified definition. At present there is a lack of a coherent vocabulary, and the view of systems differs widely between disciplines. In order to proceed and create a coherent and generativist view, the two definitions that together offer a complete view that suits the multiformal and generativist systems view will be presented. Based on this, a unified generativist Complex Adaptive Systems framework will be constructed.

### 3.3.2 Complexity Defined

The two definitions presented below will be used in concert throughout this thesis.

**Multiple formalisms** [Mikulecky \(2001\)](#) defines complexity as being:

...the property of a real world system that is manifest in the inability of any one formalism being adequate to capture all its properties. It requires that we find distinctly different ways of interacting with systems. Distinctly different in the sense that when we make successful models, the formal systems needed to describe each distinct aspect are *not* derivable from each other.

Please note that Mikulecky further narrows the concept of perspectives as formalisms. Formalisms should be understood as formal, encoded languages used to express something. Mathematics and psychology are examples of formalisms that are not derivable from each other<sup>3</sup>. Paraphrasing Mikulecky, integrating knowledge of various domains and disciplines is essential for capturing in full the properties and behavior of any system. Assuming that most people master only one or very few formalisms or disciplines, the definition essentially says that one cannot understand a complex system alone. This definition forms the basis for the multiformal and multidisciplinary study of Complex Adaptive Systems. Furthermore, one could say that modelling  $\lambda$ -systems requires that knowledge from many experts be transformed into a model able to hold different formalisms.

Mikulecky's definition is rather abstract and cannot be directly made operational. For a practical definition we need to turn to John Holland.

---

<sup>3</sup>Even though Hari Seldon ([Asimov, 1989](#)) will disagree some 20,000 years from now.

**Dynamic network** John H. Holland ([Waldorp, 1992](#)) defines Complex Adaptive Systems as:

...a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a CAS tends to be highly dispersed and decentralized. If there is to be any coherent behavior in the system, it has to arise from competition and cooperation among the agents themselves. The overall behavior of the system is the result of a huge number of decisions made every moment by many individual agents.

A Complex Adaptive System (CAS) is a system consisting of many agents that interact with each other in various ways. Such a system is *adaptive* if these agents change their actions as a result of the events in the process of interaction. Since this is the case with  $\lambda$ -systems, we can conclude that they are Complex Adaptive Systems. The system structure and content can be changed from within the system by agents that develop new behavior through learning and strategic behavior, introduction of novel agents or adoption of novel technologies. This is confirmed by [Kay \(2002\)](#), who also identified similar characteristics of  $\lambda$ -systems.

By adopting a CAS view and by including multiple formalisms, one can observe dynamic patterns of system behavior emerging from local interactions between system components ([Holland, 1996](#); [Kauffman and Johnsen, 1991](#); [Newman, 2003](#)). System structure is not prescribed in advance. Interaction occurs at multiple dimensions and across multiple levels. Thus, while the span and nature of local interactions is explicitly specified, the total system behavior is not; it emerges. CAS theory recognizes that in order to understand the evolution of real Complex Adaptive Systems, a multitude of perspectives is required to capture the richness of their dynamics and the behavior of their components.

### 3.3.3 System Levels

From the definitions presented above, three system levels become apparent. The lowest or agent level, the middle, or network level and finally the top or system level. It is important to realize that these levels do not exist per se and are conceptualizations only; they create a useful framework with which to view and understand a system. Three presented levels form the basis of the Complex Adaptive Systems framework. The levels are depicted in [Figure 3.1](#) and discussed in the following paragraphs. [Figure 3.1](#) represents a system situated in some environment.

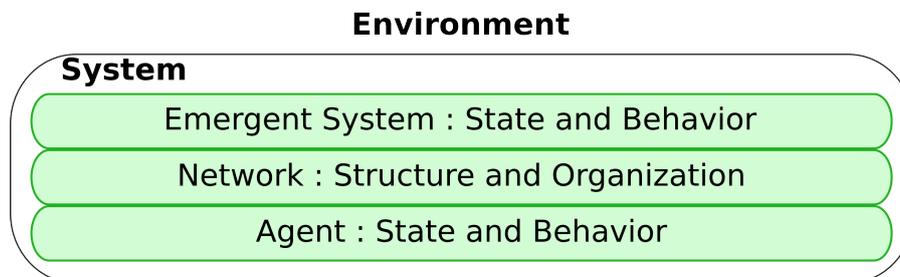


Figure 3.1: Conceptual levels in Complex Adaptive Systems

**Agent level** The level of the smallest system components is the agent level. It corresponds to the micro level discussed in Chapter 2.2. Agents are abstractions of the lowest level entities, the individual components from whose properties and interactions all system behavior emerges. It is the level that drives the generative nature of the framework. The definitions presented <sup>4</sup> all refer to these lowest level agents.

**Network level** The second level describes the structure of interactions between the agents of the system. When observing the system at this level we perceive the structure of agent interactions as a coherent entity, a network. Networks are a natural abstraction for describing things and their relations as nodes and edges. The network level corresponds to the meso level discussed in Chapter 2.2 See definitions referring to the organization of agents <sup>5</sup>.

**System level** The system level presents the emergent system properties caused by agent-level individuals and their properties, interacting through networks at the network level <sup>6</sup>. The system level corresponds to the macro level discussed in Chapter 2.2.

**Box 3: System levels example** In order to illustrate the Agent, Network and System levels, a concrete example will be presented. Let us consider the schooling behavior of fish (Bonabeau et al., 1999). It is one of the most mesmerising of natural phenomena: many thousands of individual fish behaving as a single organism, evading predators and maintaining a tight school. Let's examine the system.

**Agent** Each fish in a school is a relatively mindless automaton, following four basic rules. First, separation. A fish will preferably maintain a certain distance between itself and its neighbors in order to avoid collision. Second, alignment. A fish will try to follow the average direction and speed of its neighbors. Third, cohesion. A fish will steer towards the average position of its neighbors. Fourth, self-preservation. When a fish feels threatened by a predator, it will flee.

**Network** Each fish observes its neighbours and watches out for predators. A school is effectively a very large and dense network of perception, action and reaction. If a fish at the fringe of the school notices a predator and starts to flee, neighbouring fish react to its movement by adjusting their own speed and position. Because *everybody is watching everybody*, the signal that a predator is attacking travels quickly through the network of fish interaction.

**System** At the system level, coherent and coordinated behavior emerges from each individual acting and reacting through the interaction network of many individuals. The emergent school behaviour can be considered an entity in itself, maintaining cohesion and evading predators as if it were a single organism.

---

<sup>4</sup>(de Bruijn and ten Heuvelhof, 2000; Gandolfi, 1999; Maani and Cavana, 2000; Miser and Quade, 1985; Senge et al., 1994; Waldorp, 1992; Walker, 2000)

<sup>5</sup>(de Bruijn and ten Heuvelhof, 2000; Gandolfi, 1999; Maani and Cavana, 2000; Miser and Quade, 1985; Newman, 2003; Senge et al., 1994; Waldorp, 1992; Walker, 2000).

<sup>6</sup>(Gandolfi, 1999; Maani and Cavana, 2000; Senge et al., 1994; Waldorp, 1992)

**Approach** Appendix A presents the components of the complexity framework. In this appendix, a detailed description of each level (Agent, Network and System) is presented, its dominant field’s vocabulary is adapted and the most relevant properties of the individual levels are discussed. Where necessary, examples are provided from other fields to illustrate alternative meanings of the properties. The insights from these levels will be brought together in a unified generativist framework (see Section 3.5) describing how higher system level behavior emerges from lower levels, forming a continuum from the small to the large.

**The next step** The defining property of Complex Adaptive Systems is adaptivity, or the ability to change. Evolutionary theory will be introduced in the next section as a comprehensive theoretical framework for understanding system change.

## 3.4 Evolution

First, a short and basic introduction to biological evolution will be given. Second, evolutionary theory is translated to the domain of  $\lambda$ -systems. Finally, two concepts most relevant to this thesis, intractability and co-evolution, will be discussed in some detail.

As discussed in Chapter 1,  $\lambda$ -systems can be seen as systems that evolve. The conjecture is that to view them using an evolutionary perspective will yield insights useful in understanding and shaping them. In this thesis, evolutionary theory serves as a pervasive background theory. It views all changes in  $\lambda$ -systems in this light. It does not seek to further develop evolutionary theory itself, but to apply it. This means that the main question answered in this section is that of how to operationalize the concept of evolution for  $\lambda$ -systems.

### 3.4.1 Biological Evolution

An enormous body of knowledge exists on the evolution and co-evolution of biological systems. It started with the publication of Darwin’s book “On the Origin of Species” (Darwin, 1985) and has developed into one of the best researched and supported scientific theories today. In this section the basic tenets of the biological evolution theory will be presented, and some aspects relevant for this thesis will be discussed.

**Basic view** Darwinian evolution is summed up in the famous maxim (Darwin, 1985): “Vary, multiply, let the strongest live and the weakest die”. The theory of evolution can be seen as a generativist theory, as it describes how higher level system behavior (species changing over time) emerges from the behavior and interactions of low-level components. When organisms reproduce, their offspring are born with slight modifications in their genes, introduced through mutation. These changed organisms display a fitness that is different from an unmutated individual. While often negative for an individual’s fitness, some mutations improve an organism’s ability to grow and reproduce successfully. Over time, fitness-improving mutations accumulate, creating new forms. While this is a strongly simplified description of the actual process, it is sufficient for our needs.

**Randomness and the environment** The carrier of biological information, the DNA molecule, is highly sensitive to the only true source of randomness in the universe: the quantum-driven decay of atomic nuclei and its resulting electromagnetic radiation (Erber and Putterman, 1985). Every organism and its DNA exist within an environment permeated with radiation, which acts as a permanent driver of DNA mutation. The environment may be static but is mostly periodic within the lifetime of an organism. Yearly climate patterns and diurnal rhythms are an example of this periodic regularity. Organisms are adapted to and selected by this relatively predictable and ordered environment<sup>7</sup>. This order, or regularity, is the conceptual counterpart of the randomness affecting DNA. Life needs both a source of randomness and a source of order (Levin, 2000). Without the first it cannot change; without the second it cannot survive. The concepts of randomness and order are important aspects that also translate into the domain of socio-technical evolution.

**Multidimensional interactions** As already mentioned, many different individuals and many different species interact with each other in an ecosystem. Not only is there a great diversity among the types of organisms, but also in the way they interact with each other. Sometimes they are direct adversaries, preying on each other, competing for resources or being parasites. Sometimes the interaction is beneficial, and species live in symbiosis, their lives depending on each other. These interaction networks carry mass and energy in the case of predation, information in the case of pollination, etc. A mass flow used for energy generation is different from an information flow, even though both are based on a physical movement of DNA molecules. This diversity of interactions needs multiple formalisms to be described and is one of the reasons for the complexity of ecosystems. This multiformal/multidimensional nature of evolving system is even more apparent in socio-technical systems.

### 3.4.2 Socio-technical evolution

**Is it evolution?** Traditionally, evolution has been a term reserved for a Darwinian process that shapes living beings. Can it be applied to  $\lambda$ -systems? Yes, it can, as  $\lambda$ -systems are constantly constructed, used, changed and deconstructed by living, evolving human actors. Obviously,  $\lambda$ -systems have no DNA, so the processes of variation, reproduction and selection must have a different mechanism. Dawkins (1990) suggests that a cultural, self-replicating entity, analogous to a gene, called a *meme* is responsible for the spread, construction and evolution of culture. In the words of Dawkins and Dennet (Dawkins, 1990; Dennet, 1996):

A meme is the basic unit of information which spreads by copying from one site to another and obeys, according to Dawkins, the laws of natural selection quite exactly. Meme evolution is not just analogous to genetic evolution. It is the same phenomenon. Cultural evolution simply uses a different unit of transmission evolving in a different medium at a faster rate. Evolution by natural selection occurs wherever conditions of variation, replication and differential 'fitness' exist.

Examples of memes are traditions, technologies, theories, rules and habits. This thesis, for example, can be seen as a meme that might or might not survive over time, depending on its

---

<sup>7</sup>Of course, the environment contains catastrophic change events. These are usually so disruptive that a great majority of species becomes extinct. Nothing can evolve to cope with extremely rare, sudden and utterly devastating global catastrophes.

usefulness to the persons who are aware of its contents and find it useful enough to tell others about it. If human culture is indeed meme-based, there are several parallels and differences between biological and socio-technical evolution.

**Similarities** Ziman argues (David, 2000) that there are *structural* and *phenomenological* similarities. Structurally, both the biological, gene-based system and the socio-technical, meme-based system have variation, selection and reproduction. Both social and genetic evolutions consist of a succession of generations. In living systems these are determined by the reproduction cycle of the organism. In social systems, generations of ideas, technologies and social systems replace one another. Successful designs and ideas change and are passed on, and ineffective ones are disused and die out. Ziman observes (David, 2000) phenomena in socio-technical systems that are very similar to the phenomena of biological systems, such as diversification, speciation, convergence, stasis, evolutionary drift, satisficing fitness, developmental lock, vestiges, niche competition, punctuated equilibria, emergence, extinction, co-evolutionary stable strategies, arms races, ecological interdependence, increasing complexity, self organization, unpredictability, path dependency, irreversibility and progress.

**Differences** Ziman further notes that the most obvious difference is that technical and social artifacts are not generated randomly but are purposefully designed. This of course does not mean that randomness has no role. In fact, randomness is so important in the design of  $\lambda$ -systems that they are described as being serendipitous (Roberts, 1989). This serendipity may involve a sudden inspiration during the design process, an accident that creates something useful, or even the accidental choice for the designer that gets the task.

Another difference is that there is no strict technical equivalent of a gene, other than the concept of memes. Furthermore, the speed of evolution is much faster in culture, since there is no need for the *meatspace*<sup>8</sup>. It mostly just happens in the noosphere<sup>9</sup>, the shared human knowledge space, and not in the biosphere. Finally, created artifacts are not alive, they do not reproduce, and the human and biological ecosystem is relatively inefficient in recycling them. Some of these social constructs have a relatively weak physical basis, a very large pool, or randomness and a very high degree of connectedness. Just think about all the crazy memes you have ever seen on the Internet<sup>10</sup>. Yet evolution it is. Both biological and socio-technical evolution share the two most important properties for this thesis, intractability and co-evolution.

**Relevant properties** Based on the previous discussion of Darwinian evolution and the differences between biological and socio-technical evolution, we can distill the two most relevant properties needed for understanding  $\lambda$ -systems. These are the intractability of evolutionary processes and the concept of co-evolution.

**Intractability** Intractability is important as it defines the impossibility of the exact prediction of an evolutionary process. It therefore clearly presents the problem we face when trying to understand and steer the evolution of a  $\lambda$ -system. It can be mathematically

---

<sup>8</sup>Meatspace is the opposite of cyberspace. See also <http://en.wikipedia.org/wiki/Meatspace>

<sup>9</sup><http://en.wikipedia.org/wiki/Noosphere>

<sup>10</sup>For instance, <http://www.youtube.com/watch?v=eBGIQ7ZuuiU> had more than 20 million views in early September 2008.

proven that we will never truly know the precise effects of our actions, and we must thus act accordingly.

**Co-evolution** The concept of co-evolution is important as it makes it clear that nothing exists or evolves in isolation. Every action of every element in an evolving system will have some effect on all other elements. The concept of the coupled fitness landscape helps to visualize the interaction space. If we are ever to grasp the evolution of  $\lambda$ -systems, we must understand the co-evolving elements within them and the co-evolution of the  $\lambda$ -system with its surroundings.

Intractability and co-evolution will be discussed in greater detail in the following two subsections.

### 3.4.3 Intractability

**Algorithmic process** Evolution is an algorithmic process (Denett, 1996) of mutation, reproduction and selection across the evolutionary design space. Computational theory (Hartmanis et al., 1983) states that evolutionary problems are intractable, that is, they are *not* NP complete. NP completeness means that the computation time of an algorithm on a deterministic Turing machine is not greater than a given polynomial function of the problem size,  $n$ . The evolutionary algorithm does *not* have this characteristic. It is *NP incomplete* and exists in what is known as the EXPTIME, see equation 3.1.

$$EXPTIME = \bigcup_{k \in \mathbb{N}} DTIME(2^{n^k}) \quad (3.1)$$

In equation 3.1 DTIME is the computation time of a deterministic Turing machine. It represents the amount of time (or number of computation steps) that a 'normal' physical computer would take to solve a certain computational problem using a certain algorithm.  $n$  is the size of the problem input and can be understood as the number of parameters.  $k$  is the number of steps needed to find the solution. In the case of open ended problems, like evolution, it is the number of (evolutionary) steps performed.

**Scale of intractability** Let us provide a sense of scale for the intractability level of EXPTIME problems. Let us imagine an impossible scenario in which each electron in the universe ( $10^{79}$ ) had the computational power of today's fastest supercomputer ( $10^{12}$  instructions per second), and each worked for the entire age of the universe ( $10^{17}$  seconds) on solving the problem. This means that such an *ubercomputer* would perform  $10^{108}$  computations<sup>11</sup> in its entire existence. Now let us imagine an evolutionary process with 100 variables, evaluated over 100 time steps. In order to examine all possibilities, and thus be able to predict the outcome in advance would require  $2^{100^{100}}$ , or  $2^{10E199}$ , calculations to be performed. Now imagine how many components (bacteria, ants, animals, humans, computers etc) have interacted in an evolving system line on planet Earth over its lifetime of  $4.5 \times 10^9$  years...

---

<sup>11</sup><http://www.cs.princeton.edu/introcs/77intractability/>

**Full execution** Paraphrasing Equation 3.1, the intractability of an algorithm means that there is no faster way of evaluating the outcome of the algorithm than simply to go through all the steps and observe the outcome. The corollary of this is that the outcome, the system structure and state of the evolutionary process, cannot be predicted in advance. Intractability implies that the outcome of the evolutionary 'program' can only be found by completing its execution.

**Illustration** The cartoon in figure 3.2 depicts an evolving civilization that eventually causes its own demise by accidentally hitting the 'Universal Reset Button' somewhere in its evolution. Due to the intractability of its evolution, there is no way the civilization could predict or be aware of this event. To a cosmic observer the society's evolution is stuck in an endlessly repeating loop; the society does not have the ability to ever find out about it in advance.



Figure 3.2: Intractability of evolution (Gurewitch, 2005)

**Cause of intractability** So what causes the intractability of the evolutionary process? As already discussed in Chapter 1, in a system every component interacts with every other component. Every time something happens, a decision is made, etc.; it has an effect on all other things that can happen in the future. The rest of the system reacts, causing something new to happen, etc, etc. An illustration of this process is presented in Figure 3.3.

Let us imagine a system being at some arbitrary point 0 in the system's history. As time goes on, an event happens at time A, causing the system to subsequently move towards point A. All other possible states toward which the system could have evolved are no longer possible. At time B, another interaction event happens, again excluding countless possible future states. As the system progresses in time, across points C, D, E, F, etc., more and more of the astronomically large number of possible system states are not able to come into being. Of course, at each time step, the same astronomical number of new possible states continuously becomes possible, as can be see at point H.

**Forest analogy** Another analogy may help to grasp the concept of intractability. Imagine that you are walking on a narrow path through a very dense and dark forest (see Figure 3.3). After each step you try to guess where the path is leading and what your next step should be.

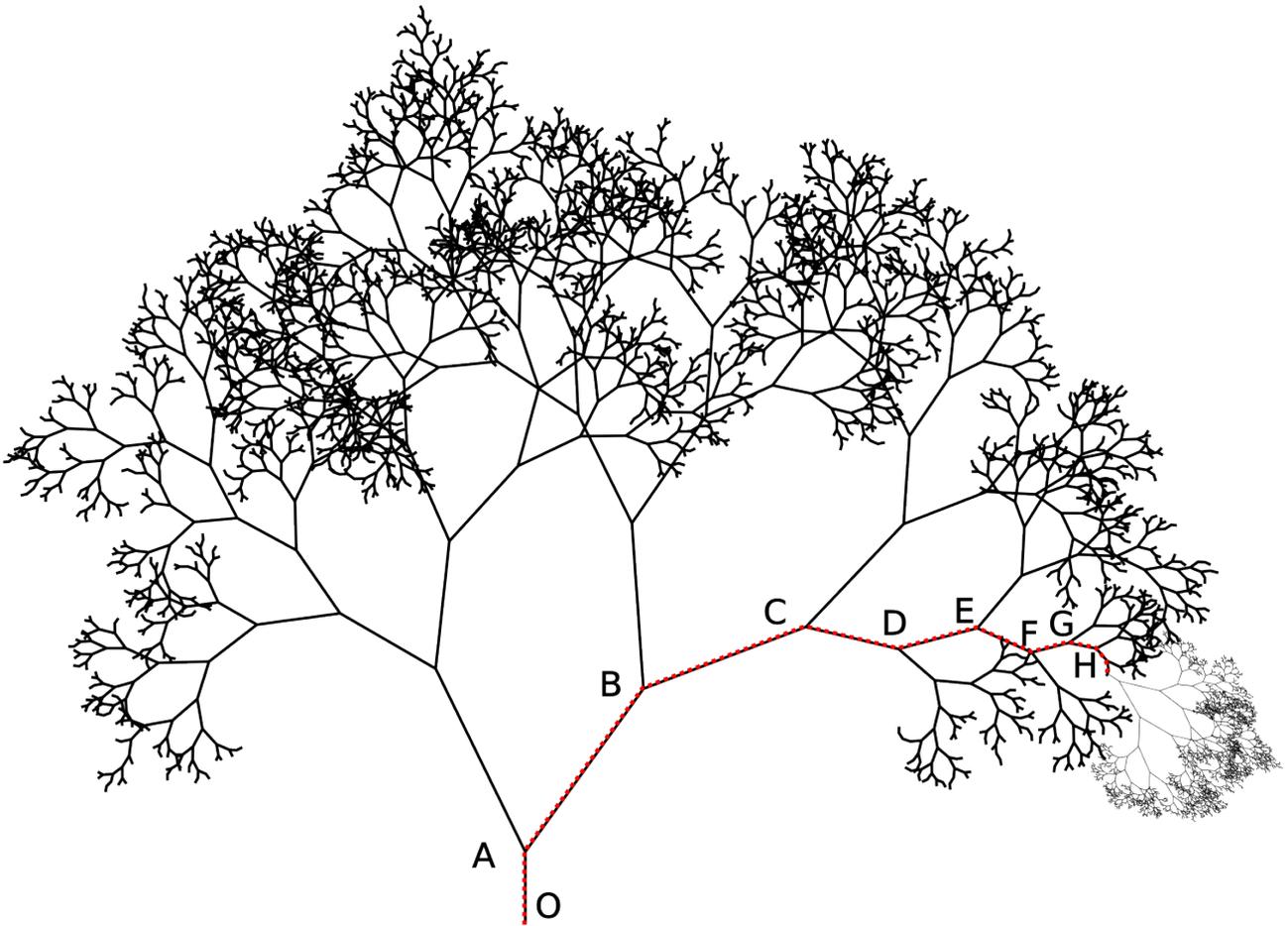


Figure 3.3: Possible and actual path of a system

The forest reacts to your every step by changing the direction of the path in front of you in some unknown way. Predicting the direction of the path is only possible if you take the reaction of the forest to each and every one of your steps into consideration. However, you do not know how the forest will react. To find out where the path leads, you must simply walk it.

**Relevance** The main consequence of the intractability of the evolutionary process is that the *exact prediction* of the evolutionary outcomes of  $\lambda$ -systems is impossible. We need to find other ways to predict, or at least better estimate, what the future may bring us, especially if we intend to change it so as to make it more sustainable. Since we cannot predict in advance, we must do things while being aware that we can *never* know in advance what precise effects our actions will have. Luckily, as discussed earlier, evolution is path dependent, and future states have a characteristic structure. Because the system's components are acting and reacting to each other and adapting in the process, co-evolution of systems components provides us with robust patterns of future states that we can explore. Therefore, co-evolution needs to be understood.

### 3.4.4 Co-evolution

Apart from illustrating intractability, the example and notion of the path and the forest reacting to each other and to the person walking exemplifies the concept of co-evolution. The person, the path and the forest co-evolve. The notion of biological co-evolution was first introduced in Chapter 1. In this section its relevance to the thesis will be further specified.

**Evolution is co-evolution** A comprehensive overview of the co-evolution literature is beyond the scope of this work. Co-evolution in biological systems has been thoroughly described in the literature of biology (Futuyma, 1983; Jantzen, 1980; Thompson, 1994). Using the system perspective developed in Chapters 1 and 3, we see an ecosystem as a system consisting of many coexisting species which constantly interact through competition, predation, parasitism, commensalism, symbiosis, etc., situated in a particular environment. As the environment changes, and as randomness drives DNA mutation, species evolve. However, they do not evolve in isolation. A change in the fitness of a single species has a direct effect on all other species with which it shares the environment. For example, a predator that becomes more effective directly reduces the fitness of its prey, but also increases the fitness of the organisms that the prey eats. Basically, a change in a single species affects all species, changing their fitness and causing a pressure to evolve further.

**Co-evolution in  $\lambda$ -systems** Rammel et al. (2007a) provides an excellent review of the historical development of the term co-evolution in its application to human systems. It is clear that the term is well established and is widely used in ecological economics (Costanza et al., 1993; Jeffrey and McIntosh, 2006; Kallis, 2007; Kemp et al., 2007; McIntosh and Jeffrey, 2004; Rammel et al., 2007a; Sen, 1993; van den Bergh, 2007; Winder et al., 2005) and the transition management literature (Kemp and Rotmans, 2005; Kemp et al., 2006). The main limitation of the literature is that it mainly deals with the theoretical considerations and consequences of a co-evolutionary perspective, offering limited practical advice on how to use the concept.

**Definition** Co-evolution in  $\lambda$ -systems is defined by Rammel et al. (2007b) as:

At a general level, we conceive of co-evolution as dynamic interactions between two or more interdependent systems which account mutually for each other's development. In detail, co-evolution can be seen as the evolutionary process among two or more components/sub-systems/systems driven by reciprocal selective pressures and adaptations between these components/sub-systems/systems. Thus, a co-evolutionary system can be defined by the totality of all interacting components/subsystems. Moreover, co-evolutionary dynamics reflect different temporal, spatial and social scales, nested hierarchies, inevitable uncertainties, multidimensional interactions and contain emergent properties.

At the abstract level, all co-evolutionary interaction between components happens via a coupled fitness landscape.

**Coupled fitness landscape** Hordijk and Kauffman use the concept of coupled fitness landscapes to show that an evolutionary process never happens in isolation (Kauffman and Johnsen, 1991; Wilds et al., 2008). A fitness landscape is a description of the conceptual environment of

an individual or species, where every point in this solution space implies a certain fitness for an individual. For a graphic illustration of a changing fitness landscape, see Figure 3.4. The x and y axes represent the possible ranges of properties of two different species. The z axis represents the combined fitness landscape of the two species, with peaks and valleys for combinations of different properties of the two species. As the species co-evolve, they attempt to find the peaks, where they are both at their maximum fitness. However, as there are more than one species evolving, it is likely that the evolution of one species changes the fitness landscape of the other, and vice-versa. This is illustrated in Figure 3.4, where, going from left to right, the fitness landscape is deformed as species evolve and acquire new traits. The fitness landscape of both species is coupled and dynamic, with each evolutionary step of one species changes the fitness landscape of the other, and vice versa. This is the main cause of intractability of the evolutionary process. If the gazelle learns to run faster, the lion gets less to eat, until it develops a new strategy for dealing with fast gazelles, reducing the gazelles fitness again. The responses and counterresponses can not be predicted in advance, as they are driven by random mutation, and are influenced by the specific adaptation of the other species.

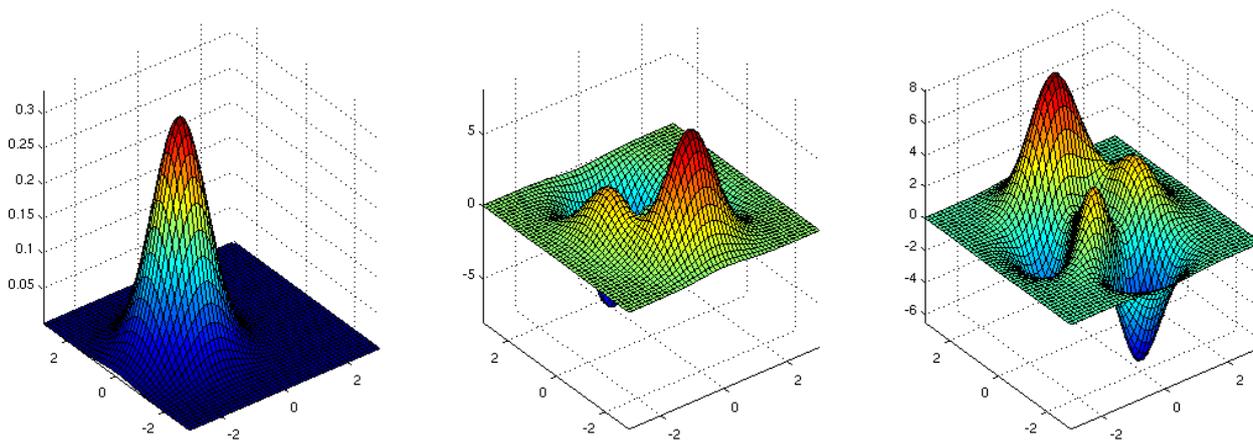


Figure 3.4: Deformation of a fitness landscape

**Next steps** As discussed in this section,  $\lambda$ -systems are evolving systems that experience co-evolution and have an intractable future path. They require multiple formalisms to be fully described and can be conceptualized to consist of three levels: agent, network and system. Given these elements, we are now ready to describe a generativist framework that will allow for the generation of evolutionary patterns of multiformal and complex  $\lambda$ -systems.

### 3.5 Complexity Framework

In this section, a generativist framework for understanding Complex Adaptive Systems is presented. It is constructed from the building blocks and insights presented in the previous sections. The framework consists of three conceptual levels: agent, network and system. Relevant properties and concepts at each level are presented. Observer and context dependency are discussed, as is the role of the randomness and order in the systems environment.

**Framework elements** Before we embark on describing the framework, we need to shortly recapitulate the conceptual building blocks presented earlier.

**Multiple formalisms** The complex nature of  $\lambda$ -systems requires that we use multiple formalisms to describe the system. A multiformal description starts with an agent description based on different knowledge domains (social, technical, etc.). A multiformal agent is capable of multiformal interactions, creating multidimensional networks (money, mass, etc.) and thus an emergent, multiformal system.

**Agent** The agent is the smallest element of the system. Its state and rules are multiformal and adaptive. See Appendix A.1 for details of the agent level.

**Network** The network level of a system describes the structure and aggregation of the interactions between agents. Evolution takes place at this level through changes in network structure. As the agent's description is multiformal, the network of agent interactions must necessarily be so, too. See Appendix A.2 for details of the network level.

**Emergent System** At the emergent system level, properties and behavior emerge from the structured interactions between agents. Again, as the agent and the resulting interaction network are multiformal, so is the overall emergent system. See Appendix A.3 for details of the system level.

**Evolution** Evolution of a system is understood as the process of change either at the agent level, where a change in agent properties and states causes a change in the interaction structure or a change in the network structure through the addition and removal of individual agents. The precise prediction of the overall system state as the result of changes at the agent or network level is impossible because of the intractable nature of the process. The change is never confined to a single element, as every change automatically causes change in other components as well, causing different system elements to co-evolve.

**Unified framework** To provide a foundation for a shared language and vocabulary, a holistic and generative framework is proposed. The goal of this framework is to offer a unifying perspective on complex systems that is domain-free and thereby usable as a shared knowledge interface between different knowledge domains. The aim is to provide a coherent, unified framework that shows that different system levels and properties are related to each other and that they interact in a generative fashion, creating a complex system from individual element interactions. It is only *one*, not *the* meta-model of a complex system.

**Framework description** Figure 3.5 presents the complexity framework. For a detailed description of each level, please refer to Appendix A. At the lowest conceptual level, an *agent* is defined as the smallest system component, converting inputs to outputs through its state and rules. Agents can interact because of their interfaces and protocol similarities. They are diverse in their states and rules and are adaptive.

In the middle conceptual level, the interactions between the agents create a network. This network has a certain topology. A change in the intensity of the connections between agents

creates network dynamics, and addition and removal of nodes and edges causes the network to evolve.

At the highest, or system, conceptual level, the entire emergent system can be seen as an entity with aggregate in- and outputs, that has an aggregate state and aggregate rules. At this level emergent properties become visible as a result of lower-level interactions. The system self-organizes, is robust and can be instable. At this level the entire system can be seen as as Agent, or the smallest system element of a larger system if the observer chooses to do so.

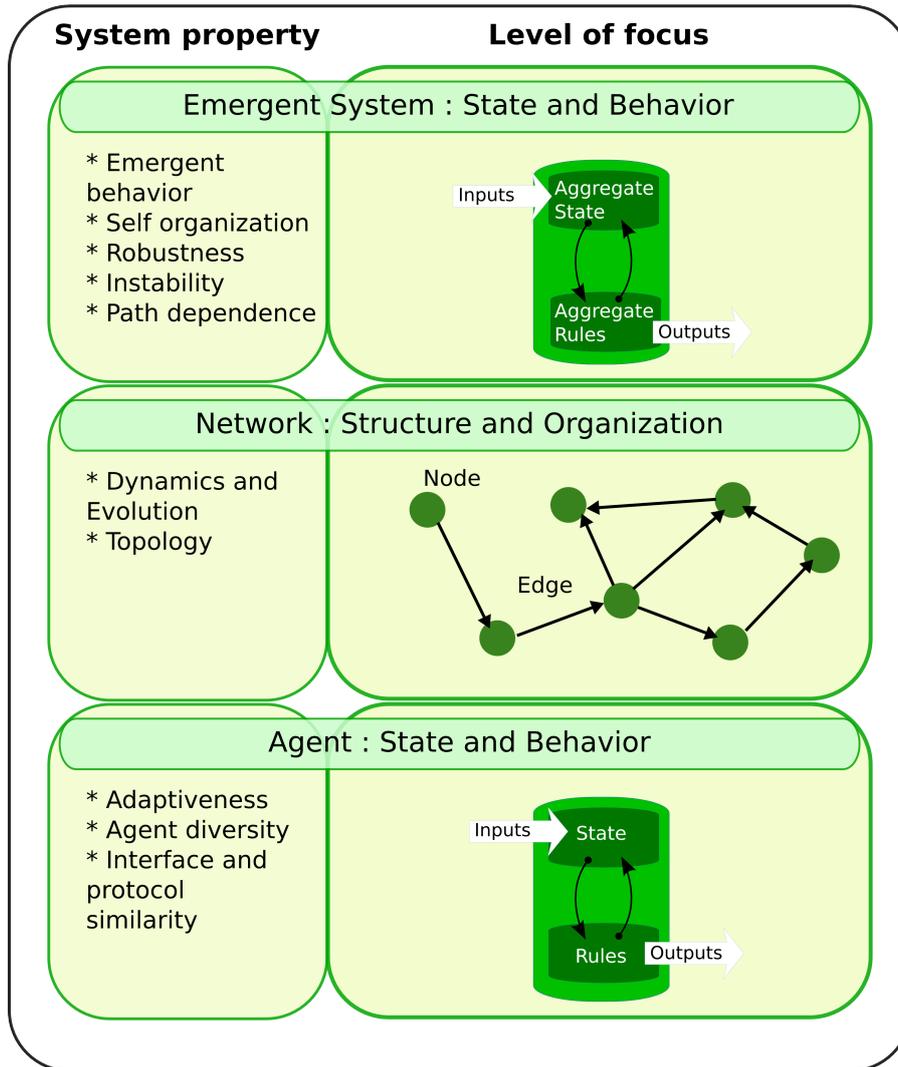
The framework is holistic, as it provides a perspective on a system from the smallest individual elements to the highest level of system aggregation. While it considers systems in its entirety, it is also reductionistic in a sense that it reduces systems to smaller elements if they are fully interrelated with other elements (Bar-Yam, 2003). It is generativist, as it understands systems as a result of continuous process of emergence across multiple levels, starting at the lowest level elements (Epstein, 1999).

Two aspects important for understanding the framework that were not discussed previously are: observer dependency and irreversibility.

**Observer dependency** When applying the framework, systems decomposition is dependent on the systems observer. The observer chooses the scale at which the system is observed, and thereby determines what is considered the smallest/largest element. For example, depending on the problem and system at hand a person can be considered an agent, whose interactions through the legal and social system lead to a the system level being an entire country. A different observer, in a different situation may choose to represent a country as an agent, and the entire planet as the emergent system. Furthermore, each observer chooses a certain perspective when interacting with a system. For example, a chair is a social, physical, chemical entity. Is the observer interested in the physical strength and the manufacturing process that produces it, or is the observer interested in the impact of uncomfortable chairs on educational performance of students? It depends on the observer which of these perspectives will be used to construct the conceptual model of the chair. Observer dependency is especially apparent when considering system behavior over time. An observer studying the plate tectonics of a region will not be able to make meaningful observations within a time scope of minutes or hours. The observer also chooses the systems aggregation level. By aggregation we mean that many agents that have the same or similar kind of behavior can be generalized into an aggregated, larger group. So the observer effectively determines what the smallest component is in the decomposition of the system. Aggregation is used to decrease the number of agents and to simplify the system. If one wants to compare the behavior of the soccer supporters of different countries, one does not consider all the individuals separately but aggregates them into homogeneous groups: i.e., the Dutch supporters and the French supporters.

**Irreversibility** Any change to any component or level of the system affects the whole system, as components co-evolve over time. Whenever an agent interaction is established, all other conceivable interactions are no longer possible. This causes an irreversibility and thus a path dependency in the system's overall behavior. This irreversibility can manifest itself in many ways. Systems could lose mass or energy, as is the case with physical systems (Prigogine, 1967). In social systems, a loss of information can often be observed. For example, groups can forget about specific knowledge, as in the case of using old technology. There are very few people in

## Observer



### Unpredictable dynamic environment

Figure 3.5: Complexity framework overview

the Western world able to make an effective bow and arrow or to start a fire without modern tools.

**Conclusion** This chapter presented the theoretical foundations of this thesis. It described the importance and use of formalisms, presented a comprehensive view of Complex Adaptive Systems and discussed aspects of evolution applicable to  $\lambda$ -systems. The chapter resulted in a holistic and generative framework for understanding Complex Adaptive Systems and their evolution. In the next chapter the framework will be used to place knowledge domains relevant for this work in a context of creating models of  $\lambda$ -system evolution.

All models are wrong, some are useful...

(Box, 1979)

## 4.1 Introduction

In the previous chapter, we presented a unified framework for understanding Complex Adaptive Systems and their evolution, for which we need multiple formalisms and thus knowledge from many different domains. Using this framework, we can start to explore the theoretical backgrounds needed for answering the second research question: “What are the content specifications of such models in terms of relevant formalisms/knowledge domains?” The answer to this question will dictate the way the model will be developed, and which knowledge domains will be necessary. This chapter is a search for a modelling approach, i.e, for the tools we need in building this model. If we dissect the main research question posed in Chapter 2: “How can we create a *model* for exploring the evolutionary patterns of  *$\lambda$ -systems evolution?*”, we can identify the knowledge domains necessary for answering it. These relevant parts and the associated knowledge domains are:

**$\lambda$ -systems evolution** The objects of study are industrial networks. A overview of the thinking on industrial clusters and how they change will be presented. Evolutionary theory was already presented in Chapter 3.

**model** There are many different ways to model Complex Adaptive Systems. The different approaches will be examined, and Agent-Based Modelling will be shown to be the method to use. Its background will be examined.

These two knowledge domains, Industrial Clusters and Complex Adaptive Systems Modelling, will be discussed. They are expected to yield the building blocks necessary for answering RQ1.

## 4.2 Industrial Clusters

In this section an overview is given of thinking about industrial clusters. A working definition of a cluster and a definition of a successful cluster are presented. Industrial ecology, a field concerned with the study of industrial networks, provides useful static metrics for possible emergent attractors. Individual firms and their processing plants are identified as the smallest system elements that cause clusters to emerge as a result of their interactions.

**Regional industrial networks** The domain focus of this work is the regional industrial clusters dominated by chemical, energy or metallurgical process industry and related infrastructure. European examples of such clusters are the Rotterdam-Rijnmond area and Groningen Seaports in the Netherlands, the German Ruhr Area, the Antwerp region in Belgium, Le Havre in France and Teeside in the United Kingdom. There are several ways one can characterize industrial regions (Vila et al., 2000). See Table 4.1.

Table 4.1: Characterization of the territorial grouping of companies. Adapted from (Vila et al., 2000)

Name	Geographical area	Type of company	Relations between companies
Cluster	Wide	Small, medium and large	Sectoral (vertical, horizontal and transverse)
Filiere	Diverse	Small, medium and large	Sectoral (vertical)
Industrial City	Small	Small, medium and large	Multi-sectoral
Industrial District	Small	Small and medium	Sectoral
Micro cluster	Small	Small, medium and large	Sectoral (vertical and horizontal)
Milieux	Small	Small and medium	Sectoral (innovation)

**Cluster definition** The most widely used definition of clusters in the literature, and the one used in this work, is given by Porter (2000):

Clusters are geographic concentrations of interconnected companies, specialized suppliers, service providers, firms in related industries, and associated institutions (e.g., universities, standards agencies, trade associations) in a particular field that compete but also cooperate.

In this work the term 'field' will be used loosely. Processing industry will be considered as a field, irregardless of whether it involves chemicals, energy or metals processing.

**Clusters as Complex Adaptive Systems** Porter (2000) further discusses a number of important issues for clusters. The importance of the 'quality' of local demand is emphasized. A high local demand gives the cluster a global competitive advantage. High local demand can be seen as a cluster asset that augments its global competitiveness: if the cluster can cater to strong local demand, it has a head start in catering competitively to global demand.

Another interesting point Porter raises is the ability of a cluster to develop itself when "several parts of a cluster change simultaneously" (Porter, 2000). In other words, clusters evolve (change their internal structure) through simultaneous changes in multiple components. They are adaptive. Clusters can also suffer from a conservative force: cluster culture or a sort of 'group think' may reinforce certain outdated behaviors, suppress new ideas and create rigidities that prevent the adoption of improvements. This constraining rigidity - or path dependency - is a further reason behind the complex nature of clusters.

Ribas et al. (2003) specifically discusses the clustering of the chemical industry. Chemical companies operating in a cluster showed a higher average production and also a higher Return On Sales. This better performance is explained by the availability of "a series of externalities and positive synergies that arise precisely from the fact that they share resources and capacities in the same geographical area". This can be seen as an example of clusters as self-organizing systems.

**Successful cluster** As already presented in Chapter 1, the goal of this thesis is to support the RDAs in making their clusters as successful - and thus as sustainable - as possible. Dijkema et al. (2005) defines a successful cluster as follows:

The prosperity and sustainability of any regional cluster depends on the activities located therein. (...) A region's prosperity and continuity is robust when the number of companies is sufficient and diverse, when the intensity of their activities and the volume of their transactions is stable or growing and when aging assets are timely replaced by new investments.

Given the definition above, and the fact that industrial clusters are  $\lambda$ -systems, a useful abstraction needs to be made in order to understand these clusters and help the involved RDAs achieve successful cluster evolution.

**Industrial Ecology** The body of knowledge needed for an analysis of socio-technical  $\lambda$ -systems has only recently emerged; the Industrial Ecology (IE) community had postulated useful paradigms on the required state of  $\lambda$ -systems (sustainability) and their preferred structure (networked ecosystems) (Frosch and Gallopoulos, 1992). Life Cycle Assessment (LCA), Material Flow Accounting (MFA) and Substance Flow Accounting (SFA) provide images of a system's physical performance. Industrial Ecology, operating from a first principles paradigm, attempts a broad aggregated systems description of the technosphere and mainly deals with large-scale energy and mass flows (see e.g. Allenby (1999)). A critical review of the IE body of knowledge has been done by Verhoef et al. (2004). Using the IE paradigm and Dijkema's definition, it is apparent that evolving  $\lambda$ -systems must have as an emergent characteristic a continuous increase in the total eco-efficiency of their energy and materials use, business continuity and socio-economic prosperity.

**Static and system level** The main drawback of traditional IE tools is that they are largely static and suffer from a lack of content in their representations of system structure and technology. In situations where the dynamics of systems are important, and where intervention rather than analysis is called for, IE approaches to date have been largely inadequate (Dijkema, 2004; Verhoef et al., 2004). While analysis methods on the present state of the physical part of  $\lambda$ -systems abound in the growing IE body of knowledge (Allenby, 1999; Graedel and Allenby, 1995), recipes for how to arrive at the preferred states of these Large-Scale Socio-Technical Systems appear to be largely lacking.

Furthermore, IE focuses on the analysis of systems at the emergent system level. This level is not suitable for a generativist approach, since it does not deal with the individual agents, but rather the emergent system structure. IE currently uses a primarily static, linear and monodisciplinary perspective in studying the technosphere. It lacks the ability to truly combine multiple formalisms. Therefore, IE is not a suitable formalism for modelling  $\lambda$ -system evolution. It does, however, offer a suitable approach for analyzing the emergent cluster structure, as will be demonstrated in Section 8.4.

**Next steps** In this section we have established that the Porterian view on clusters provides a suitable perspective, fitting in a generativist view of  $\lambda$ -systems. The traditional tool for studying industrial clusters, Industrial Ecology, has been shown to be inadequate as a generativist tool. It does provide a useful set of tools for the analysis of emergent industrial network behavior. The next step, then, is to examine ways to model the emergence of industrial clusters.

## 4.3 Modelling Complex Adaptive Systems

In this section several tools for modelling Complex Adaptive Systems, as well as their suitability for creating generativist models of  $\lambda$ -system evolutions, are examined. Criteria are formulated based on Ashby's Law of Requisite Variety, and it is demonstrated that only an Agent Based Model can fulfill this role. A definition of ABM is presented and typical-use cases are discussed. Firms, identified in the previous section as the relevant agents, are seen to consist of a technology and decision-making processes. Process engineering is used as the dominant view on technology and, together with business economics principles, it describes the agent's state. Rational decision theory is used as the relevant domain for describing the agent's decision-making process.

As discussed in Chapters 2 and 3,  $\lambda$ -systems are Complex Adaptive Systems. In order to respect their complex nature they must be suitably represented using the generativist complexity framework presented in Chapter 3. To help us understand  $\lambda$ -systems and their evolution, the framework needs to be operationalized through multiformal and generative models. In order to arrive at a set of requirements for a modelling tool capable of such a task, we must identify the special properties of CAS that the modelling tool must be able to capture and represent.

### 4.3.1 Requirement

**Requisite variety** In essence, there is only one requirement for a model of a Complex Adaptive Systems. The model needs to represent the system as well as possible. This is formally expressed by Ashby's Law of Requisite Variety (Ashby, 1968), which states:

The variety in the control system must be equal to or larger than the variety of the perturbations in order to achieve control.

Paraphrasing the definition above, a model system or controller can only model or control something to the extent that the model has sufficient internal variety to represent the real system. At the first glance, it may seem odd to use a top-down, control theory concept when describing a requirement for building models of  $\lambda$ -systems. However, a more commonly used formulation of Ashby's law states that "a model system can only model something to the extent that it has sufficient internal variety to represent it"<sup>1</sup>. If we take this comment outside the scope of control theory, we can see that it implies that if we are to build successful models of CAS, these models themselves need to be CAS.

**Essential properties** What is the minimum set of properties of CAS that need to be modelled? Which properties make a Complex Adaptive Systems a CAS? Summarizing Chapter 3, we can identify three main properties:

**Multiformal** CAS must be represented through multiple formalisms.

**Bottom-up and distributed** CAS consist of many heterogeneous but related components in dynamic interaction, generating emergent system behavior from the bottom up.

**Adaptive** CAS are by definition adaptive. In the case of  $\lambda$ -systems, we argued that they evolve over time.

Given the three identified properties, the modelling tool for describing a Complex Adaptive Systems, as with a  $\lambda$ -system, therefore must be able to contain multi-domain and multi-disciplinary knowledge, be generative and be adaptive. That is, it must exhibit adaptive, emergent behavior through interactions of adaptive heterogeneous low-level elements.

**Available tools** How can we meet the above criteria, and what modelling tools or components would be of use? Answering this question could lead to an examination of many, many fields in which modelling is used. There are even more modelling techniques available. Reviewing them all would be outside the scope of this thesis. The literature review was therefore limited to Complex Adaptive Systems modelling literature (Borshchev and Filippov, 2004; Epstein, 1999; Remondino, 2004; Shalizi, 2006) and fields dealing with  $\lambda$ -systems, such as Process System Engineering (Dijkema, 2004), Macroeconomics (Duchin, 2005; Leontief, 1998) and Operations Research (Bankes et al., 2002). System level analysis tools, such as Statistical Thermodynamics, were excluded, as were pattern recognition tools such as Neural Networks, as they are not generative. The following types of modelling tools were found to be possible candidates:

**Computable General Equilibrium** CGE models are static equilibrium models (Jones, 1965; Leontief, 1998) mainly used in neoclassical economics, based on linear equations. They are used to describe overall states of  $\lambda$ -systems, such as markets and industrial regions (Duchin, 2005). Their advantage is that they have a relatively elegant and simple mathematical formulation and allow the possibility for analytical solutions. However, CGE models assume a static system structure and the existence of static equilibria.

---

<sup>1</sup><http://pespmc1.vub.ac.be/REQVAR.HTML>

**Dynamic Systems** DS models are used to model physical systems. They are based on mathematical models consisting of state variables and algebraic differential equations of various forms over these variables (Rosenberg and Karnopp, 1983). DS models are based on first principles, rooted in laws of nature and have a continuous representation of time. This gives them great accuracy in describing system behavior. They can display chaotic behavior and forms of adaptation (Strogatz and Henry, 2000). Due to their differential equation basis, DS are unable to encode multiple formalisms and thus assume a static system structure.

**System Dynamics** SD models involve a top-down approach. They are tools based on differential equations, describing the interaction between variables as stocks and flows (Forrester, 1958; Forrester and Wright, 1961). They assume a fixed structure between variables. SD models can display chaotic and even adaptive behavior. However, they necessarily assume a static internal system structure.

**Discrete Event Simulation** DES combines a bottom-up and top-down approach. DES conceptualizes the world as consisting of many discrete entities that change their states in response to some external event, usually ticks of a clock. Entities in DES models are passive objects that represent people, parts, documents, tasks, messages, etc. Give certain system inputs or states of other objects, these entities change states, e.g. people arrive and leave, documents get created, approved etc. (Boer et al., 2002; Boyson et al., 2003; Corsi et al., 2006; Gordon, 1978).

**Agent-Based Modelling** ABM uses a bottom-up perspective. Individual agents act and react to each other, following their internal rules. The overall system behavior is emergent. Agents are software entities, described through computer algorithms (Jennings, 2000b; Shalizi, 2006). ABMs conceptualize the world as resulting from the interactions of many different entities. The algorithmic nature of agents means that they can encode many different formalisms. Analytical solutions to agent interactions, however, are often impossible.

**Most suitable tool** Of the presented tools, ABM is the only one that can be adaptive, generative and multiformal. Furthermore, ABMs are modular in nature, which allows for different formalisms to be encoded (see Appendix A.1). It is the only tool presented that satisfies Ashby's requirement. Furthermore, in the words of Borshchev and Filippov (2004):

Agent Based approach is more general and powerful <sup>2</sup> because it enables capturing the capture of more complex structures and dynamics. The other important advantage is that it provides for construction of models in the absence of the knowledge about the global interdependencies: you may know nothing or very little about how things affect each other at the aggregate level, or what the global sequence of operations is, etc., but if you have some perception of how the individual participants of the process behave, you can construct the AB model and then obtain the global behavior.

---

<sup>2</sup>than System Dynamics, Dynamic Systems or Discrete Event Simulation

### 4.3.2 Agent-Based Modelling

Thus, the main modelling tool used throughout this thesis shall be Agent-Based Modelling (ABM). In this section the theoretical background of ABM will be discussed. Practical details describing the creation of ABMs are described in Chapters 6, 7 and 8.

**Definition** Agent-based models take agents (components) and their interactions as central modelling focus points. Stuart Kauffman has been quoted to say that “an agent is a thing which does things to things” (Shalizi, 2006).

Furthermore, Shalizi (2006) states that:

An agent is a persistent thing which has some state we find worth representing, and which interacts with other agents, mutually modifying each other’s states. The components of an agent-based model are a collection of agents and their states, the rules governing the interactions of the agents and the environment within which they live.

Another perspective is provided by Tesfatsion (2007):

In the real world, all calculations have real cost consequences because they must be carried out by some entity actually residing in the world. ACE<sup>3</sup> modelling forces the modeller to respect this constraint. An ACE model is essentially a collection of algorithms (procedures) that have been encapsulated into the methods of software entities called ‘agents’. Algorithms encapsulated into the methods of a particular agent can only be implemented using the particular information, reasoning tools, time and physical resources available to that agent. This encapsulation into agents is done in an attempt to achieve a more transparent and realistic representation of real world systems involving multiple distributed entities with limited information and computational capabilities.

**Computer program** In essence, an agent is an computer program. Jennings (2000b) defines the agent as “ an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives”. Agents are reactive, proactive, autonomous and social software entities. Agents are:

1. Clearly identifiable problem-solving entities with well-defined boundaries and interfaces;
2. Situated (embedded) in a particular environment - they receive inputs related to the state of their environment through sensors, and they act on the environment through effectors;
3. Designed to fulfill a purpose - they have particular objectives (goals);
4. Autonomous - they have control both over their internal state and over their own behavior; and

---

<sup>3</sup>Agent-Based Computational Economics; essentially ABM with agents containing economic decision models

5. Capable of exhibiting flexible problem-solving behavior in pursuit of their design objectives - they need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and proactive (able to act in anticipation of future goals).

(Adapted from [Jennings \(2000b\)](#))

**Why are Agent not just Objects?** Shalizi <sup>4</sup> states that :

While object-oriented programming techniques can be used to design and build software agent systems, the technologies are fundamentally different. Software objects are encapsulated (and usually named) pieces of software code. Software agents are software objects with, additionally, some degree of control over their own state and their own execution. Thus, software objects are fixed, always execute when invoked, always execute as predicted, and have static relationships with one another. Software agents are dynamic, are requested (not invoked), may not necessarily execute when requested, may not execute as predicted, and may not have fixed relationships with one another.

**ABM vs MAS** When studying the literature dealing with agents and the models based on them, we encounter two related but distinct types of models spread. In addition to Agent Based Model, one often encounters the concept of Multi-Agent Systems (MAS). Before proceeding to examine our agent, it is important to differentiate these two, as they are often, but incorrectly, used interchangeably.

**MAS** *How can I make a ...?* Multi-Agent Systems are employed to engineer a certain desired emergence from a system. MAS acknowledge that a given control problem (e.g. traffic control, agenda synchronization, etc.) must be solved using discrete, parallel, autonomous components, namely agents. Work needs to be done to ensure that all types of conflict resolution are performed in order to achieve the desired outcome, (i.e., no traffic jams, and non-conflicting appointments). The following are typical examples of MAS in the literature: [Hadeli et al. \(2004\)](#) describes a MAS for process design using component collaboration and local information, [Lee \(2003\)](#) presents the design of an advanced e-commerce agent, [Negenborn et al. \(2006\)](#) presents a MAS predictive control system for transportation networks, and [Lanzola et al. \(1999\)](#) focuses on the design of cooperative agents in a medical MAS system.

**ABM** *What happens when ...?* Agent-Based Models are constructed to discover possible emergent properties from a bottom-up perspective. ABM acknowledges that reality consists of many components acting in parallel. ABM describes these entities and lets them interact in parallel, observing all possible interaction models. There is no desired state or task that needs to be achieved, only an exploration of the system's possible states. Some typical examples of ABMs are: a model of farmers' adaptations to climate change by [Schneider et al. \(2000\)](#), a model of the co-evolution of autocatalytic economic production and economic firms by [Padgett et al. \(2003\)](#) and the model of an abstract economy by [Kauffman \(2008\)](#). Models presented in this thesis also fall squarely into the ABM category.

---

<sup>4</sup><http://cscs.umich.edu/~crshalizi/notebooks/agent-based-modeling.html>

**Use of ABM** Now that we have clearly delineated what ABMs are, we need to explore how they are used. [Janssen and Ostrom \(2006\)](#) argue that modellers generally use ABM in three situations. These are: high quality data being available, role-playing situations and laboratory experiments. To these three situations, we add two more use cases: case study exploration and future mapping. These use situations are elaborated below.

**High quality data** The first situation for using ABM is when high quality empirical data are available. This allows the creation of stylized facts and the use of ABM to examine which micro behavior/processes emerges from those stylized facts. Example of such uses are: the models of market bubbles and crashes by [Giardina and Bouchaud \(2003\)](#) and [Challet et al. \(2001\)](#).

**Role playing** The second situation is in role-playing games. We can examine the context in which the agents/players find themselves and their potential interactions. Examples of such uses are: the models of negotiation by [Barreteau \(2003\)](#) and [Etienne \(2003\)](#); models studying identity, by [Guyot et al. \(2007\)](#); land use patterns, by [\(Castella et al., 2007\)](#); resource management, by [Guyot and Honiden \(2006\)](#); and food security and climate change, by [Bharwani et al. \(2005\)](#).

**Laboratory experiments** The third situation for ABM use is closely related to role-playing games. In cases where laboratory experiments are possible, human decision making in certain situations can be examined and compared to a theoretical model. Examples are: the work of [Duffy \(2001\)](#) in modelling the use of financial markets; and the power market design by [Koesrindartoto et al. \(2005\)](#).

**Case studies** The fourth situation involves specific case study analysis. Multiple sources of incomplete information are available. We can construct a theory of the situation using the agents and examine which fits the incomplete data best. This will be the approach used in this thesis. Examples are: a model of computer network security, by [Gorodetski et al. \(2001\)](#); the model of CO<sub>2</sub> trading by power generation companies, by [Chappin \(2006\)](#); and a per household model of distributed power generation, by [Houwing and Bouwmans \(2006\)](#).

**State mapping** Finally, ABM can be used as a potential future state mapping tool to examine where a system could go, given its current state and rules, as was discussed in Chapter 1. In this case, the model becomes an exploratory tool for examining possible system attractors. Examples are the two different co-evolutionary models of artificial economies by [Padgett et al. \(2003\)](#) and [Kauffman \(2008\)](#).

In this thesis, ABM will be used as a future mapping tool, based on case studies with incomplete information. We will construct theoretical agents based on actors in the case studies and use the agents' interactions to play out possible future development scenarios. In order to do that, we need to further operationalize the agent description, which is currently purely theoretical.

## 4.4 Literature Review

This section presents a review of literature on modelling  $\lambda$ -systems and their evolution. It is aimed at identifying prior modelling efforts and, in the case that there are none, at identifying whether the necessary components exist. The review also attempts to find out whether the relevant knowledge domains have previously been explicitly brought together.

In order to establish which, if any, prior work has been performed in the generative, multi-formal and evolutionary modelling of  $\lambda$ -systems, a broad literature review was undertaken. The review consisted of an ISI Web Of Science search using the terms 'agent model\* soc\* tech\*', attempting to find all mentions of *agent-based models of socio-technical systems*. The search query assumed a logical 'and' between the terms, and the \* matched any extension of the root word. The search was performed across the entire ISI WOS database. At the time of writing <sup>5</sup>, the search returned 260 publications. A first quick scan revealed a widely varying relevance of papers. In order to increase the relevance, the following fields were excluded: all medical applications, all robotics papers and all computer security/computation resource sharing. Such papers were considered to be outside the relevant scope. Furthermore, publications dealing with intelligent agents, MAS, etc., that were not simulations but designs of systems that perform some task, were not considered. The focus was on simulations of systems using agents. The remaining 171 papers were considered for this review. The review used the scoring and overview structure presented below:

**Title** Obviously.

**Domain** What is the scientific discipline/domain?

**CAS** Does the paper use the concepts of Complex Adaptive Systems explicitly?

**ABM** Does the paper implement an ABM?

**LSTS** Does the paper consider  $\lambda$ -systems?

**Evolution** Does the paper consider a modular and/or evolutionary approach?

**Social process** Does the paper present an explicit social process for collecting knowledge?

The result of the review is presented in Appendix G.

**Existing models** The goal of this literature review was to establish whether previous generative models of  $\lambda$ -system evolution are available. Six papers (or 4 %) were found to deal with topics related to  $\lambda$ -system evolution.

[Pahl-Wostl \(2005\)](#) argues that in the management of evolving  $\lambda$ -systems, distributed decision-making is better than centralized. The author explicitly discusses the involvement of actors in ABM modelling of  $\lambda$ -system evolution ([Pahl-Wostl and Hare, 2004](#); [Panebianco and Pahl-Wostl, 2006](#)). The focus of models is on drinking- and waste water management, and are as such not

---

<sup>5</sup>October 29, 2007

directly relevant to industrial clusters. The stakeholder involvement notions will, however, be applied in the System Decomposition Method presented in Chapter 6.5.

Kun et al. (2007) present an agent-based model of diffusion of technological developments in socio-economic systems. It is a cellular automata model, using stochastic models to describe the diffusion of technology. The main mechanism is described by the authors as follows: “Agents of the model can represent individuals or firms which use different level technologies to collaborate with each other. Costs arise due to the incompatibility of technological levels and to different technological providers. Agents can reduce their costs by adopting the technologies and providers of their interacting partners” (Kun et al., 2007). Path-dependent clusters of “high technology” are observed to emerge. The statistical and abstract descriptions of the technology, in addition to the lack of explicit economic interaction, make this approach unsuitable for our goals.

A similar approach is used by Zhang (2003) to create a stochastic cellular automata model of a Schumpeterian industrial districts. Authors include social effects through the role of entrepreneurs, in addition to the role of spatial proximity emphasized by Schumpeter (1983). The approach is unsuitable for our use as it is a stochastic description of agents and has an extremely simplified model of technology.

Zhang et al. (2005) present a theoretical game model of multi-layer infrastructure networks. The authors conceptualize  $\lambda$ -systems in a manner compatible with the socio-technical view presented in this thesis. The model describes a network with a static structure, where the flow through the network is optimized by agents playing an inverse Stackelberg game (Basar and Olsder, 1980; von Stackelberg, 1952) with an assumed central coordination authority. The presented approach, even though it conceptualizes the problem in a useful fashion, seeks global optima. It does not consider evolutionary change and is not generative.

Hadeli et al. (2004) present a model for processing plant design using unit operations collaboration and local information. It is an interesting concept, creating self-organizing emergent processing plants from unit operation interactions. This conceptualization chooses agents below the firm/processing plant level, which are below the smallest level considered in this thesis. This approach, while conceptually useful, focuses on the technical aspects in great detail and is not suitable for our purposes.

Padgett et al. (2003) present an abstract model of an economy, based on an autocatalytic co-evolution between economic production and firms. While the approach is not directly applicable to modelling multiformal and network aspects of  $\lambda$ -system evolution, it is an interesting alternative approach to describing system evolution.

The most relevant model found in the literature is Boero et al. (2004)’s model of micro behavioral attitudes and macro technological adaptation in industrial districts. The model presents an abstract supply chain with explicit agent reasoning styles. Agents are boundedly rational, with limited access to information, limited memory and limited information processing abilities. This conceptualization is useful and will be used in this work. The most important limitation of Boero et al. (2004)’s model is the lack of an explicit technology description, and with it the lack of a physical and thus multiformal reality. Furthermore, there is no focus on network structure and its evolution, but on the ability of the cluster to change its technology content in order to supply a certain demand.

**Other relevant publications** Albino et al. (2006) present a model of innovation in industrial districts. The model deals with the social and learning aspects of innovations in firms. It

provides a useful model for agent adaptation. The main difficulty with this approach is its very abstract description of technology. Operationalizing the adaptation model to an explicit technology description is a challenge.

[Andrews et al. \(2005\)](#) present a model of a small firm, which is interesting because it models the emergent behavior of a firm from employees' interactions. Based on a sociological, psychological and management perspective, the paper examines the company's performance depending on different worker properties, such as making errors. While the level of aggregation is too low for our purposes, it is an interesting generative model that could be used to examine firm behavior.

[Garcia-Flores and Wang \(2003\)](#) and [Aldea et al. \(2004\)](#) suggest that agents need not only a communication language and a standard interface in order to interact, but also a shared model of the world. This is a strong call towards a systematic conceptualization of knowledge necessary for model development. These insights will be used in the design of the System Decomposition Method in Chapter 6.5.

[Keirstead \(2006\)](#) calls for an integrated, flexible ABM framework, using a shared common language to investigate policy and behavioral effects on domestic energy consumption. The author further argues that in order to make good models we need engineering, economics, psychology (decision-making) and sociological knowledge. We wholeheartedly agree, and this thesis can be seen as an attempt to answer this call.

A number of publications<sup>6</sup> are reviews, model meta analyses and discussions on tools. They all basically ask the same question: "Is ABM science?" and the answer is a universal "yes". [Janssen and Ostrom \(2006\)](#) discuss how to use empirical data in an ABM.

[Ottens et al. \(2004\)](#) find that traditional systems engineering process is problematic when applied to the design of (socio-technical) systems, and call for a new approach, where in addition to physical and social elements and their functional relationships, normative and intentional relationships should be introduced. This is a clear call for multiformalism in modelling  $\lambda$ -systems.

**Elements present, integration missing** On the basis of the literature review in Chapters 3 and 4 we can conclude that the theoretical components necessary for the creation of a multiformal, generative ABM of  $\lambda$ -system evolution are already present. Furthermore, we can conclude that there are no publications that explicitly combine all of the necessary building blocks into a single and coherent, generativist and multiformal approach. What is missing is a systematic integration of these knowledge domains into an operational and practical system that engages interaction between fields in order to create such models. This thesis aims to make that connection.

## 4.5 Operationalizing the Agent

In this section an operational description of the agent is presented. The theoretical backgrounds of the two main formalisms used to describe the agent's state and rules, process engineering and corporate finance, are presented.

---

<sup>6</sup>([Berger and Schreinemachers, 2006](#); [Bryson et al., 2007](#); [Sansores and Pavon, 2006](#); [Tsfatsion, 2007](#))

**Recapitulation** In Chapters 1 and 2 we argued that  $\lambda$ -systems are complex systems whose evolution needs to be understood in order to steer them towards a more sustainable future state. In this chapter, we have explored industrial clusters and established that Agent Based Models are a useful approach to model their evolution. In a generativist perspective, we need to define the smallest system element and the types of interactions of which that element is capable. In this thesis, a firm will be abstracted as an agent.

**Firm as an agent** The smallest element in a  $\lambda$ -system such as an industrial cluster is a firm and its production facilities. A firm and its technology is a multiformal beast. The employees and management of the firm form a social network that uses economic and business reasoning to decide on suppliers, product pricing, (dis)investments, etc. It has to deal with changing markets and faces stiff competition to survive. The physical assets operated by the firm have to meet performance targets, have a designed capacity and a certain environmental impact. These aspects involve many different formalisms in their description. On top of that, the firm's technology changes over time, as does the social network that makes decisions. This diversity of formalisms and their adaptive natures cause the regional industrial clusters that consist of such firms to be complex.

**Adaptive agent** In order to operationalize a complex entity such as a firm, a multiformal agent must be created. By definition, the description of such an agent is never complete. While a certain number of formalisms and a certain level of detail may be sufficient to answer a particular question about possible future states, it is likely that the requirement will increase over time, as more and more sophisticated descriptions will be called for in the future. We therefore must be prepared to conceptualize an agent that is not only adaptive in state and behavior, but also has the ability to incorporate new formalisms, new states and new rules over time.

**Agent state and rules** We will start the agent description with two basic formalisms that will form the theoretical base for agent state and rules. The agent's state is conceptualized to consist of technical and economic parameters. The behavioral rules are defined by the operational aspects of technical installations and by business management logic. On this dualistic basis, a multiformal and evolving agent can be created.

### 4.5.1 Agent State

The agent state is a collection of variables that we find useful to represent. These can be simple numerical values or complex concepts. As the agent is a firm owning a production facility, the agent's state is conceived to consist of technical and economic state variables. This dichotomy makes the agent multiformal. For further discussion on the agent state, please refer to Appendix A.1. The technical domain is conceptualized using the input/output perspective from the field of PSE. The business and economic aspects are based in the field of corporate finance.

**Input/output perspective** The discipline concerned with a physical description of technology in process engineering is Process Systems Engineering (PSE). PSE approaches systems from a mass and energy flow perspective. Every entity is conceived as a thing that converts

mass and energy from one state to another. Since mass and energy are conservative quantities, systems behavior can be simply defined in equation 4.1:

$$\frac{d \text{mass}}{dt} = \phi_{\text{mass}_{in}} - \phi_{\text{mass}_{out}} + \phi_{\text{mass}_{production}} \quad (4.1)$$

$\phi$  denotes a quantity per time flow (e.g. kg/s). Obviously, this equation also holds for energy. By specifying the mass and energy balance of a physical component, all relevant quantities are known. This serves as a basic, first principles physical model of technical network components. Processing plants are nodes that transform mass and energy from one state into another, and edges are the mass/energy flows between the nodes. The agent's state then consists of a description of the in and out flows of energy and mass. Any other parameter relevant for describing the technology is also a part of the state, such as the operational scale of the technology, alternative operational states, start up and shutdown times, etc. Due to the modular nature of agents, other more specific technological aspects can be added when necessary.

**Corporate finance** In addition to technological aspects, the agent's state consists of economic variables. The field of Corporate Finance (Brealey et al., 2004; Damodaran, 1997) offers us insights into the relevant state variables. Assets, debt, operational costs, investment costs, etc. are examples of the economic state of an agent. These variables can be seen as the moving parts of the economic machine of the agent. The economic variables, while based in a different formalism, are none the less interdependent with the technology variables. The running cost of a processing plant is dependent on the operation scale of the installation. Just like mass and energy, money flows in and out of the firm and hopefully accumulates<sup>7</sup>. These multiple formalisms are distinct yet interrelated parts of the agent's identity.

## 4.5.2 Agent Rules

An agent's rules describe the way inputs and the agent's state are transformed into new agent states or outputs. The second part of the agent's multiformalism is captured in the rules. Each domain adds a new behavioral rule to the agent and can be related to an existing rule from another domain. Initially, an agent has technical and economic behavioral rules.

**Technical** The technical behavioral rules are based on the law of conservation of mass and engineering principles and follow a gray-box approach. For example, the total mass input must be equal to the total mass output, and a multiplicity of flows is allowed. PSE domain knowledge<sup>8</sup> describes how different types of mass and energy flowing into the agent are converted into other types of mass and energy flowing out. Furthermore, PSE deals with the design and dynamics of processing units, for example when being scaled up or down, as most chemical plants have a minimum operating load and cannot be freely downscaled.

**Rational decision-making** Decision-making rules of single individuals are notoriously difficult to encode into models. Nonetheless, when aggregated into firms, the emergent collective

---

<sup>7</sup>Obviously, there is no law of conservation of money.

<sup>8</sup>For example: (Coulson and Richardson, 1999; Elvers et al., 1988; Green and Maloney, 1984; Hobson and Pohl, 1984)

behavior can be described as rational. This rational decision-making paradigm is strongly related to economics. Its main premise is that actors have needs or preferences on the basis of which they choose between alternatives (Frederickson and Smith, 2003; Shepsle and Bonchek, 1997). For an agent to be 'rational', it must optimize whenever a decision is needed. For example, when evaluating alternative contracts, the agent will select the cheapest possible contract, and when setting a price for its products, it will ask for the highest possible price. It seems then that agents are perfectly rational. However, this is not completely true.

**Weak bounded rationality** Bounded rationality is defined as follows: “Boundedly rational agents experience limits in formulating and solving complex problems and in processing (receiving, storing, retrieving, transmitting)” (Simon, 1997, 1982; Williamson, 1981). While agents often have the computational ability to make the optimal choice, there are several limiting factors that make them partially bounded instead of perfectly rational. First, agents do not have perfect information of the entire system. Agents act and react based on the local, limited information available to them. Furthermore, agents can only optimize on a preprogrammed, limited set of criteria. So the optimum the agents find is always a local one. They are not limited by their computational ability, like humans are, but by the number and diversity of concepts that can be encoded in their programming.

## 4.6 Next Steps

This chapter has discussed the modelling foundations needed to create models of  $\lambda$ -system evolution. A theoretical background on industrial clusters, the Agent Based Modelling approach, and the basic background of the agent were presented.

Yet this is not sufficient to be able create models that comply with Ashby's requirement of requisite variety. If we are to model an evolving system, the method for building such models, not just the model itself, must be evolutionary. That is the topic of the next chapter: how to apply evolutionary thinking to the design method for building models of  $\lambda$ -systems, and how to decide whether we are doing it correctly.



## CHAPTER 5

# MODELLING METHOD AND REQUIREMENTS

One general law, leading to the advancement of all organic beings, namely, multiply, vary, let the strongest live and the weakest die.

(Darwin, 1985)

### 5.1 Introduction

This chapter argues that the method of developing models of  $\lambda$ -system evolution itself needs to be evolutionary. Two core concepts of the method, collaborative research and the co-evolutionary approach, are discussed. The co-evolutionary method is conceived as a continuous interaction between the technical design of a simulation engine, the design of a social process for knowledge encoding, the actual knowledge encoded and the number of facts collected. By constantly improving one or more of these aspects, new questions and insights are created, and a continually improving method for modeling and understanding  $\lambda$ -system evolution emerges. A number of guiding principles for shaping of the co-evolutionary method are discussed, leading to a formulation of requirements for the co-evolutionary modelling method. Finally, the practical steps of the method itself are presented, and the issues surrounding the verification and validation of the co-evolutionary method are discussed. As a matter of clarification, when talking about the modeling process, we mean the act of creating models. By modeling method we understand the stepwise “recipe”, or activities that describe how the modelling process is performed.

**Method for modelling** In this chapter our aim is to answer RQ3: “What are the specifications for a method that would create such models?” The goal of this thesis is to create models that will allow us to explore *what if* questions about possible future states of  $\lambda$ -systems. These models will help the decision makers to get a hold of the levers and steering wheels of a synthetic evolving system that exhibits behavioral dynamics similar to real world systems. Using these models, decision makers can then explore possible futures and analyze the consequences of different possible actions. To answer these questions, nontrivial models are needed, and such models are equally nontrivial to make. How can we build models that are sufficiently complex to perform this task?

**From simple to complex** Gall (2002) provides us with a somewhat tongue-in-cheek but nonetheless important observation on complex systems:

A complex system that works is invariably found to have evolved from a simple system that worked. The inverse proposition also appears to be true: A complex system designed from scratch never works and cannot be made to work. You have to start over, beginning with a working simple system.

We can examine the above statement in the light of Ashby's law (see section 4.3.1) applied not only to the models themselves, but to the method of creating those models. Paraphrasing, if the result of a method is to be a successful evolving model of a Complex Adaptive System, the method of creating it must also be complex and adaptive. We are not only making generative models; we also need to have a generative method for creating these generative models. This insight leads us to the central paradigm of the modelling method: *If we are to model a Complex Adaptive System, not only must our model be a Complex Adaptive System, but the method for creating the models must itself be a Complex Adaptive System. We must start with a simple method for generating simple models, and evolve so that over time a complex method will generate complex models.*

This notion of the method for evolving models from simple to complex enough to sufficiently describe a real world  $\lambda$ -system is central to the modelling method presented in this chapter.

**Readers guide** In the next section (5.2.1) the key concepts of collaborative research and evolutionary modelling are addressed. Subsequently, the guiding principles are elaborated (5.3) and requirements are developed (5.4). The chapter concludes with a description of a method that meets the requirements. Abiding with the most important insight and requirement from evolutionary thinking, the presented method is inevitably an initial, simple version.

## 5.2 Core Concepts

In this section the core concepts of the co-evolutionary method, collaborative research and the co-evolutionary approach, are presented. In collaborative research the importance of social processes, use of multiple formalisms, recorded history and openness of tools is discussed. The co-evolutionary approach will identify the co-evolving elements and the environment of the modelling process. These two core concepts will deliver the guiding principles of the co-evolutionary method for creating models of  $\lambda$ -system evolution.

### 5.2.1 Collaborative Modelling

This section will present several issues regarding collaborative research and group model building. The discussion on knowledge engineering in Chapter 4 made it clear that collaborative research is unavoidable when creating multiformal models of  $\lambda$ -system evolution. Pahl-Wostl (2007) furthermore argues that integrated, participatory and adaptive social processes are needed when trying to understand changing  $\lambda$ -systems.

**Problems and advantages** There has to date been a lack of publications on collaborative modelling using Agent-Based Modelling. Collaborative modelling, however, in the form of Group Model Building, has been researched extensively in the System Dynamics community ([Andersen and Richardson, 1997](#); [Vennix, 1999](#)). The main problems identified with group modelling are that modellers have to play many different and sometimes contradictory roles during the process ([Richardson and Andersen, 1995](#)), the modelling process is lengthy and tedious for the stakeholders ([Vennix et al., 1993](#)) and that there is a lack of systematic approaches to group modelling - it is more art than science ([Andersen et al., 1997](#); [Rouwette et al., 2002](#)).

Despite the problems identified, we argue that group modelling is worth the effort. The main advantages of collaborative model development as identified by [Berger et al. \(2007\)](#) are that it generates trust in the model, since participants can identify themselves with model components. It also promotes ownership, since the stakeholders are involved in specifying the model. Furthermore it improves model quality - both in terms of the quality of conceptual knowledge encoded and in terms of quality of data used in the model and it ensures that relevant questions and problems are addressed through constant discussion. Finally, it is useful as it teaches the users about the model.

Upon inspection of these advantages and disadvantages, it is obvious the success of any collaborative process not only depends on the participants but also on the design of the collaborative process.

**Guidelines** [Keirstead \(2006\)](#) in favor of making collaborative multidisciplinary models as “there is a need to work within the established disciplines, since most research assessment agencies are still monodisciplinary.” Furthermore the author argues that “Knowledge must be modular.” and “as many different types of modelling tools as possible should be used. Such models must, according to [Keirstead \(2006\)](#) be :

- theoretical and practical;
- allow aligning of disciplines in a transparent manner
- allow for both structure and process description
- be both top-down and bottom-up.

The author concludes that “an agent based vocabulary is a practical and accessible way for integrating the various aspects“ and that ABM would be an appropriate methodology to achieve the above.

The guidelines presented by the author, while valuable, are not specific and operational. [Westerberg et al. \(1997\)](#) discuss in their seminal paper the practical issues related to the design of a collaborative modelling process. Their insights can be summarized as:

- We need a record of the group’s history.
- We need diversity in formalisms.
- We need to control our design tools.
- Learning can be autocatalytic.

These insights are elaborated in the following paragraphs.

**Recorded history** As mentioned in the introduction of this chapter, we need to build complex models from successful simple ones. Chapter 3 identified path dependency as a major aspect of evolving systems. In support of this, [Westerberg et al. \(1997\)](#) argue that most companies or teams only use a small fraction of their intellectual capital when involved in a collaborative process and fail at systematic learning. In the words of [Westerberg et al. \(1997\)](#):

If a company can capture its history in a useful form, it can form the basis of learning and reuse of its design process ... Without a model of how decisions impact the effectiveness of a proposed design process, we can only use our intuition to test the likely impact of alternative processes on these goals.

The authors' observation can be understood to be of the lack of understanding of path dependency and the lack of recognition of its importance to learning in groups. What you know now was determined by what you knew in the past, and it will determine what you can learn in the future. A learning system must be aware of this.

**Multiple formalisms** [Westerberg et al. \(1997\)](#) further demonstrate that approaching a design problem from many different angles, by different teams at the same time, achieves better results than a single search strategy.

Owing to the diversity of backgrounds typically present, [Bucciarelli \(1984\)](#) states that a design team's first activities are to negotiate the vocabulary its members will use, how it will make decisions, what decisions it will make and has made, etc.

The authors also call for the use of structured data and knowledge collection systems. While not explicitly mentioning them <sup>1</sup>, the authors clearly refer to the use of ontologies.

**Open source** Since we are designing an evolving method, it is clear that we must retain the ability to modify any part of the tool chain we use. [Westerberg et al. \(1997\)](#) emphasize the importance of open source software:

... [Its] design is an evolution of the artifact description, of the information being gathered and organized, and of the design process itself; designers themselves should be enfranchised with the power to carry out this evolution without the attendant delays required when they can not and/or are not allowed to modify their design support software.

Also acknowledged is the importance of recording the attempts that end in failure. These are often them most valuable in terms of learning. Mentioned is the need for both standard interfaces between design components and a degree of design anarchy within those components. Just as in any evolutionary process, evolution of the model creation method requires both randomness and order if it is to function correctly.

---

<sup>1</sup>The paper was written in 1997, well before the concept of ontologies was widely known outside the Artificial Intelligence community.

**Autocatalytic evolutionary process** Finally, [Westerberg et al. \(1997\)](#) point out that the collaborative development of models, tools and approaches creates a recursive process of improvement and an increase of knowledge. We can consider this comment in the light of Complex Adaptive Systems. Collaborative, shared learning creates a knowledge feedback loop. The process is basically autocatalytic, which means that the increase of the group’s knowledge is its emergent property. “We also believe in Learning By Doing as an operative principle” ([Westerberg et al., 1997](#)). This is a clear dedication to evolutionary principles. Since evolutionary processes are intractable, one must go through the motions in order to reach the outcome of such a process.

**Conclusion** Collaborative modelling is difficult to formalize and perform, but it is worth the effort since it increases the involvement and trust the stakeholders have in the outcomes. Collaborative teams must have a good record of their past actions in order to be able to learn from their mistakes. Encoding multiformal knowledge requires the collection of formalized knowledge. Open source tools are important, as teams must have the ability to adapt the tools they use. A collaborative modelling process is an autocatalytic co-evolutionary process. The insights presented will form the guiding principles for the creation of the modelling method and its requirements.

## 5.2.2 Evolutionary Approach

**Co-evolutionary method** This section will explore co-evolutionary thinking within the context of evolving models of  $\lambda$ -system evolution, from simple to complex. Complex  $\lambda$ -systems are not engineered in the traditional sense. Rather, they evolve from simple systems as a result of series of path-dependent, multi-stakeholder decisions and initiatives ([Bijker et al., 1987](#); [Nikolic and Dijkema, 2007](#)), as already discussed in Chapters 2 and 3. Furthermore, this evolution is shaped by changes in a system’s environment. Knowing this, and considering [Ashby \(1968\)](#)’s Law of Requisite Variety, we conjecture that a model can represent an evolved complex system only if it has itself evolved from simple to complex. Applying Ashby again, in order to evolve such a complex model, the method leading to it must itself be complex.

Paraphrasing, only an evolutionary process can be used to describe the process of evolution. In order to model the co-evolution of a socio-technical system, a co-evolutionary approach between different aspects of model development will be used to build the actual models.

The approach consists of an evolving series of case studies detailing and continually improving the social knowledge collection process, elaborating the simulation engine, collecting new multiformal knowledge and collecting new facts. This method is continually generating hypotheses for future work and conclusions in both social process and technical design development. Allowing for the evolutionary nature means acknowledging the path dependent, adaptive, chaotic and intractable nature of the method we are involved in.

**Co-evolving aspects** The process of creating ever more successful models is conceptualized as an evolutionary process involving four aspects of the model creation process. These aspects are interacting within a coupled fitness landscape and trying to “survive”. The concept of coupled fitness landscapes was extensively discussed in Section 3.4.4.

Our four co-evolving aspects are:

**Technology** Technical aspects of model building: which software and hardware system to use, how to organize the modelling software components, how to store data, how to analyze results, how to scale the model, etc.

**Social process** The design of the social process of involving stakeholders in identifying and collecting relevant knowledge and providing feedback on the model's outcomes. Relevant aspects include how to select the right participants, the script of the collaborative process, the manner in which feedback is organized, etc.

**Domain Knowledge** The formalized and encoded domain knowledge representation of  $\lambda$ -systems. Formalized microeconomics, chemical engineering, psychology, etc. are examples of domain knowledge.

**Facts** The factual information that describes components of the  $\lambda$ -system, their interactions and overall system behavior. For example, specific processing plants, their in- and outputs, economic performance data, etc.

As already discussed in Section 3.4.4, evolution is only possible if each aspects continually deforms the fitness landscape (Kauffman and Johnsen, 1991). In this case this means that each aspect contributes new questions and insights to the model development method, making itself more visible or fit by providing insights and results and by exposing the inadequacies of other aspects. In order to illustrate the co-evolution between four aspects that the modelling method consists of, a simplified example is presented in Box 4.

**Box 4: Co-evolutionary method example** In this example, an RDA asks a process engineer to create a model of a local industrial network, so that better decisions can be made about which type of company to invite to join the cluster. A co-evolutionary modelling process could look like this :

**Generation 1** The engineer creates a simple model consisting of agents that have mass in- and outflows. It is built by one person and is based on process engineering knowledge. It encodes facts about the relevant mass flows and it accurately represents the mass flow structure in the region. However, the RDA would like to know how the cluster could react to a change in oil price.

**Generation 2** An MBA is invited into the team, and the model's technical design is extended to encode product pricing, company assets and price setting logic. This model now encodes both process engineering and corporate finance knowledge. The model shows possible network structures, depending on oil price levels. The response, however, is unrealistic, as there is no knowledge on global oil price dynamics.

**Generation 3** A macroeconomist is involved. She provides knowledge on long-term global price movements, and observes that the capital market interest rate changes over time as well. This prompts a change in the model design to enable agents to borrow money in order to invest in new facilities. The model is run and provides possible cluster development pathways, depending on different oil price and interest rate scenarios.

**Generation 4** A resident psychologist realizes that firm's risk attitudes will be relevant if they are to fit in the cluster. The model design is changed, adding the ability to encode risk perception, and since this considerably increases the parameter dimensionality, the model scheduler is adapted to allow model execution on the High Performance Computing cluster. As the outcomes are evaluated against possible agent risk strategies, the RDA estimates that they need to invite a bioelectricity producer with high risk tolerance to join their cluster in order to allow it to grow. However, it is realized that there is a lack of precise factual information on bioelectrical processes, so a data collection campaign is started. The process is continued until the RDA is satisfied.

**Fitness** Evolution requires some form of selection to be present in order to function. How can we know which improvement in the different aspects to cull? Changes between two generations are improvements only if they are 'fit', that is, if they provide us with added useful knowledge or generate new questions, compared to the previous generation. If the fitness is low, the change should not be used. Fitness, or usefulness, within this design method has two aspects. First, the new solution must fit within the requirements (this will be defined in Section 5.4). Second, it must increase our knowledge of  $\lambda$ -system evolution, or the modelling method itself. An improvement must satisfy both.

Any knowledge that is not useful according to the fitness function defined above is not pursued further; it is inactivated in the meme pool (see Section 3.4). It should never be fully removed (forgotten), since the environment can change, and an old and currently useless meme

might become useful again.

**Environment** Fitness is not only defined relatively between the aspects, but also relative to their environment. In this case, the co-evolutionary process between the modeling aspects takes place in a dynamic, socially constructed environment shared by those aspects. The environment of this co-evolutionary process consists of the modellers and the stakeholders who determine what is useful and what is not. This social environment contains elements that are external to the four interacting aspects, such as the sudden availability of resources (money, data, case studies) forcing the co-evolutionary process into an unexpected direction.

**Evolving environment** This environment forms an integral part of the co-evolutionary process. Not only is the co-evolution between the four modeling aspects path dependent (the performance of one aspect depends on past activities of others), but its environment is also shaped by it. The judgment of usefulness by the stakeholder is determined by current and past performance of the modeling aspect. This means that there is no absolute, or objective measure what an improvement is. The co-evolutionary process also shapes the expectations of the stakeholders, inspiring them to try new things or discouraging other courses of action. During this process, the environment of the co-evolutionary process adapts, and in essence co-evolves with the four aspects.

**Variation and multiplication** In this way, a constant deformation of the landscape, i.e., a constant push to change and improve, is created. Constant hypothesis creation and falsification becomes possible. Improve the technical part, see what that teaches you; then keep the best technical insights and improve the social part, and see what new knowledge it provides you, etc. Every new evolutionary move/hypothesis generates new knowledge. The method mimics the natural processes of variation and multiplication (Darwin, 1985). Multiple new ideas are created from older ones, effectively creating knowledge generations.

**Learning as evolution** This modeling method is an iterative and generative process of hypothesis development, testing, falsification and improvement. In itself this method does not deviate from the standard scientific method as defined by Bacon, Popper and others (Popper, 2002). However, the explicit consideration of the evolutionary nature of scientific learning allows for implicit insights to become explicit. For example, the evolutionary perspective reinforces the notion that we *must* make mistakes. The ability to learn from errors is key to the method (see Section 5.2.1). This, in turn, implies among others that mechanisms must be in place to record those errors and requires the ability to reverse decisions in a modular fashion.

**Ownership of mistakes** Science maintains that “a negative results is also a result”, and demonstrating that something is wrong is part of the essence of the scientific method (Popper, 2002). Successful scientific progress consists of doing an endless number of things that don’t directly work but that do help others in trying other things that might. Scientists, however, are social entities. In that context, producing results that are demonstrated to be wrong is often seen as making a mistake and is difficult to accept (Edmondson, 2004). For example, in medical drug research there is strong evidence for systematic underreporting of negative results (van Veldhuisen and Poole-Wilson, 2001).

The co-evolutionary method, just as any scientific process, requires that mistakes be made all the time. In order to make the method work we need to actively reformulate errors as valuable contributions and explicitly encourage making them. After all, in biological evolution, an overwhelmingly small fraction of mutations ( $\ll 0.01\%$ ) is directly beneficial to the organism (Ayala and Kiger, 1980). The goal is to retain the ability to make enough mistakes so that a useful one will occur, without being overwhelmed by the 'failures'.

**Intractable future** Another important insight from the evolutionary perspective is that one cannot plan ahead very well. The intractable nature of the evolutionary algorithm (Dennet, 1996) does not allow us to exactly plan future research steps. One typical example of such inability to plan is this thesis. In a more traditional PhD thesis one would plan ahead the case studies that are needed to support the argument. In a PhD thesis on an evolving method, the case studies become obvious only gradually. Their nature depends on the type of insights gained before, on the social setting in which the work is performed and on external environmental factors such as an unexpected availability of new data.

**Conclusion** Co-evolution is conceptualized as a fitness landscape deformation between the social process of collecting knowledge, the design of the simulation engine, the formalized knowledge and the facts collected. Issues of variation, multiplication and selection were discussed, as was the importance of the environment. The collaborative learning process is placed in the context of intractability. Together with the previous section on collaborative modelling, this section forms the basis for the guiding principles for modelling method development.

## 5.3 Guiding Principles

This section will present the guiding principles that will be used to create the requirements for the co-evolutionary modelling method. The guiding principles are based on the insights from the complexity framework (Chapter 3), the knowledge domains (Chapter 4) and the insights obtained from the core concepts. The principles discussed are: local optimization, the absence of a termination criterion, the importance of path dependency and historic record, the trap of sunk cost, the need for a common interface, the necessity of modular design and shared effort.

The previous section discussed the collaborative modelling and evolutionary background of the methodology for modelling  $\lambda$ -system evolution. Based on these backgrounds, principles for guiding the co-evolutionary processes can be deduced that will result in the creation of the requirements for the method, thus defining the co-evolutionary fitness landscape. There is no guarantee that the list of principles is complete. Other sets might be possible. However, we conjecture that the presented principles are sufficient for defining a useful set of requirements for the method.

Recognizing this evolutionary perspective explicitly, we also must recognize the fact that the first method to be developed will be simple, incomplete and possibly partly incorrect. Nonetheless, it will be good enough as a first start.

**Local optimization** When we consider being in a evolutionary process, the notion of “good enough” is important. There are two main reasons why good enough is so important. First, what is perfect design anyway? It is impossible to determine, since the environment (and thus the fitness landscape) that determines whether or not the design is perfect is constantly changing. Perfect is thus a fluid, ever-changing and unattainable goal, not a fixed, attainable state. Furthermore, evolutionary processes are not teleological. Evolution has no grand goal or purpose. Instead, evolution is a local optimizer, meaning that each incremental step is valuable in itself. Evolution, given its algorithmic nature, does not produce the ‘best possible’ solution; it is not a global optimization process. It creates solutions that are *good enough* for the given situation (Darwin, 1985; Dennet, 1996).

Second, even if we could determine the perfect design, it would certainly be a waste of resources to achieve it, since good enough is exactly that, good enough. Evolution, for example, is not about designing the perfect predator, just a predator that predates successfully in the here and now.

When creating the possibility for a evolutionary series of models that would model  $\lambda$ -system evolution, we must make incremental, small and good enough steps. We should not design the perfect model that one day, in the distant future when it is completed, will exactly do what we need it to do. Instead, we should start with a small model that helps us to understand  $\lambda$ -system evolution a little better and adapt or change it as new insights, knowledge, facts and techniques become available.

**No termination criterion** Since we are developing a series of models that ‘grow’ over time, the question that arises is when to stop, as evolutionary processes do not have an inherent termination criterion. Life does not just stop suddenly<sup>2</sup>. Natural systems do not progress in the sense of gradual improvement, but instead simply continue to evolve. If we are to take the evolutionary nature of growing models seriously, this means that the model building method never stops, nor should it. Obviously, this also holds true for all scientific work. Of course, there are many practical conditions that indicate when a particular model or case study is finished, as will be discussed in Section 5.5, but each model should be developed with the sense that it is just a small (but essential) step in an ongoing journey.

**Path dependency** Co-evolutionary development of models means that a model’s future is dependent on its past. In biology, natural evolution is path dependent and is sometimes summarized as “more and more variation on fewer and fewer themes”. Natural evolution modifies and adapts the material it has at hand, only rarely coming up with totally new designs (Dawkins, 1990). For example, the basic axially symmetric vertebrate design has evolved into a myriad of shapes and environmental niches. The importance of this principle is that a modelling effort will become path dependent the longer it goes on, and that therefore care must be taken to prevent lock-in and the inability to change. In the evolution of models we observe similar phenomena. In economics, for example, Leontief (1998) came up with his idea of input/output models of economies in 1941. Today, General Equilibrium models have much more power but still use the key structuring elements of Leontieff’s input/output table. In chemical engineering, the concept of unit operations was developed in 1923, and is still pervasive in all chemical engineering models. Path dependency also implies that history is important. Where you are

---

<sup>2</sup>Obviously, organisms are born and die all the time, but Life seems eternal.

today is determined in part by where you came from. If you observe errors generated by today's models, one source might be the data, a second could be the implementation or setup of the model. In the latter case, only if you can inspect your history can you start to unravel the error's cause.

On the upside, path dependency implies that each successive model will have all the best components of all the past models. On the downside, path dependency manifests itself in the fact that each and every decision made regarding technical design, social process design, knowledge collected and stored facts will have known and unknown but inevitable consequences later, and some of these will prove to be a limitation.

**No sunk cost** Sunk costs are costs that have been incurred and which cannot be recovered. In biological evolution, when the process of gradual development stops producing viable solutions, for example when the environment drastically changes, designs that do not work anymore are summarily abandoned and new ones are explored. This happens regardless of the time and 'evolutionary cost' that was spent on creating the species. Nature is insensitive to sunk costs. The fact that dinosaurs were highly evolved, diverse and dominant life forms on planet Earth for hundreds of millions of years was irrelevant when rapid environmental change disrupted their fitness landscape and drove them to extinction. Mammals, marginalized up to then, were able to take over as a dominant life form.

When creating a series of models for  $\lambda$ -systems, inevitably the past will come to haunt us when our model appears to be inappropriate or unusable for a priority question, despite the effort and money spent. Psychologically and culturally determined sensitivity to sunk costs prevents the rapid disposal of ineffective model design. Wrong or inefficient designs are kept in place simply because there has been a large resource investment in them (Arkes and Blumer, 1985; Knox and Inkster, 1968). Unlike biological evolution, which would lead to scrapping all and losing everything invested, our approach would be to consciously disassemble the model, take out some unusable or faulty parts and reassemble it while adding new materials, thus minimizing the perceived loss of sunk costs. The method of model evolution for  $\lambda$ -systems must therefore be equipped to dispose of unsuitable parts when necessary. And of course, we must be able to take out and replace parts.

**Modularity** Systems are made of system components that by definition require common interfaces between them in order to interact. When interfaces are standardized, system components can be interchanged. For example, modularity is one of the factors that enabled the Linux software ecosystem to be so successful. Linux was designed with modularity and reuse in mind, so it is very easy to "plug and play" components, increasing the overall functionality of the system. It allows unexpected combinations of components to be made, emerging new products and services. It has been argued<sup>3</sup> that even if the source code of Microsoft Windows was made public, it could never emerge the same type of ecosystem, as it is not built in a modular fashion. The main lesson is that standardized interfaces between system components are essential in any modular, evolving system. In an ABM for  $\lambda$ -systems we can also see three layers of universal interfaces that determine the interaction between agents, making them modular in the process.

**Inside the agent** Interfaces inside agents allow an agent's identity to be formed. The agent's identity is formed when data, knowledge and rules from different formalisms in-

---

<sup>3</sup>[http://www.oreillynet.com/pub/a/oreilly/tim/articles/architecture\\_of\\_participation.html](http://www.oreillynet.com/pub/a/oreilly/tim/articles/architecture_of_participation.html)

teract. The technical formalism thus describes how a given good is produced as well as which resources and how much of them are required for its production. The economic formalism interacts through the internal interface and determines the price that needs to be asked for that good. The decision formalism determines which supplier is preferred for acquiring the resources.

**Between the agents** Interfaces between agents allow agent interactions to form the emergent behavior of the  $\lambda$ -systems. An agent's external interfaces describe all possible interactions of which it is capable; in other words, these interfaces span the agent's potential interaction space.

**Between model, domains and users** Interfaces must be socially constructed and shared. Since the aim is to encode multiformal knowledge, many people will be involved in the process. Since having an interface with which nobody can interact is not very useful, the interfaces between domains, and thus between agents, are best constructed by the social group from which the multiformal knowledge is drawn. Again, this is an evolutionary process, in which the interfaces become shared and accepted between participants.

In Appendix [A.1](#) these interfaces are extensively discussed.

**Shared effort** Any evolutionary process is necessarily a co-evolutionary process; one does not evolve alone (see Section [3.4](#)). Furthermore, when we consider the notion of multiple formalisms as discussed in Section [3.3](#) it becomes obvious that a shared, concerted action is necessary to tackle multiformal, complex problems. The core concept of collaborative modelling was extensively discussed in Section [5.2.1](#). Coupled to this, the newest insights in social diversity ([Page, 2007](#)) reinforce the idea that successful understanding  $\lambda$ -systems must be a shared effort involving a great diversity of people.

Science, despite the objective and value-free nature of the scientific method, is done by people in groups. Whenever people are involved, egos matter, personalities clash and all sorts of irrational emotional issues come into play ([Calvin et al., 1957](#)). The social interaction process is chaotic, path dependent and adaptive. In situations where teams must collaborate the design of the social process must be organized in such a way that it can deal with the inevitable conflict. Collaboration rules, group habits, workflow scripts and team atmosphere must all be at least taken into consideration [Jehn and Mannix \(2001\)](#) and, where possible, maybe even explicitly engineered.

## 5.4 Requirements

In this section the requirements for the co-evolutionary modelling method and for the method outcomes are defined.

**Recapitulation** The previous sections introduced the notion of collaborative modelling and the co-evolutionary approach to modelling, in which technical design, social process design, formalized knowledge and collected facts co-evolve in a coupled fitness landscape. Based on these core concepts and on the scientific and modelling background presented in Chapters [3](#) and [4](#), a number of guiding principles were defined. In this section those guiding principles will be

operationalized as requirements for the modelling method and its outcomes. The requirements can be seen as the fitness function or selection mechanism for the coupled fitness landscape of the co-evolving aspects. If a change in one of the aspects does not meet the requirements, or if the method produces models that do not meet the criteria, these new adaptations will be considered unfit and will not be propagated in the next generation.

**Requirements** Herder (1999) defines functional requirements as specifications of functions that the design must provide. There are two types of requirements: method requirements and outcome (or model) requirements. Method requirements help shape the method design. They are specific and based on the guiding principles of evolution and modelling. The outcome requirements help establish whether the models created by the modelling method are good enough. They are fairly generic and are based on the usual criteria for scientific output. A summary of the requirements is presented in Table 5.1. The requirements are elaborated in Sections 5.4.1 and 5.4.2.

Table 5.1: Overview of Method and Outcome Requirements

Method	Outcomes
Open source	Useful
Sufficient community diversity	Testable
Organically growing	
Recorded history	
Enforceable authorship	
Modular	

### 5.4.1 Method Requirements

**Open source** Scientific work has traditionally been based on peer review. Research methods, data and results are presented to others for inspection, repetition, confirmation and improvement. However, the work is still done behind closed doors, and only when deemed ready is it carefully released. While this has worked well, new multidisciplinary problems require more radical forms of collaboration (Kepler et al., 2006). One should not only be able to examine the end result but also the knowledge creation process itself, with all of its faults and dead ends. The process of knowledge creation, especially when creating software models, can be made visible by using an open source approach during development. Furthermore, allowing the computer code used and developed to be open encourages standardization. This is captured in the guiding principle of shared standards. Why reinvent the wheel when a perfectly functional implementation is freely available? An example is the Colt pseudo-random number generator and statistics library developed by CERN (Hoschek, 2004). Finally, a freely accessible shared knowledge base (what an open source computer code essentially is) promotes community building, since it is possible for others to get involved. Work developed and published with open access in mind has a greater research impact than when it is not freely available (Antelman, 2004). Therefore, this requirement can be summed up as: *Is the work openly accessible?*

**Sufficiently diverse community** This requirement is based on the guiding principle of shared effort. As already discussed in Section 3.3, a  $\lambda$ -system can never be understood by a single formalism or a single person. In order to successfully undertake a large and complex process of understanding  $\lambda$ -system evolution, a creation of a diverse social network is necessary (Page, 2007). The diversity of people involved in the process has to be large enough to cover all the main conceptual areas needed in the modelling effort. Therefore, this requirement can be summarized as: *Is there a sufficient diversity of people involved?*

**Organic growth** Based on the guiding principle of local optimization, the requirement of organic growth can be formulated.

If we consider, for example, a species that develops and adapts to its environment, we can see that the adaptation consists of a series of local optima. Each individual of the species 'wants' just one thing: to survive and reproduce (Darwin, 1985; Dawkins, 1990). This individually selfish behavior creates an overall better species. In social systems, mutually contingent self-interest serves a similar purpose (Karp, 2006). All participants in a social system work on improving their own well-being, wealth, status, etc. by collaborating on the creation of something much bigger than they could achieve alone, without limiting the others. This causes the system to change itself while ensuring that the changes are in the self-interest of all participants and are internally motivated. Therefore, this requirement can be summed up as: *Does the system develop from the inside out?*

**Recorded history** Based on the guiding principle of path dependence, and in order to fulfill the requirement of falsifiability, the method and the work done according to it must have as accurate a historic record as possible (Westerberg et al., 1997).

Achieving a historic record is traditionally done by archiving the created documents. In the case of digital data and software code, this can be achieved by housing the generated formalized knowledge in a Version Control System. For example, the project specifications document has all previous versions available for inspection. Full versioning basically means that we can go back to any point in time during the development of the method or model and be able to recover the state of knowledge and factual information that was available at that time. The main reason necessitating this capability is the chaotic nature of evolutionary processes. During such a process, often times decisions are made that significantly affect the system's future development pathway. From those points onward, the system can continue in a number of possible directions. If the development process leads the system to find itself in a dead end, we must be able to go back to those turning points and undo/redo them. Two distinct types of knowledge need to be recorded and placed in a Version Control System: formalized knowledge and facts, and unformalized knowledge. The first is relatively easy to achieve with software tools. The second is much more difficult, since a lot of unformalized knowledge is transferred through interpersonal communication. Therefore, this requirement can be summarized as: *Is a versioned history of knowledge accessible?*

**Enforceable authorship** This requirement is based on the principle of local optimization and shared effort. Enforceable authorship is a requirement that allows us to know who did what, when. In a well-designed system, this requirement is automatically achieved in concert with the previous one. There are two reasons for this requirement. First, since we are involved

in a social process of methodology development and model building, no matter how good our record keeping is, there will always be aspects of design decisions and fact collection that are not recorded. The record of authorship provides a link to the social memory of the group. The second reason is that in an academic environment it is important to give credit where credit is due. With a good authorship record, knowledge can always be attributed to its original creator, making intentional and unintentional plagiarism far less likely. Therefore, this requirement can be summarized as: *Is it recorded who did what and when?*

**Modular** The functional requirement of modularity is based on the guiding principle of shared standards. The design must be such that any component can be replaced without breaking the overall functionality (Egyedi and Verwater-Lukszo, 2005) of a system. The replacement can either be functionally identical yet implemented differently, or it could extend the system's functionality. In a sense, the entire design can be seen as a System Of Systems (DeLaurentis and Crossley, 2005), in which the overall system consists of coupled components, and the components have porous boundaries and have a meaningful functionality in themselves. Furthermore, modularity also implies reusability. By creating a consistently modular design, we are effectively building up a library of components that can be recombined at will to create novel models or methods. Therefore, this requirement can be summarized as: *Does the system consist of interchangeable modular components?*

#### 5.4.2 Outcome Requirements

**Useful** The usefulness requirement is based on the guiding principle of local optimization. Nature does not make things that will be useful at some point in the future, only ones that are useful right now. This immediate usefulness of the outcomes of the modelling method can be applied at two levels.

First, at the outcome level, the question is whether we have produced a model that is useful to the stakeholders; can the stakeholders use the model outcomes as intended?

Second, at the method level, the question is whether the modeller designing the modelling method has learned something useful from the outcomes; i.e., are the outcomes of the model useful for improving the modelling method?

As long as the answer is yes on at least one of the levels, we can consider this requirement to be met. Given this qualification, usefulness can be seen as a somewhat ambiguous requirement, since it can always easily be made to be met. Even when a step in the methodology or a case fails to produce the desired results, we can still learn what does not work in terms of modelling method, and thus it is useful. Usefulness is therefore not an objective criterion. Usefulness can only be determined in interaction between the stakeholders and the modellers. This requirement can be summed up as: *Did the model outcomes deliver useful insights to stakeholders or modellers?*

**Testable** The most important requirement for any knowledge generated within the scientific endeavor is falsifiability or testability (Popper, 2002). Again, we can differentiate two levels of testability: the method and the model produced by it.

In order to allow testability at the model level, several requirements need to be satisfied. First, as specified by the open-source requirement, the computer code and data that the model consists of must be known. Second, each model run must be fully repeatable. In a model that

has stochastic components, two further requirements are necessary in order to ensure repeatability. First, the archived recorded history allows for each developed model to be retrieved and rerun. Second, by using an open source deterministic pseudo-random generator and a known seed value, all the (pseudo-) stochastic components of the models can be exactly repeated. A model run that meets the above specifications is fully repeatable and thus testable.

Considering testability at the method level, the fact that we are part of a co-evolutionary process makes the issue more problematic. Repeatability in the narrow sense discussed above is only possible in fully deterministic, non-chaptic systems. Learning always occurs when a social network is involved in an evolutionary process. Redoing the development process with the same social network will necessarily yield different results.

However, the method itself is repeatable, due to the historic record of unformalized and formalized knowledge. It is therefore possible to test whether an execution of the same method will yield similar results if a different social network of similar qualification is used. Obviously, the models produced by the two networks can never be identical, but they can be sufficiently useful and similar to falsify the methodology. Thus at the method level, testability must be defined as the ability to repeat the method with a different group of people and achieve sufficiently similar and useful models both times. It follows that repeatability is defined as a similarity of patterns, not exactness. We conjecture that this is as far as testability of a social learning process can be taken. This requirement can be summarized as: *Can it be tested?*

**Conclusion** This section presented the requirements for the method and outcomes of the co-evolutionary modelling approach. Method requirements are: the use of open source tools, sufficient diversity within the modelling community, organic growth, a record of history with an enforceable authorship and modularity. The models produced by the method are required to be useful and testable. Now that we have described the method conceptually, its guiding principles and requirements, it is time to explicitly present the practical steps involved in creating models of  $\lambda$ -system evolution.

## 5.5 Method description

### 5.5.1 Method

Given the guiding principles and the requirements, the method for generating successive models is rather straightforward.

The basic assumption of the co-evolutionary method is that it proceeds in steps or generations consisting of case studies. Each case generates insights that are used to improve the modelling method of the following case. The method is repeatable, and the content of the cases is determined by the social network involved.

**Four aspects** Each case has a social, technical, knowledge and factual aspect, as discussed in Section 5.2.2. Knowledge describes the actual domain specific knowledge necessary for modelling. The social aspect describes how the knowledge collection will be organized, i.e., which techniques and collaboration scripts will be employed in order to collect the relevant knowledge from the stakeholders. The technical aspect describes the modelling tools used to

encode this knowledge and to analyze the results. The factual aspect is the collection of facts and data used to fill, calibrate and validate the models produced.

**Unbalanced progress** In order to ensure knowledge generation, the modelling method leading to each case is unbalanced, that is, one aspect will be extended/explored more than the others. For example, a novel modelling tool will be tested, while keeping the social and factual designs constant, in order to learn about its effectiveness. The goal is to keep a *ceteris paribus* assumption as much as possible, as is usual in modelling work. We are of course limited by the practical requirements of the case study, so this might not always be possible. This method of unbalanced model development is illustrated in Figure 5.1.

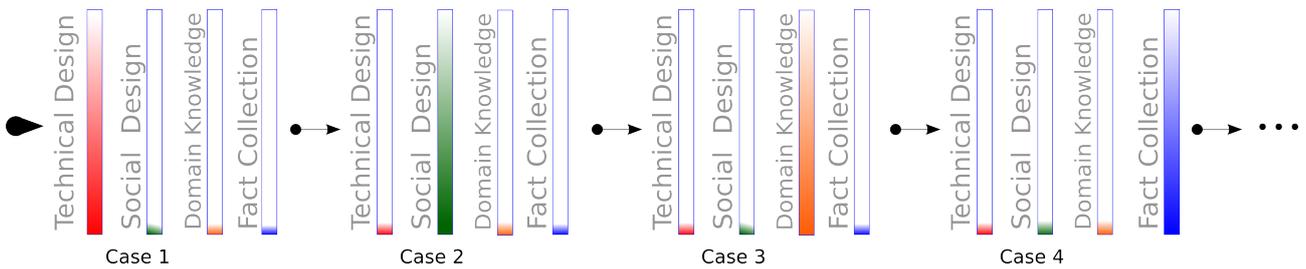


Figure 5.1: Initial method steps

**Hypothesis generation** In Figure 5.1 each block represents a case study with a corresponding model. The colored bars depict the height of the deformation of the coupled fitness landscape in that dimension. Each case starts with a number of hypotheses. These hypotheses are at two levels:

- Whether the planned extension of the methodology will prove to be useful; and
- Whether the specific case assumptions will hold true.

We assume that we start with the following technical feasibility hypothesis: It is possible to technically design a model that has the necessary properties to model  $\lambda$ -system evolution. If this hypothesis is not rejected, we will proceed to design a social process. The next hypothesis is: The proposed social process design is able to involve stakeholders in modelling and understanding  $\lambda$ -system evolution. The third step starts with the hypothesis: The collected and formalized domain knowledge about the  $\lambda$ -system under study is useful in understanding the system's evolution. The fourth hypothesis is: The collected facts on the  $\lambda$ -system under study will allow for useful model creation and novel insights into the system's evolution process. After these four initial steps, the modelling method is extended in whichever direction offers greatest insights, and the direction it will take cannot be predicted in advance.

**Stop criterion** There is no theoretical end to this process. In a more practical sense, one stops when the project defined with the stakeholders is completed and the stakeholders are satisfied and/or the resources for modelling are consumed.

## 5.5.2 Activities

Building models for any given case study requires a set of activities to be performed. Consecutive models and model improvements are developed in a coherent method with multiple case studies zooming in on different aspects. This means that some activities may become trivial for cases completed later in the series, as they may exhibit much similarity to activities completed in a previous study. As a matter of course, novel aspects will always have to be elaborated in greater detail.

Since the implementation per case study is dependent on the evolutionary method of model development, the activities are described in generic terms. These have served as blueprints for the modelling activities in real cases reported in Chapters 6, 7 and 8. The series of activities to be completed for each case study is as follows:

- Create the collaboration conditions;
- Collect and formalize knowledge;
- Collect facts;
- Implement the model;
- Verify the model implementation;
- Analyze the model outcomes; and
- Validate the model.

**Create the collaboration conditions** The case study starts with the creation of the necessary social conditions for performing the model development. This may include identification of stakeholders and experts, agreements on the collaboration process, etc. It serves as an initialization of the social network that will support and use the model and its outcomes.

**Collect and formalize knowledge** Using the social network created, a system decomposition method is created. The specific aspects of such a method are discussed in Section 6.5. During this process a relevant description of the  $\lambda$ -system under study is created and formalized, and relevant domain knowledge is identified and formalized.

**Collect facts** This activity is focused on increasing the factual content of the case study. During this activity relevant facts are added to the formal description created in the previous step. This activity can be strongly tied to the previous step, but it need not be. The involved stakeholders can sometimes have the relevant factual knowledge, but at other times it has to be collected from other sources.

**Implement the model** During this activity the technical design of the model is created and implemented. Using the software tools conforming to the requirements, a software implementation of the formalized knowledge and  $\lambda$ -system is created.

**Verify the model implementation** Once the technical implementation is completed, the model needs to be verified, in the sense that it must be free from programming errors and cannot contain behavioral artifacts. We need to ensure that the model is what we intended it to be.

**Analyze the model outcomes** During this activity the model’s outcomes are analyzed. Basically, we examine the model across the parameter space and identify the model’s attractors. This provides insights into the range of possible behaviors that the model is capable of. This activity provides the bulk of the insights into  $\lambda$ -system evolution.

**Validate the model** Validation of Agent Based Models describing possible patterns of  $\lambda$ -system evolution is exceptionally difficult and non-trivial. It will be discussed in greater detail below.

### 5.5.3 Verification and Validation

As already mentioned, evolution is an intractable, generative process. This means that the outcome can not be determined in advance. This begs the question of how to make sure that the generative models, generated by the evolutionary method, are verifiable and validable. In other words, is this science? In the words of Epstein ([Epstein, 1999](#)):

Does the hypothesized microspecification suffice to generate the observed phenomenon?; be it a stationary firm size distribution, a pattern of alliances, or a nonequilibrium price time series. The answer may be yes and, crucially, it may be no. Indeed, it is precisely the latter possibility, empirical falsifiability, that qualifies the agent-based computational model as a scientific instrument.

As long as models generated by the co-evolutionary method confirm to the requirements of testability, the answer is a resounding yes. We can therefore continue to discuss verification and validation in the next sections.

#### 5.5.3.1 Verification

**Types of inputs** There are four classes of inputs to the method that need to be verified. These are presented in [Table 5.2](#).

Table 5.2: Types of inputs and ease of verification

	Physical	Social
Facts	Easy. Objective, Measureable	Medium. Subjective, Measurable
Knowledge	Medium. Objective, Formalized	Hard. Subjective, Unformalized

We can make a distinction between facts and knowledge and between physical and social domains. Knowledge is defined in this case as the codified experience of its users ([Stefik, 1995b](#)). Physical facts are easy to verify. One can objectively measure them. Physical knowledge is more

difficult, since it represents an encoding of physical laws of nature into a knowledge structure of the domain experts. Knowledge needs to be formalized in order to be used in the co-evolutionary modelling method. However, since it is based on physical reality, it is relatively easy to verify the encoding. Social facts are difficult to verify, as they are not objectively defined; they are dependent on the social context from which they are taken. Since they are measurable, their verifiability is medium. The most difficult aspect to verify is social knowledge. Not only does this knowledge represent a codification inside the domain expert's mind, but it is also dependent on a subjective experience of social reality. It represents a codification of the social consensus on a certain aspect of social reality, and great care must be taken to achieve stakeholder consensus during the process.

**Verification on two levels** Verification needs to be performed at two levels. At the multiformal knowledge level, it is important to verify that the knowledge encoded is indeed that what the domain experts have contributed, i.e., is the encoding correct? At the simulation level, it needs to be verified that the simulation code corresponds with the knowledge collected, and that the code indeed does what it was designed to do.

**Knowledge level** Verification of whether the knowledge encoded is in fact the knowledge that the experts communicate is performed through the social process. It is built into the formalized social process, which is a participative method that involves continuous interaction between the modellers and the domain experts (see Section 6.5). The codification and formalization of the knowledge is an integral part of the method. This means that the domain experts can check the formal representation of their knowledge during the process, verifying the formalizations that the simulation will use.

**Simulation level** The correspondence between the design (collected knowledge) and implementation in software is done using revision control, providing a historic development path and by using unit testing. Revision control records all the changes to the computer code base over time, providing the ability to go back to any point in time and examine what was done when and where. Unit testing is a technique that focuses on developing tests for functional components of computer code. Functional tests can be specified for computer code elements by the stakeholders, and the implementation can be tested as to whether it performs that function. Both concepts are discussed in greater detail in Appendix D.

### 5.5.3.2 Validation

Validation is exceptionally difficult when dealing with models that explore patterns of future developments of systems. Epstein argues (Epstein, 1999):

From an epistemological standpoint, generative social science, while empirical, is not inductive, at least as that term is typically used in the social sciences (e.g. as where one assembles macroeconomic data and estimates aggregate relations econometrically).

The relation of generative social science to deduction is more subtle. The connection is of particular interest because there is an intellectual tradition in which

we account an observation as explained precisely when we can deduce the proposition expressing that observation from other, more general, propositions. ... In the present connection, we seek to explain macroscopic social phenomena. And we are requiring that they be generated in an agent-based computational model. Surprisingly, in that event, we can legitimately claim that they are strictly deducible.

This deducible nature of ABMs offers two main paths for validation: historic replays of evolution and scenario testing through expert consultations. In the first path, we attempt to recreate a current situation by taking a (somewhat) known starting point in the past, together with the change of environmental conditions over time, in order to come to a  $\lambda$ -system state as observed today. In the second path we explore interesting development scenarios for the  $\lambda$ -system evolution and discuss these with experts/stakeholders.

**Consequence of intractability** Traditionally, the validation of models is done by performing experiments. If the modelled outcomes correspond with observed reality, the model is validated. In the case that we need to make a prediction about the future state of the world after some change has been implemented, the validating experiment would consist of making such a change in the real world, and observing the effects. However, running such an experiment at the scale of  $\lambda$ -systems is clearly impossible. That means that validation is *impossible in the traditional sense*. We cannot form a hypothesis, perform a real world experiment and see if the theoretical model outcomes correspond with reality. However, the *method of creating models* that describes the evolution of  $\lambda$ -systems can be validated. It can be repeated with different stakeholders and domain experts, and the resulting models can be compared.

**Future states** The outcomes of the social and technical design method are not quantitative predictions about the future state of the  $\lambda$ -system. The outcomes are predictions about the diversity and identity of possible future states of a system, most of which will never occur in reality. The outcomes are in a sense a look into the potential futures of the system. Moreover, when we have the knowledge of what the outcomes are likely to be, it is almost guaranteed that that particular outcome will not happen, since the knowledge itself will cause the future to be changed. It is a kind of feed-forward regulation process. The real outcome of the simulations is increased insight and knowledge about the possible evolutionary states of the system - and that outcome can be validated.

**Social validation** Since the goal of social validation is to provide insight and increased knowledge, the approach is validated by the stakeholders themselves. The approach is therefore validated at the moment that stakeholders believe that they have increased their knowledge about the evolutionary processes and the possible states of the system. Traditional social science tools, such as expert consultations and interviews, are used to judge this validity.

**Conclusion** This section first presented the steps of the co-evolutionary method for creating models of  $\lambda$ -system evolution. The method described was a co-evolutionary development of four modeling aspects, shaping each other as they develop. Guiding principles and the method and outcome requirements generated from them were defined. The specific activities of the modelling process were discussed, as were the issues of model and method verification and

validation. We are now theoretically and practically equipped with the tools needed to begin modelling the evolution of  $\lambda$ -systems.

As is often said, the proof of the pudding is in the eating. In the next chapter, we will start the modelling method and explore three case studies that serve as a start for the evolutionary process.

## Part II

### Practice: Co-evolutionary method in action



## CHAPTER 6

# LEARNING: CASE STUDIES AND KNOWLEDGE ENGINEERING

*Disco Ergo Sum*<sup>a</sup>

---

<sup>a</sup>I learn, therefore I am.

### 6.1 Introduction

**Three cases and a result** Three case studies are presented in this chapter<sup>1</sup>. The Flow-based Evolution model broadens our options for the exploration of technical design. In the Combination of Infrastructures case, we investigated the social process for knowledge formalization and fact collection. In the Chocolate Game case study, lessons learned were used to implement a new social process design and technical design.

**Case description structure** Each case study is presented using the following format:

**Focus point** Each case study is part of the learning process; each is another step toward developing the modelling method and generating domain-specific knowledge and content.

**Hypotheses** Where appropriate, hypotheses are formulated which relate to the modelling method and to the case study.

**Case description** Here we present the subject of the case, the social and technical networks involved and the case-specific questions to be addressed.

**Model Details** The description of the model details covers the knowledge encoded, the facts collected and the details of the simulation engine implementation.

**Case study results** The most important model results and insights in  $\lambda$ -system evolution are explained.

---

<sup>1</sup>Parts of this chapter are based on the paper “Facilitating the modelling of complex adaptive systems” by P.J. Beers, I. Nikolic, P. Bots, G.P.J. Dijkema, submitted to JASS.

**Method development conclusions** This is a reflection on, and summary of, the progress in building a modelling method, along with ideas for its future development.

## 6.2 Flow-based Evolution

**Case abstract** This case study was performed in order to explore the implementation of the conceptualization of a  $\lambda$ -system as an input/output-based flow network in an ABM model. The Flow-based Evolution model describes an abstract industrial network, with nodes being the producers and consumers of mass flows (goods), and mass flows being the edges. The network evolves over time through additions of new agents and through the restructuring of edges. In the simulations completed by the agents the inputs, outputs and types of mass flows are generated randomly. The main results are that the conceptualization can be implemented through a working simulation model. The initial implementation is limited in scope, and the current technical design cannot be easily extended. However, it served well as a first test. The main insights for model development are that the modelling approach helps to determine which types of facts need to be collected, that technology description and the economic decision-making processes must be separated and that in addition to general network metrics, domain-specific metrics are needed.

### 6.2.1 Focus Point

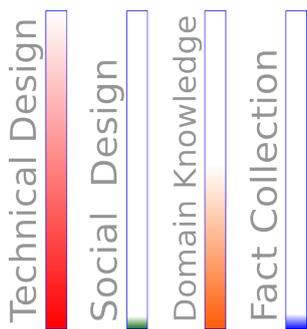


Figure 6.1: Case focus

**Methodological focus** The method focus of this case study was on the technical modelling aspects. Domain knowledge collection was limited, see Figure 6.1. Through the technical design of the model, a first abstraction of a  $\lambda$ -system was created. This involved questions about the knowledge and data necessary to describe the  $\lambda$ -system (see Section 5.5). The idea was that these questions would later enable us to design the social process needed to collect relevant knowledge and facts.

In this first step in the evolutionary process of model development, the technical design provided a tangible starting point. We expect it to help us start the social process and fact collection around a 'real' product, and it is in accordance with the guiding principle of local optimization (see Section 5.3).

**Case focus** The case study content focused on the development of a simplified model of a process industry. Using domain expertise in process engineering and the basic process formalism presented in Section 4.2, a model design was developed and implemented that captures the basic components of evolving  $\lambda$ -systems, the agents (production plants) and their interactions (resource and product flows). This model makes it possible to test the applicability of network metrics defined in Section A.2.

### 6.2.2 Hypothesis

**Basis of the hypothesis** One of the objectives of this thesis was to gain a better understanding of processing industry networks (2.3.1), particularly the dynamics of network evolution. An important property of the processing industry is that processing plants convert flows of specific raw materials (mass or energy) into flows of products (see 4.2). The physical foundation of a

process industry network is therefore an interconnected mass flow network. Therein, process plants are the nodes and the flows among them are the edges. Each of these nodes is a discrete entity with its own dynamic behavior. Over time, nodes are added or removed and edges are formed or broken. This discrete, dynamic flow-based characteristic of the system is what needs to be captured in a useful modelling abstraction, in order to be able to address questions such as: “How does the making and breaking of material streams between processing plants affect the form and function of the network?” Therefore, the hypothesis tested in this case study was:

**Hypothesis** *It is possible to create a template for models that simulate the growth of industrial networks which is based on agents that have a given number of certain types of in- and outflows and connect to each other by locating the outflows of other agents that match their own inflows.*

### 6.2.3 Case Description

**Case objective and goal** The case objective is to create a coherent model of network evolution of networks connected by material flow exchange. Our goal is to create an agent-based model that respects chemical and physical flow conversion principles. We do not aim for quantitative descriptive power, however, but rather aspire to creating a model template - a model structure and a formalization of the basic process engineering domain concepts that is both generic and extensible. When filling the template with facts, a network model should result that exhibits essential behavioral network characteristics that are similar or analogous to the behavior of real-world processing networks. While face validation thus is an option, verification to real-world data of the model template is impossible, as discussed in Section 5.5.3.

**Template and model** As stated, the model template is an abstracted representation of a process industry network with nodes as producers and consumers of mass flows, and mass flows as edges. The network evolves over time through additions of new agents and through the restructuring of edges. Thus the model template contains a structural description of a system, of behavioral mechanisms and the basic processes and laws of nature. These are formal representations of knowledge. In order to run a model, this model template needs to be filled with the appropriate facts that would then make it into an ABM of the specific case studied and allow its simulation.

**Knowledge and facts** In the Flow-based Evolution model, no real industry has been modelled and no real facts have been implemented, because our purpose is to first create a model template. The test model does not need to contain real factual information, as our first objective is to create something with a run button that is a suitable representation of a process industry network. To this end, the random generation of agents, their inputs and outputs and the types of mass flows allows us to test the model template and create a model that can be run. In such a 'fact-free' model, an agent that represents some randomly specified 'processing plant' still 'knows' that the total mass of its inflows must match the total mass of its outflows. And it may also know that its total cost of operating some machinery is a sum of operational costs and discounted investment costs. In the fact-free model, any agent will check its mass balance and calculate its yearly operating costs; it does not matter whether it has a realistic value for the size of its flows and associated costs. In other words, an agent in a fact-free model

can be fully randomly generated; a die can be thrown to determine any of the types and values of its flows and costs, but it will still act in a coherent and realistic manner upon these values.

Would a model template filled with random facts not result in garbage in, garbage out? Yes, at the agent level, but our hypothesis is at the network level. The first part of our hypothesis says that we can create a (generic) network growth model, the second part says that when running the model, network development patterns and network characteristics will emerge that are analogous to real industry network evolution characteristics. And we expect this to hold, even if the agents in the network are random. Turning the argument around, 'randomly generated' is the largest possible 'agent space and interaction space'. Filling the template with cognate facts that are proven and domain specific would limit the possible 'network space'. That is, with randomly generated facts, the model will create the widest possible range of network structures, at least some of which should resemble real evolving network structure patterns.

**Flow-based Evolution model** Our objective was to create a model template that would allow process industry network evolution simulation. In order to test and explore the usefulness of the model template, the Flow-based Evolution model was created as a first implementation. As this model is fact free, careful consideration was given to what should be included in the model and what not. At all times, it was of prime consideration that the focus was on establishing whether we were on the right track with our modelling approach by testing whether the model would create appropriate network evolution patterns. Using these criteria, the modelling assumptions were:

- No conservation of mass
- No physical distances
- Output is food
- No agent death
- No economics
- Input/output technology
- Connections are forever

These assumptions are discussed in the following paragraphs.

**No conservation of mass** We decided not to implement conservation of mass because this greatly simplifies the algorithms needed to model the connecting process between agents. Thus, only the identity of the flow matters, not its magnitude. Over an edge, conservation of mass is respected by assuming that the total production of one node equals the total consumption of the other, and once an output of a node is connected, it cannot be connected to another node. Therefore, it is impossible to have a situation in which a single supplier delivers a certain type of mass flow to multiple clients.

**Distance and infrastructure** We assume an unconstrained world where there is no physical distance and there is abundant availability of infrastructure. All nodes in the simulation are assumed to be connected either directly via pipeline or via road connections. If nodes want to connect to each other, this is always possible. There are no delays or costs involved due to distance.

**Output is food** The network growth model is an industrial analogy to the ecological notion of food webs. This requires shared standards and the ability of biological entities to eat each other (see Section 5.3). These principles are applied in the model. The organisms - firms - produce outputs that can be eaten by other firm needing inputs. Who should eat what is defined by the random proto-technology description.

**No agent death** Firms (agents) that are added to the network are permanent. No mechanism for node removal has been implemented. The model detailing is too limited to enable inclusion of a meaningful removal model.

**No economics**  $\lambda$ -systems only evolve when somebody, somewhere makes a decision. A flow-based network evolves when companies make a selection of suppliers for their raw materials. In order to reduce the complexity of the real world, it is assumed that the only decision made is the decision of whether a supplying agent can provide a previously unconnected flow of the correct type for the required input. No internal economic model and no concept of prices or value has been included.

**Input/output technology** Instead of detailed technology descriptions, agents possess a concept of technology through the input and output types defined and the set of inputs and outputs they possess (agent 1 converts input A to output B; agent 2 converts C to D, etc.). This proto-technology determines that the agent 'wants to eat' and which outputs it creates that can be 'eaten' by others. Where the agent can get its food is determined by the decision-making process.

**Connections are forever** In order to simulate the evolutionary path of a  $\lambda$ -system, the model assumes that once connections between nodes are established, they will remain until the end of simulation. This makes the model fully path-dependent. The order of appearance of nodes determines the development path. In order to analyze the effect of path dependence, this behavior is made optional and can be replaced by a history-free behavior, in which all connections are broken after each step, and the system is rewired every time a node is added.

## 6.2.4 Model Implementation

In this subsection an overview of the model implementation is given. The Java source code of the model is available online <sup>2</sup>. Building upon the basic agent formalism (section 4.3), the model consists of agents that have a state and behavior (Appendix A.1), which may interconnect and which live in some world (Figure 6.2). The world contains the rules and algorithms for agent creation, it determines the network growth stop condition, schedules the agents' actions and

---

<sup>2</sup><http://gux.tudelft.nl/svn/FlowBasedEvolution/tags/PhDThesisVersion/>

collects the required statistics. It acts as the agents' context. Each agent receives a label consisting of the input flow and output flow numbers. An agent's state consists of the in- and outflow descriptions and its connectedness status. Each flow has a name, a color denoting its status and a record of the nodes it connects.

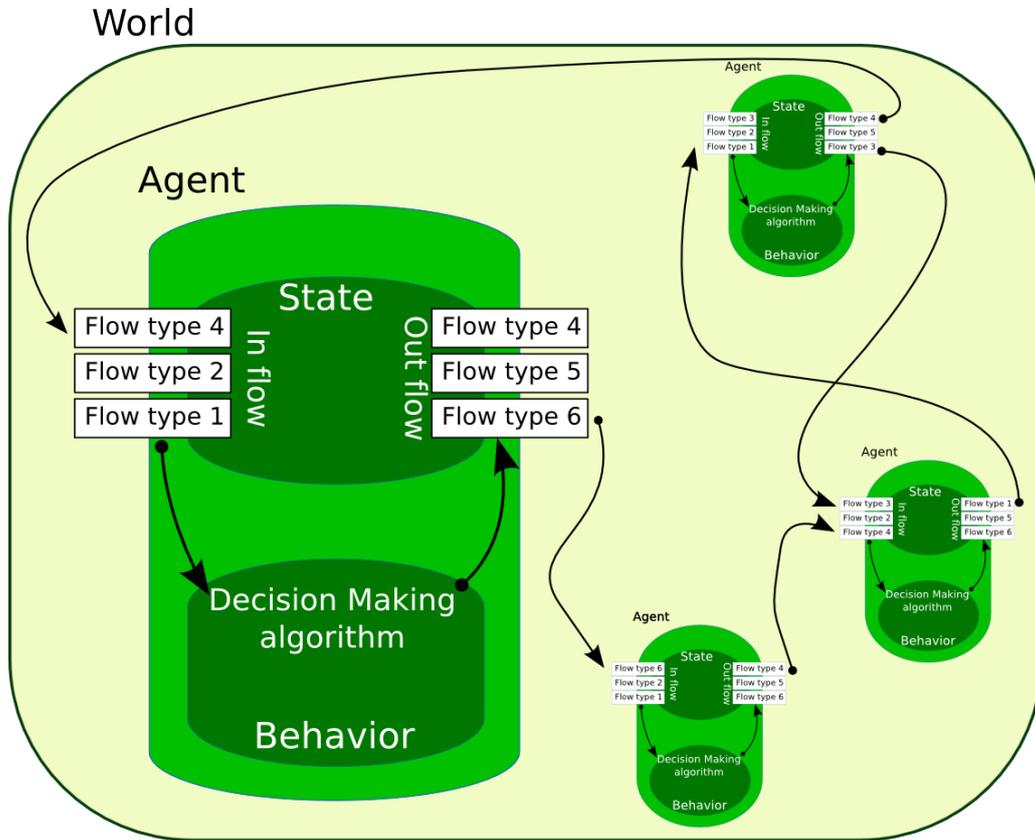


Figure 6.2: Layout of the Flow-based Evolution model

**Decision-making algorithm** In accordance with the Flow-based Evolution model's conceptual design, the agents only decide which other agent will be the source of their incoming flows (6.2.3). Algorithm 6.2.4 gives the simplified encoding.

**Random agent generation** In the Flow-based Evolution model the agents are fact-free and randomly generated (see 6.2). Each agent has between 1 and X input flows of different types, and 1 to Y outputs of different types, drawn from a uniform distribution. There are Z types of flows, called Flow-1 to Flow-Z. The in- and outflow sets are randomly drawn from the type pool. The first component of the in- and outflows list is considered to be the 'main' or reference flow. Once agents have connected an output and input of the same type with a flow, no other connections from that specific output or to that specific input are possible. For example, Agent A has a 4 outflows. The third is of type Flow-12. Agent B has 2 inflows, and the first is also of type Flow-12. A connection is formed between Agents A and B of type Flow-12. No other agents that require Flow-12 as input can connect to Agent A, and Agent B can not connect any other Agents that have outflows of type Flow-12.

---

**Algorithm 6.1** Flow connection algorithm - pseudocode

---

```
for all inFlow in InFlows do
  if inFlow.connected = FALSE then
    wantedFlowType ← inFlow.getFlowType
  end if
  otherAgents ← getAgentsWithFlows(of wantedFlowType)
  for all otherAgent in otherAgents do
    if inFlow.getFlowType = otherAgent.getOutFlowType then
      if otherAgent.getOutFlow.connected = FALSE then
        connectFlow(agent, otherAgent)
      end if
    end if
  end for
end for
```

---

**Network evolution metrics** The following network statistics are collected at each time step: degree distribution, average shortest path length, number of unconnected flow types, degree of completeness and degree of connectivity. The first two are defined in Section A.2. The number of unconnected flow types describes the number of flow types that are available in the network for connection. Degree of completeness and degree of connectivity are defined below.

**Degree of connectivity**  $D_{con}$  answers the question “How much do we deviate from a fully connected graph?” This metric is expressed by equation 6.1 where  $n$  is the number of edges and  $N$  the number of nodes.

$$D_{con} = \frac{n}{\frac{1}{2}N(N-1)} \quad (6.1)$$

**Degree of completeness**  $D_{com}$  is defined by eq. 6.2 where  $n$  is the number of edges,  $N$  the number of nodes, and the  $Deg_{theory}$  is the theoretical maximum degree of a given node.

$$D_{com} = \frac{n}{\sum_{i=1}^N Deg_{theory}} \quad (6.2)$$

This metric is useful for simulated and well-defined networks in which the theoretical maximum degree of a node is known and is a special case of the Degree of Connectivity. These metrics give us a sense of internal coherence of the  $\lambda$ -system. When the metrics are low, this indicates a  $\lambda$ -system that is more ‘open’ to the outside world. High degrees point to a ‘closed’ system that is mainly internally focused.

## 6.2.5 Experiments and Results

In this section the performed experiments and the most important results are presented. The the modelling method insights are discussed in the next section. The graphs presented in this section are meant to convey the general model behavior, not the detailed numerical values, and are therefore compressed to save space. Full resolution images are available online<sup>3</sup>. The model

---

<sup>3</sup> <http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/FlowBasedEvolution/CaseResults.html>

runs can be replicated by running the executable binary available online <sup>4</sup>. All the results presented are typical model outcomes that have large stochastic components, so any repeat run will necessarily have a slightly different outcome.

**Path dependence experiment** The first parameter to have its effect tested was the presence or absence of history in the run. The resulting network metrics are presented in Figure 6.3. We can observe two interesting things. First, the overall shapes are rather similar. The degrees

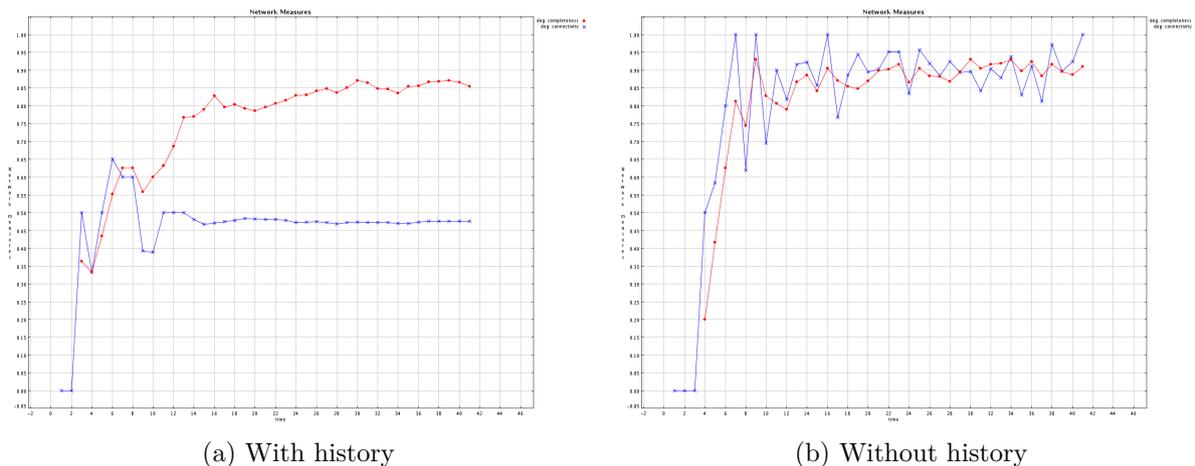


Figure 6.3: Network evolution depending on presence of history. Y axis: 0 ... 1, X : 0 ... 45. Degree of completeness is red, Degree of connectivity is blue.

of completeness, for example, stabilize at the same value. Second, the non-historic run is much 'noisier', and that the degree of connectivity widely oscillates over time. This makes sense from the perspective that the history-free run is effectively a time series of one-shot network structures, each sequentially created from a larger number of nodes, without any path dependence. Given the large noise in the non-historic run, all of the following experiments were performed with the network retaining its history.

**Randomness** Please note that the randomness in the model is *deterministic*. It is produced by a pseudo-random number algorithm (Hoschek, 2004) and is controlled by a given seed value, usually the current time in milliseconds. Furthermore, the order in which an agent interacts is determined by iteration over a list. To prevent order bias, the order of agents in this list is randomized at each time step.

**Diversity ratio experiments** After examining the model's path dependence, the diversity ratio between different agent types is examined. This is the ratio between the number of flow types and the number of possible in- and outflows of agents. All experiments were performed with path dependency turned on. There are three interesting situations worth examining.

The first situation has relatively few available flow types and complex processing capabilities. This corresponds to a refining and bulk chemicals situation, where a very large diversity of

<sup>4</sup><http://gux.tudelft.nl/svn/FlowBasedEvolution/tags/PhDThesisVersion/flowBasedEvolution.jar>

products is produced from crude. The third situation is the opposite extreme; the processing facilities are narrowly specialized, while there is a large availability of stream types to be processed. This corresponds to a situation in the specialties market, where there is a very large number of compounds available, and where a processing plant uses a few of them to create even more specialized compounds. The second situation represents a mix of the two extremes. The corresponding parameter settings are:

**Few Types and Many Flows** Few types of flows (5) and many in/outflows (15)

**Similar Types and Flows** Similar number of flow types (7) and many in/outflows (7)

**Many Types and Few Flows** Many types of flows (15) and few in/outflows (5)

**Linear network topology** First we will examine the effect of the diversity ratio on the network topology, see Figure 6.4. The graph is laid out using the Fruchterman-Rheingold (Fruchterman and Reingold, 1991) force-directed algorithm. Nodes that are more connected are plotted closer to each other than those less connected.

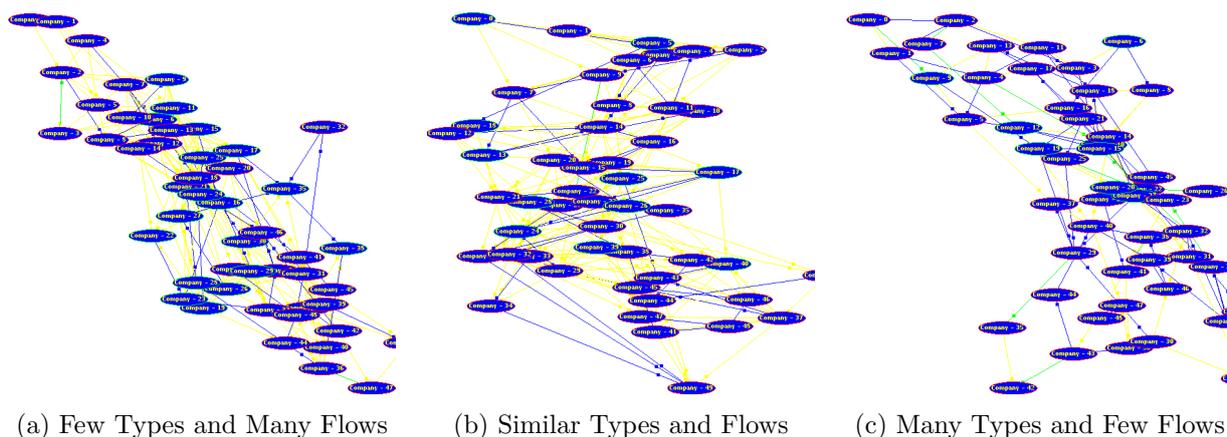


Figure 6.4: Network structure depends on diversity ratio. See text.

**Types of connections** In Figure 6.4 we can discern three types of edges between nodes. Green edges denotes a primary-primary flow connection, blue indicates a primary-secondary connection and yellow shows a secondary-secondary connection. The primary node is considered to be the 'main' or reference in- and outflow analog to the main feedstock and main product of a plant. Graph metrics do not discriminate between the flow types.

In the 'Few Types, Many Flows' and 'Similar Types and Flows' situations presented in Figures 6.4(a) and 6.4(b) we can observe that the majority of links are yellow, meaning that the network is mainly connected by secondary-secondary flows, auxiliary to auxiliary. In the 'Many Types, Few Flows' case, presented in Figure 6.4(c), the flows are mainly primary-primary or primary-secondary.

**Linear structure** It is interesting to note that in the first situation the network structure is linear, or chain-like. Due to the scarcity of types, all flow types are connected early on, and network growth can only continue when new nodes appear. Newcomers can not easily connect back to the main cluster. In the second situation the structure is much more interconnected, and in the third case the situation is similar to the first. Scarcity limits interconnection with the already present nodes, and a more linear structure is created. This can be demonstrated by examining the average shortest path length.

**Relatively long Dijkstra length** In Figure 6.5, the average shortest path length calculated using Dijkstra’s algorithm (Dijkstra, 1959) is presented. The path length is dependent on the diversity ratio. The Dijkstra length confirms the observation above. In the first situation we

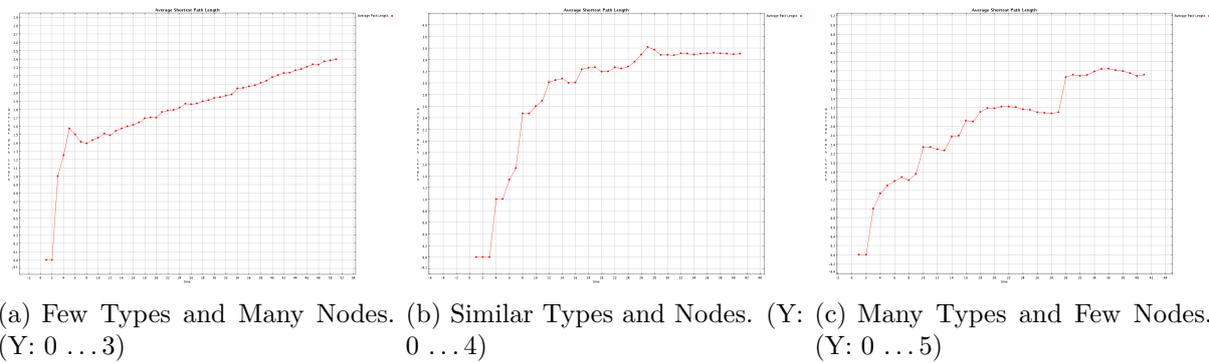


Figure 6.5: Average shortest path length depending on diversity ratio

can see the length monotonously rising with the number of added nodes. Addition of nodes makes the overall network diameter larger (see Section A.2.2). The second situation value stabilizes at around 3.5, meaning that the new nodes interconnect the existing network as they are added, and thus the network diameter is effectively the same. The third case again displays an increase in the relative size of the network, albeit in a different manner, since the growth is now limited by availability of nodes with new wanted inflow types.

**Not a scale-free network** Furthermore, it is interesting to note that the Dijkstra length has relatively large values, higher than 2.5. For a graph of such small size (50 nodes) this value is very large. Furthermore, when observing the degree distribution histograms in Figure 6.6, we can see that they do not follow a power law distribution.

Scale-free networks have degree distributions that follow power laws (Barabasi and Albert, 1999; Barabási et al., 2001) and relatively short Dijkstra lengths for their size. For comparison, an e-mail exchange scale-free network reported by Ebel et al. (2002) has a Dijkstra length of approximately 5 for a network of around 60,000 nodes. We can conclude that these graphs are not scale-free. The model’s designed inability of a node to connect to multiple other nodes makes scale-free networks impossible.

## 6.2.6 Domain-specific Insights

The lack of mass conservation and the way flows are connected between agents precludes insights about the functioning of real-world  $\lambda$ -systems. The model did, however, provide a wealth of

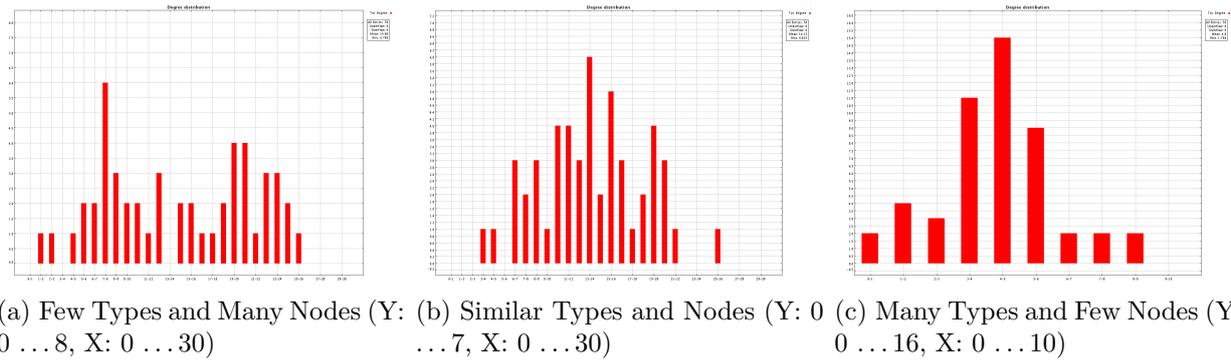


Figure 6.6: Degree distribution depending on diversity ratio

model development insights, which will be presented in the following subsection.

## 6.2.7 Method Development Conclusions

In this subsection we will discuss the lessons learned about the co-evolutionary model building method. We will start by revisiting the hypotheses and discussing the insights obtained from this first generation model. Finally, the modelling method and outcomes of the case study will be tested against the requirements defined in Chapter 5.

**Hypothesis** The hypothesis posed for this case study was: *It is possible to create a template for models that simulate industrial network growth, the template being based on agents that have a given number of in- and out flows of certain types and that connect to each other by locating outflows of agents that match their inflows.*

The hypothesis can be confirmed. The created agent abstraction works and allows us to build the model template relatively easily. It is indeed useful to view a processing industry as a network of materials processing nodes, evolving through node addition and edge reconnection. Using the fact-free approach, random agents can be created, exploring a wide design space of possible network structures, without the burden of data collection. While domain-specific insights are limited by the absence of closed mass balance, several important method insights were obtained:

- Starting the co-evolutionary method with technical model design works.
- The modelling approach used helps to determine which types of facts need to be collected.
- A modular description of technology and the economic decision-making processes is needed.
- Domain-specific metrics are needed in addition to generic network metrics.

**Start with the technical dimension** The co-evolutionary method could have been started with a design of a social process, or by knowledge and fact collection. However, choosing to use a starting point in the technical domain allowed us create a practical 'something with a run button' and allowed us to start asking practical questions, as we can examine the model and its output.

**Knowledge and facts** The modelling approach used can be seen as a formalization of the traditional process engineering perspective. Agents are abstract multi input / multi output mass processors. The model was developed fact-free. In other words, all the facts that are necessary to describe the agent are specified and it is assumed that no actual values for them are available. To run a simulation, facts have to be invented or randomly generated. This then gives the freedom to specify which types of facts are needed without actually having them, thus shaping the subsequent fact collection process.

**Modular implementation** The way agents are implemented in this model proved not to be very practical. For example, retrofitting a closed mass balance proved to be quite unwieldy. A better, more modular way to organize the simulation code is necessary. Modularity is also necessary to enable inclusion of other formalisms in the future. For example, business accounting and basic price-based decision-making is currently not included in the model. Furthermore, not only must different formalisms be modular, but formalisms themselves must be set up in the same fashion. For example, there are many ways to set a price, and they must be easily exchangeable.

**Graph metrics** The standard graph theoretical metrics were used to examine the evolved network properties. While they provide some interesting insights, such as a linear network growth and non-scale free properties, they are not adequate to describe network performance when the identities of flows matter. When the identities of edges and nodes are important, different metrics are called for, and those cannot be found in graph theory.

**Future challenges** Several future challenges were identified. A more complete and modular description of technology is needed, where the mass balances across agents form the physical limitation to agent behavior. Furthermore, a mechanism for encoding other (multi) formalisms, such as economic decision-making, is needed. This suggests a clear path for future work. First, a mechanism for fact and knowledge collection is needed that will involve stakeholders and collect relevant domain knowledge and facts. Second, network metrics that take node and edge identity into consideration are needed.

**Requirements checked** Table 6.1 presents an overview of the modelling requirements and the performance of the model.

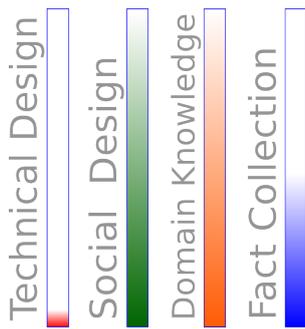
Table 6.1: Performance of the case study

Requirement	Score	Explanation
Method		
Open Source	Partly	A the time of writing the source code is available to peers, but not to the general public.
Sufficient community diversity	No	The model is developed as a first learning case, with the author as the only stakeholder.
Organically growing	No	The model is built as a conceptual test. While extension is kept in mind the current technical design does not allow for easy extension.
Recorded history	Yes	Versioning is initiated using CVS system. Due to technical problems the models early history is lost. More robust versioning needed.
Enforceable authorship	Yes	Personal accounts are used to track code commits.
Modular	No	As the model was a technology test, no modularity is implemented at this point.
Outcome		
Useful	Yes	The model is useful at the meta-level to the modeler.
Testable	Yes	During model development, model was not versioned, so repeatable testing of intermediate states is not possible. Experiments performed at the time of thesis writing are properly versioned, and can be repeated and tested.

## 6.3 Combination of Infrastructures

**Case abstract** The methodological focus of the second case study is on designing a social process for formalizing multiformal knowledge on spatial combinability of infrastructures in relation to social, legal, safety and technical aspects. In the social process a proto-ontology of infrastructures and their combinability is created. The facts collected allow the construction of a fitness landscape of combinations, the shape of which depends on scores assigned by stakeholders. The main insights for the model development method are that a multiformal knowledge and fact collection process involving a social network of stakeholders is operationalizable and practical. The developed ontology, while useful for the case study, is too rudimentary to hold more complex knowledge and facts and is impractical to extend to other knowledge domains. Technical implementation of the ontology must be made extendable and adaptable. The social process must be made more robust to path dependency.

### 6.3.1 Focus point



**Methodological focus** In this case study <sup>5</sup> the focus is on the design of the social process for knowledge formalization and on fact and knowledge collection. While this case study has ample technical model design aspects, they are not the focal point of the work presented here, as they do not directly concern an ABM.

The first learning case study concluded that there is a need for a knowledge and fact collection mechanism. If the models developed using the co-evolutionary modelling method are to be useful, they must sufficiently represent reality and they must be socially acceptable. Rather than collecting facts on a case-by-case, ad-hoc basis, we

seek to develop a social process for knowledge and fact collection that can be employed in a wide range of case studies and that yields results in a format that can be used directly in a computer algorithm and in social interaction processes.

**Case Focus** The previous case study focused on the nodes of an industrial network, the processing plants. This case study focuses on the infrastructure that connects them. It examines the factors that determine the spatial combinability of infrastructures from multiple disciplinary perspectives. Combinability is an aspect that plays a role in the evolution of spatially constrained  $\lambda$ -systems. The choice for this case study was jointly determined by the need to understand infrastructures better and by the practical opportunity offered by a project with the Rotterdam Port Authority (RPA).

---

<sup>5</sup>The Combination of Infrastructures case study was developed as a part of the project entitled “Combination of Infrastructures: A systematic exploration of (im)possibilities for space-saving infrastructures in the Rotterdam Harbor and industrial region” (Dijkema et al., 2007a). Parts of this section have been published as (Dijkema et al., 2007b).

### 6.3.2 Hypothesis

**Knowledge formalization** *Knowledge from many different disciplines can be formalized to enable fact collection and storage in a practical data collection and processing system. Continuous stakeholder involvement can be ensured by interfacing said system to models to generate simulations and feedback to stakeholders.*

### 6.3.3 Case Description

**Space scarcity** In port regions, infrastructures such as roads, rail, electricity, pipelines, docks, tunnels, etc. take up a considerable amount of space. At the time of a port's initial development, space may appear to be an abundant resource and the capacity requirement may appear adequately predicted. Over time, however, port activities and the infrastructure capacity to sustain them may evolve beyond expectations. When a region evolves under greenfield conditions, public port infrastructures compete with private enterprise for space. Under brownfield conditions, bottlenecks may develop where infrastructure congestion occurs at times of peak demand, while lack of space prohibits traditional infrastructure expansion.

**Port authority** To anticipate infrastructure congestion in the Rotterdam A15 infrastructure corridor, among others, the Port of Rotterdam has successfully implemented combinations of industrial pipelines in tunnels. While seemingly trivial, this approach is highly complex due to mixed ownership of different pipelines, safety and maintenance aspects. As a result, the question emerged as to why such spatial infrastructure combinations weren't used more often. In retrospect, the port organization implicitly assumed that ample potential might exist throughout the port to reduce the space allocated to infrastructures. The way to achieve this was assumed to be through spatial combinations, realizing more infrastructure capacity per acre.

**Systematic approach missing** Because of the very large number of infrastructures present and the myriad of factors influencing the spatial combinability of infrastructures, combining multiple infrastructures poses a great challenge for the port authority. No method existed to bring widely different combinability aspects together and systematically understand the design space. By design space we mean the complex socio-technical fitness landscape of possible combinations. In this space some combinations are safer, more feasible and more desirable than others. In the case where very many infrastructures exist, each having many different aspects, the design space of combinations is very large.

**CoI Project** In 2004, a joint TU Delft and RPA project, "Combination of Infrastructures" was started. The task at hand was to create a method for systematic exploration and assessment of the (im)possibilities of intense spatial combination of infrastructures. An 'infrastructure combination' was defined as a close spatial configuration of two or more infrastructures. These infrastructures were assumed to have a mutual influence on individual infrastructures' performance characteristics and/or allowed utilization. These aspects must be taken into account in port planning, infrastructure design, management and operation.

**Main assumption** The crux of the approach followed in this case study is the assumption that distinct, individual infrastructures have characteristics that are indicative of their combinability. Those characteristics are independent of the actual combination. The overall compatibility index for all possible infrastructure combinations with respect to safety, regulatory issues, spatial quality and the technical characteristics is a cumulative property.

**Social network** As infrastructures are complex systems, their combinations are complex as well. As discussed in Chapter 4, Complex Adaptive Systems can only be understood through multiple formalisms - and multiple formalisms require multiple people. As mentioned in the paragraph on methodological focus, the methodological goal of this case study is to develop a social process for knowledge and fact collection. In order to develop such a social process, a group of people involved - a social network - must exist. Such a social network will execute the developed social process and hopefully produce the desired knowledge and facts.

Therefore, a social network was initiated, consisting of the people with the following associations:

**Problem owner** The Rotterdam Port Authority (RPA), infrastructure planning department and business development department.

**Domain expert** Internal experts from the TU Delft, spanning the spatial planning, safety, legal and technical domains. External experts from the RPA, involved in actual planning, design and implementation of infrastructure projects.

**Modeller** Researchers from the TU Delft, including the author.

This social network includes a wide distribution of domain experts from both industry and academia. It expands the social network that is needed for a successful project.

**Social process** As discussed above, a social process for knowledge and fact collection is developed. It consists of 5 steps. Steps 2 to 5 are can be iterated as necessary to improve the quality of the outcome.

1. **Identify infrastructure** Relevant infrastructures need to be identified in an iterative social process.
2. **Define combinability** Landscapes and their aspects are defined and encoded in a data collection tool (start of the iterative loop).
3. **Collect data** The infrastructure combinability data are collected in a social consensus building process.
4. **Create landscape** The combination landscape (see next section) is calculated from the combinability data.
5. **Improve landscape** The initial design and data are presented to external domain experts and adapted based on their feedback (end of the iterative loop).

The following section will present the details of this process.

### 6.3.4 Details

**Combination fitness landscape** The main assumption of the CoI study is that each infrastructure has additive properties that describe its combinability, independent of the combination. So the combinability of a infrastructure can be seen as a certain combined 'height', or fitness, consisting of the height or fitness of individual aspects, see Figure 6.8. When combinations are examined in terms of pairs, they form a fitness landscape of points that represent the combined fitness of those two infrastructures. Figure 6.8 illustrates the combinability approach and process.

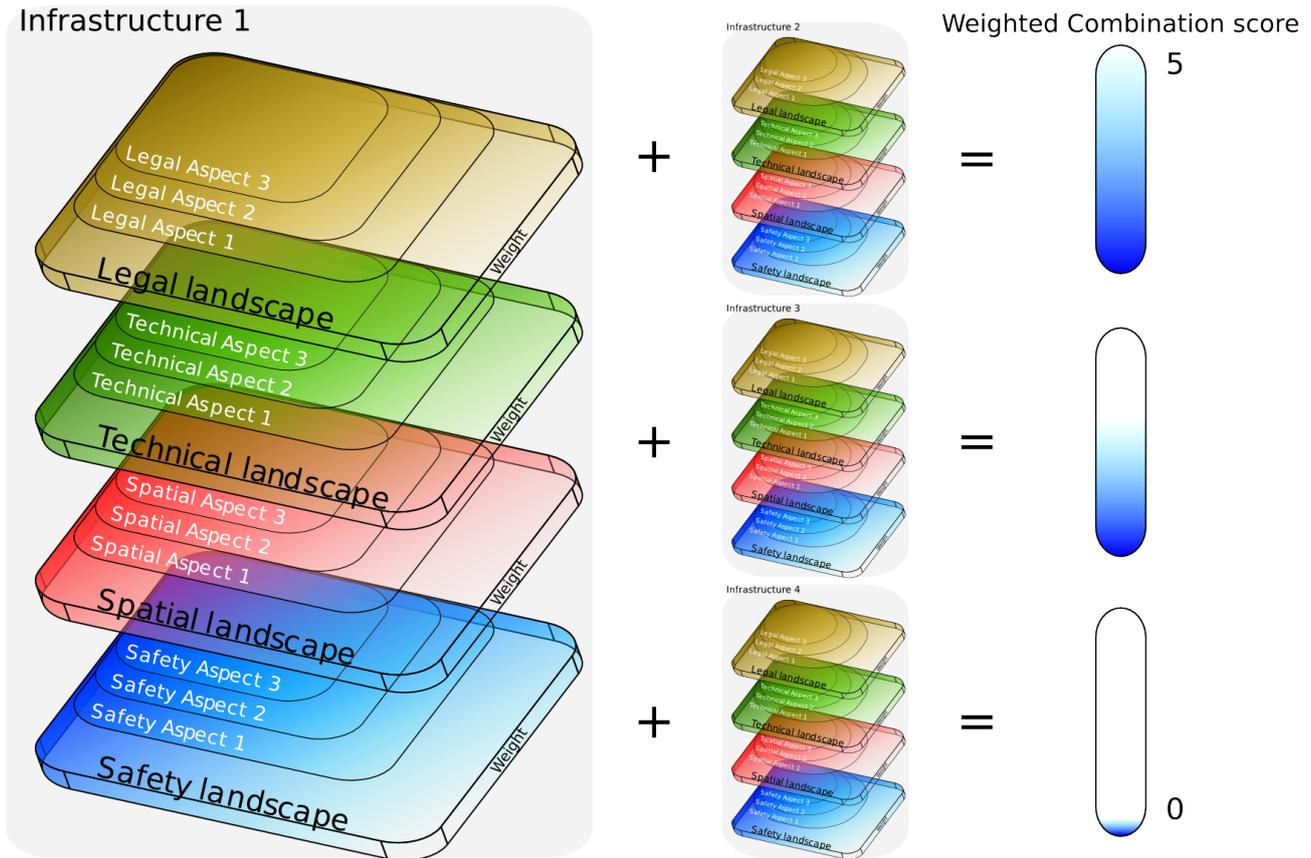


Figure 6.8: CoI fitness landscape creation process

**Species and the environment** When creating a fitness landscape, it is important to define the species and the environment. We consider the individual infrastructures to be species, whose interaction has a certain fitness. The fitness of a combination is determined by the environment. This environment consists of the social valuation of each infrastructure's combinability and the social valuation of the relative importance of each combinability aspect. The valuation consists partly of intrinsic physical aspects of the infrastructure and partly of the collective social valuation of it.

**Infrastructure categories** First, in a series of meetings and workshops, a structured inventory of port infrastructure was developed. This is depicted in Figure 6.8 as the gray infrastructure entities. Three infrastructure categories were identified:

**Traffic and transport** These are fixed infrastructures, such as roads and railroads, upon which a variety of discrete objects - vehicles - move between nodes such as container terminals, passenger terminals, seaports, airports, etc.

**Utilities** This category spans fixed infrastructures that are developed to transport one specific good over a dedicated infrastructure. Examples are electric power, drinking water, natural gas and telecom.

**Industry** This category only includes pipelines. Each and every pipeline is unique for the type of product it carries and the companies it connects.

These categories were found to span a total of 45 distinct infrastructure types. For the full list of infrastructures, please refer to Appendix C. This categorization is not definite or clear-cut. Industrial pipelines, for example, mostly carry chemicals but often also carry fuels, whereby they should become categorized as utilities. Despite these minor problems, this is a simple and usable categorization.

**Combinability** Second, based on the assumption that distinct combinability characteristics of infrastructures can be identified, this concept was elaborated in a series of brainstorming and expert meetings. This resulted in the identification of four infrastructure characteristic landscapes that are indicative of their combinability. Each landscape is a distinct formalism. They are:

**Safety** Not only distinct infrastructures, but also any combination, must be safe. For example, the amount of electromagnetic radiation given off by the infrastructure directly impacts its combinability.

**Spatial effect** Any combination can increase the use of space and be must acceptable with respect to spatial quality. For example, a highway needs to be routed with few sharp turns, making it easy to combine with rail.

**Technical** Specific technical aspects of an infrastructure may improve or reduce combinability. For example, wireless communication towers need relatively rare and easy maintenance and are thus easy to combine.

**Legal and organizational** Each distinct infrastructure has its own legal and regulatory regime. For example, EU regulation on road development and the complex land ownership structure in the Netherlands make roads somewhat difficult to combine with other infrastructure.

These landscapes are presented as the colored layers in Figure 6.8. Each of these landscapes contains a number of aspects that give the landscape its final shape. This classification essentially defines an ontology, a formal specification of a conceptualization. (See Section 6.5)

**Data collection** Third, once the landscapes and their aspects are identified and the structure of the ontology defined, the factual information can be added. Collection of information requires domain experts with a variety of backgrounds. Experts must score the aspects of each landscape and establish the relative weights. The structure of the ontology and the data collected are a codification of expert knowledge that can be processed to generate a fitness landscape for all possible infrastructure combinations.

**Fitness landscape creation** Fourth, a model for evaluation of the infrastructure combinations' fitness was developed. This involved two steps:

**Scoring infrastructure's characteristics** A scale running from 0 to 5 was used, where 0 indicated a 'deal breaker', meaning no possible combinability for the distinct infrastructure, and 5 indicated a 'deal maker', a highly advantageous combinability.

**Relative weight** The relative weight of each characteristic was established, running from 0 to 1.

**Calculation** Calculating a weighed score over all characteristics was done by inputting the information from the two previous steps in a spreadsheet and using a suitable matrix manipulation package to calculate the weighed scores of all 2025 infrastructure combinations and map the results (see algorithm 6.3.4).

**Initial social process** Internal experts created and filled the initial ontology during a group brainstorm session. Ideally, this would have been performed with both internal and external experts. For practical reasons the internal experts were used to pre-seed the discussions with the external experts and thus reduce the required time.

**Fitness landscape calculation** Even though this case is not focused on technical design, we still need a way to model the fitness landscape. The fitness landscape calculation method is presented in algorithm 6.3.4. The full source is available online <sup>6</sup>. This algorithm creates a fitness map of combined pairs of infrastructure. The algorithm can be adapted to combine higher order pairings.

---

**Algorithm 6.2** Simplified fitness landscape calculation

---

```
x ← numinfra
y ← numlandscape_aspects
score ← ones(x, x)
environment(x, y) ← excelsheet
for a = 1 to x do
  for b = 1 to x do
    for c = 1 to y do
      scorea,b = scorea,b * environmenta,c * environmentb,c
    end for
    scorea,b =  $\sqrt[2]{score_{a,b}}$ 
  end for
end for
```

---

**Collaboration via wiki** The initial steps of the social process were facilitated through a collaborative wiki system <sup>7</sup>. The main reasons for introducing the wiki system were that the expert group was fairly large and the logistics of organizing of brainstorming sessions and

---

<sup>6</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/code/CVI/cvi/cvi\\_2d.m](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/code/CVI/cvi/cvi_2d.m)

<sup>7</sup><http://wiki.tudelft.nl/Project/CombinationOfInfrastructuresEvaluation>

meetings was not trivial. Furthermore, wikis offer enforceable authorship and a historic record of the communication across it <sup>8</sup>. The wiki proved mainly useful for the internal expert meetings. The extremely open and free-form collaboration environment that wikis provide was found to be unsuitable for external expert involvement.

Two main barriers to wiki adoption were identified. The first barrier was the reluctance of the already very busy experts to learn a new tool. The second, more serious, barrier was the necessary shift in mindset when using a wiki. Traditionally, knowledge is perceived as power, and knowledge shared is perceived as power lost <sup>9</sup>. A more bottom-up, collaborative and social network-oriented mindset is required for wiki-style collaboration, where one realizes that the more knowledge is shared or generated by a person, the more 'power' that person has.

**Group Decision Room** To overcome the experts' problems with the wiki, a Group Decision Room (GDR) (Kolfshoten et al., 2006) environment was used for external expert feedback meetings. GDR offers a simple, anonymous and structured feedback mechanism. Guided by specific questions, the experts can share their knowledge quickly and safely. The perceived safety is mainly an effect of the system's anonymity, allowing contended topics to be discussed without personal consequences. Figure 6.9 gives an impression of the activities during the workshop.



Figure 6.9: Data collection and feedback workshop

**Feedback process** The GDR-facilitated feedback process used the following guiding questions:

1. Introduction to the Combination of Infrastructures project.

---

<sup>8</sup>See Section 5.4 for requirements of the modelling process)

<sup>9</sup>For wiki discussion on this very topic, please see <http://wiki.tudelft.nl/Research/WhyPeopleDislikeWiki>

2. Explanation of GDR and a test round. “How to make Rotterdam the biggest harbor in the world again?”
3. GDR round 1a, Inventorying: “What are the relevant infrastructures?” (pre-seeded with categories)
4. GDR round 1b, Free Brainstorm: “What determines the combinability of infrastructures?”
5. GDR round 1c, Vote (1-5): “What are the deal breakers and deal makers for combinability of infrastructures?”
6. Consensus-building: compare generated list with project-team generated ones.
7. GDR round 2: Relative weighing of landscapes and aspects.
8. Scoring of individual infrastructures.
9. Discussion of the results and their significance for the RPA policy.

This feedback process resulted in a refined scoring of the combinability.

### 6.3.5 Case Study Results

**Social process** This was a first attempt at a design of a social process that brings multiple formalisms together within one formal (software) system. The two-part process with internal and external experts has both advantages and disadvantages. The advantage is mainly practical. Pre-structuring the ontology saves time in the discussions with the external experts, whose time is often very limited. The negative aspect is that the external experts do not feel involved in the design and do not feel ownership over the outcomes. Furthermore, quite some time was spent explaining the relatively complex logic behind the approach and the ontology.

**Ontology** Figure 6.10 presents an excerpt from the matrix-based ontology and the spreadsheet based data input form.

The complete ontology developed by the project group can be found in Appendix B.

**Fitness landscape evolution** As presented in Sections 3.4 and 5.2.2, the shape of a fitness landscape changes as interacting species adapt and evolve. We can observe this process of co-evolution between qualitative and quantitative insights and social consensus. Figure 6.11 presents the infrastructure combination fitness landscape before the involvement of external experts and after.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1																				
2		Type infrastructuur	Waalhaven	BeoordelingsCriterium																
3				Betekenis NB score 5 = showsteler (meest gunstig voor (combinatie van) aanleg infra; score 1 = showstopper																
4				Veiligheid								Juridisch-Organisatorisch				Ruimtelijk				
5				Kernische energie	blits, rook en/of warmtestraling	gevaarlijke stoffen	water (overstroming, verdrinking etc.)	elektriteit en/of elektromagnetische signalen	ontsmikbaarheid	veiligheidsgevoel	EGENDOM VAN INFRA?	SECTOR GEREGULEERD?	CONCURRENTIE OP INFRA?	VERGUNNINGEN VEREIST?	PROCEDURE-TIJDEN	LOBBYERBAAR HEID	REGELGEVING	Ruimtedoorvaart / (barricadeerend / versnipperend)	Ruimteslag	Stapbaarheid naar BUREAUS
7				1=zeer veelhoog	1=zeer veelhoog	1=zeer veelhoog	1=zeer veelhoog	1=zeer veelhoog	1=zeer slecht	1=zeer negatief	1=versnipperd	1=zeer specifiek	1=Volledig	1=Veel	1= Zeer Lang	4= Geen Speedrun?	1=zeer groot	1=zeer groot	2=alleen maar?	1=st
8				2=veelhoog	2=veelhoog	2=veelhoog	2=veelhoog	2=veelhoog	2=slecht	2=negatief	2=gr	2=specifiek	2=Volledig	2=Veel	2= Zeer Lang	3= Geen Speedrun?	2=groot	2=groot	3=half verdept	2=st
9				3=weiniglaag	3=weiniglaag	3=weiniglaag	3=weiniglaag	3=weiniglaag	3=redelijkgoed	3=positief	3=redelijkgoed	3=specifiek	3=Volledig	3=Veel	3= Zeer Lang	4= Geen Speedrun?	3=groot	3=groot	4=verdept	3=st
10				4=zeer weiniglaag	4=zeer weiniglaag	4=zeer weiniglaag	4=zeer weiniglaag	4=zeer weiniglaag	4=zeer goed	4=zeer positief	4=zeer goed	4=specifiek	4=Volledig	4=Veel	4= Zeer Lang	5= Geen Speedrun?	4=groot	4=groot	5=aanne	4=st
11				5=geen	5=geen	5=geen	5=geen	5=geen	5=geen issue	5=net bevoor	5=in een hand	5= niet	5=Geen	5=Geen	5= Heel Kort	6= Veel Speedrun?	5=zeer klein	5=zeer klein	6=op onderun?	5=st
12																				
13				n respondenten	3						n respondenten	3					n respondenten	4		
14		1 Lokale Weg	ja	3.7	3.7	3.7	5.0	4.7	3.7	2.7	5.0	4.0	3.7	3.0	3.7	3.3	2.8	2.8	3.0	
15				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
16		2 Ontsluitingsweg	ja	3.7	3.7	3.3	5.0	4.7	4.0	2.7	5.0	4.0	3.7	3.0	3.3	3.0	2.3	2.3	3.5	
17				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
18		3 Snelweg	nee	2.7	3.3	3.0	5.0	4.7	3.7	2.0	5.0	4.0	3.7	3.0	1.3	1.7	1.3	1.8	4.3	
19				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
20		4 Wegtunnel	ja	2.7	3.3	3.0	4.7	4.3	2.0	1.3	5.0	3.7	3.7	2.3	2.0	1.7	2.8	2.8	4.3	
21				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
22		5 Weg op brug	ja	3.0	3.7	3.3	5.0	4.3	2.7	2.0	5.0	3.7	3.7	2.3	1.7	1.7	2.5	2.5	1.0	
23				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
24		6 Parkeerterrein	ja	4.3	4.0	3.7	5.0	4.7	3.3	2.7	3.7	4.0	3.7	3.0	3.7	3.8	2.8	4.0		
25				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
26		7 Bushalte	ja	4.3	4.7	5.0	5.0	4.7	3.7	2.7	5.0	3.7	3.7	3.0	5.0	3.7	4.8	4.8	2.8	

Figure 6.10: Excerpt of the matrix-based ontology

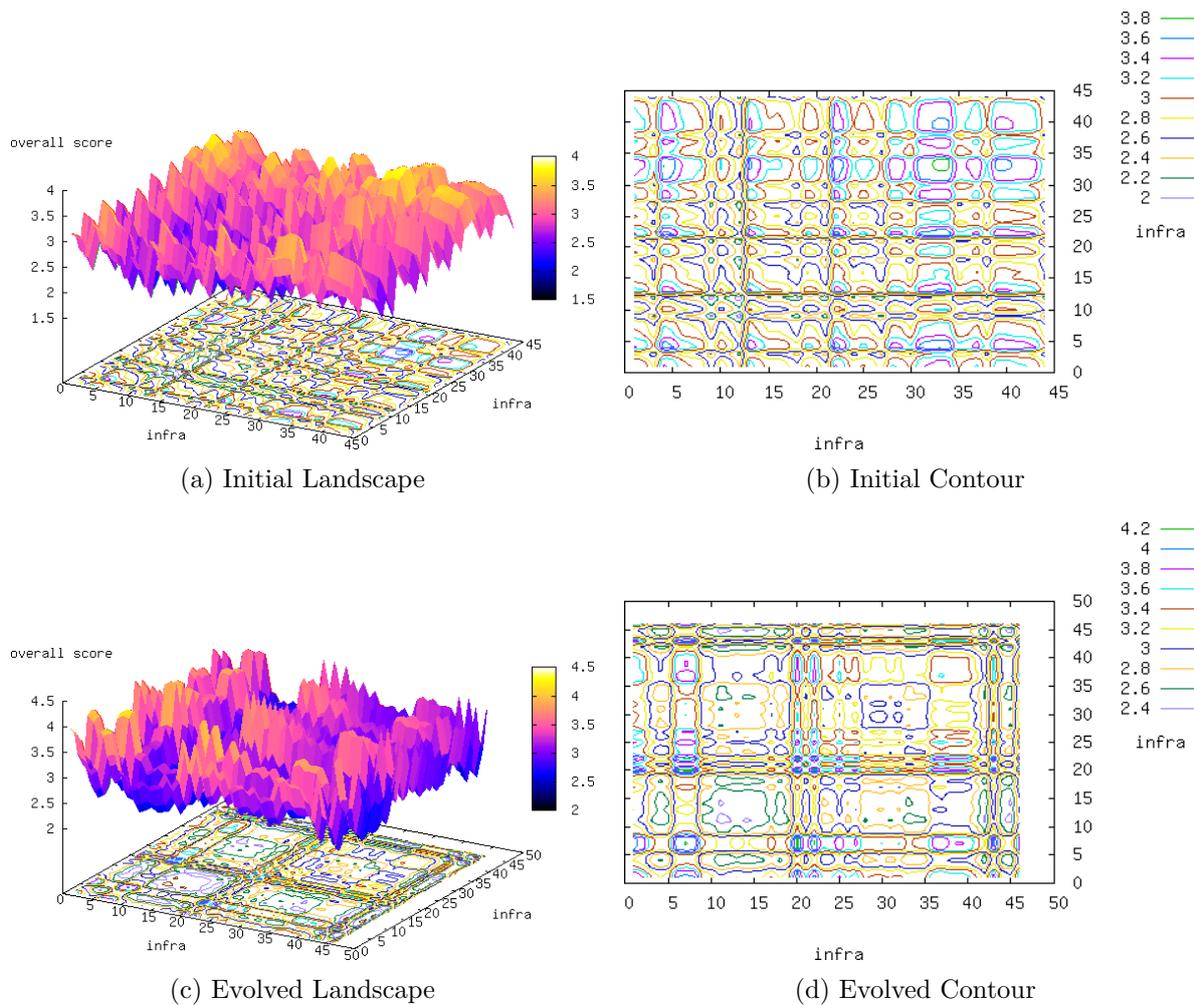


Figure 6.11: Fitness landscapes for all infrastructure combinations, scored by the project team and by the RPA team.

It is interesting to note that the internal experts are less discriminating than the external experts. The second landscape is much more rugged and shows larger variation between the worst and best combinations. This can be attributed to the external experts' greater practical experience with the infrastructures in question and the academics' tendency to relativize their own work.

### 6.3.6 Domain-specific Insights

The focus of this case study was not on the domain insights. For the insights in the combinability of infrastructures, the reader is referred to the project report (Dijkema et al., 2007a). However, there are several domain insights that can be presented here.

Due to the social process and the knowledge collected, the RPA has for the first time a systematic, socially constructed overview of combinability of infrastructures. Up to now, combinability issues were examined on a case by case basis, without organization-wide input. After the CoI project, the RPA had a clear sense of what can and cannot be combined, based on expertise from many different parts of the organization. The list provided no surprises in terms of new combinations. This is a very positive result for the RPA, as it shows that the implicit, unsystematic 'gut feelings' of the involved experts functioned relatively well. The created fitness landscape offers a basis for future decisions and ongoing discussions when the social, legal, safety and technical environment changes.

### 6.3.7 Method Development Conclusions

**Fact and knowledge collection successful** Referring back to the case hypothesis, it can be concluded that the multi-formal knowledge and fact collection process is possible and functional. The social network was able to use the social process to collect knowledge on combinability (the infrastructure classification and the combinability aspects) and facts on combinability (the actual infrastructures and their scores). The knowledge and facts are multiformal, covering legal, spatial, safety and technical domains. Their knowledge and facts are at the same time available for computer processing.

**Crucial role of the social process** The social process facilitating the ontology creation and fact collection is crucial. A lot of effort has to be given to designing a robust process that is shared by all involved parties. The conditions under which the social network is created and operates are as important as the knowledge of the experts. The ontology pre-structuring needs to be performed carefully in order to prevent too much steering by the external experts and to increase the stakeholders buy-in.

The main lesson learned is that the predictability and pace of the process is important to the stakeholders. The process as executed during the case study contained too many surprises. First, the process progressed too slowly and in a fashion that was intransparent to the external experts, as there were many theoretical issues that had to be solved by the internal expert team. Later, the process progressed too quickly, as the conceptual width and depth was too great and the technical tools employed were too novel for the external experts.

Furthermore, the transition from no shared language to the shared formal language was too sudden. Because the internal experts developed a classification of combinability (effectively a language) and presented it to external experts, the external experts went in one step from from

an unshared and unformalized state to a shared, formalized state. This step was conceptually too large for the external experts and did not create sufficient buy-in. See Section 6.5 for a discussion on knowledge states. Depending on the goals of the project, this may or may not be a problem.

**Structure of knowledge** In the current generation of the knowledge formalization process, the employed ontology and the facts collected are 'flat'. It only encodes *is a* knowledge of the  $x=1, y=2$  structure. In this case study, this was sufficient. However, it is easy to see that more complex situations would need to have an ontology that is able to capture the problem's structure as well as facts about it, to include both *is a* and *has a* types of relationships. While not strictly impossible with a spreadsheet implementation, it is exceedingly impractical. Restructuring the ontology to accommodate expert feedback is equally complicated. The tool for knowledge and fact encoding is too rigid and does not allow for easy change. Furthermore, in this case study the knowledge base had no versioning applied. This makes future changes very difficult to manage.

**Further development** Two main directions should be followed in the future. The current technical implementation of ontology creation and fact collection is inadequate and must be made more robust, more extensible and more adaptable. The social process must be more carefully designed and adapted to yield information on the system structure.

**Requirements checked** Table 6.2 presents an overview of the modelling requirements and the performance of the model.

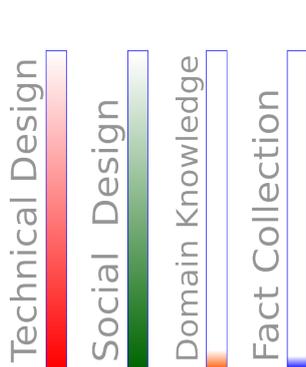
Table 6.2: Performance of the case study

Requirement	Score	Explanation
Method		
Open Source	Partly	The source code of the calculation method is available to peers, but not to the general public. Social process has been published.
Sufficient community diversity	Yes	Expert and user diversity was sufficient. Quality of community was lacking.
Organically growing	Partly	Project requires external parties to proceed.
Recorded history	No	Attempted and failed.
Enforceable authorship	No	Attempted and failed.
Modular	Partly	The description of infrastructures and their landscapes and aspects is probably portable to other projects.
Outcome		
Useful	Yes	Useful both at the process level to the modeler, as well at the case level to the problem owner.
Testable	Yes	Social experiments are possible, observation of shifting insights and knowledge directly readable from the fitness landscapes.

## 6.4 Chocolate Game Model

**Case abstract** The third learning case study presents a generic ABM of the process industry. The model is based on the system decomposition of a chemical process industry sector. The resulting system description was translated to a suitable analogy: a production chain for chocolate bars. Using this analogy, a game was developed and played to elucidate and explain complexities and interdependencies in the corresponding industrial network. The concepts extracted during the system decomposition and the development of the game were formalized in an ontology (a specification of concepts). This ontology served as the foundation of the representation of reasoning, communications and transactions in our multi-agent system. A prototype generic multi-agent model was implemented and developed in Repast to serve as a simulation engine for real process industries and other applications. The main learning points from the case study are in the technical model design dimension. The abstraction of discrete flows will be reimplemented as continuous flows, and a more flexible ontology encoding tool will be sought. The social process was found to be useful and will be further refined in the next generation.

### 6.4.1 Focus Point



**Methodological focus** The Chocolate Game case study focus is on social process development and technical implementation<sup>10</sup>. We implemented the lessons learned from the previous two case studies in this case. First, the technical model design needed to be made complete with closed mass balances and a modular setup separating the technology description and economic aspects. Agents needed to be able to process discrete flows (streams of pieces), make purchasing contracts and make pricing decisions. Second, the social process for knowledge and fact collection needed to be redesigned in the light of experience from the previous case study.

Figure 6.12: Case focus

**Case focus** The practical aspects of the case study involved developing a chocolate production network as an analogy of a processing industrial network. The analogy was examined and decomposed in a formal ontology. The ontology was used to create a conceptual description of the system that was used to create a serious game. This chocolate game is a paper-based business game designed to examine actor system dynamics. To this end, playing the game requires human players. Their actions, negotiations and strategic behavior are recorded to allow ex-post interpretation so that these can be formalized into a model of the actors involved in real world industry. Completing this formalization also serves as a process wherein the aptness and completeness of the ontology is verified. The players' behavior is observed and translated into agent rules. Together with the ontology, this makes it possible to create an ABM with which to formally examine behavior of the game.

<sup>10</sup>Parts of this section have been published as "Towards a Generic Approach for Analyzing the Efficiency of Complex Networks" by K.H. van Dam, I. Nikolic, Z. Lukszo and G.P.J. Dijkema (K.H. van Dam and Dijkema, 2006).

## 6.4.2 Hypotheses

**System decomposition** *It is possible to develop a social process script that will extract and structure domain experts' knowledge in a manner suitable for creating an agent-based model of network evolution.*

**Game as a template** *It is possible to use a serious game as a template of an industrial network and use it to test the decomposition created by the SDM.*

**Discrete flow processing** *It is possible to construct an ABM that describes a industrial network processing discrete flows.*

## 6.4.3 Case description

**Games and models** The basic idea of this case study is that a combination of a system decomposition method, a serious game and an ABM is a useful tool for analyzing the emergent properties of industrial networks. As discussed in Chapter 4, Agent Based Models are Complex Adaptive Systems that display emergent properties. Serious Games (SGs) (Mayer and Veeneman, 2002) can be understood as Agent Based Models that employ human players instead of agents to generate emergent behavior. Applying both ABMs and SGs to the same system provides us with insights into system behavior that would not be possible with just one tool alone. While SGs are not strictly formalized and deterministic as computer models, they are able to display a broader range of possible behaviors, as humans are rather inventive when dealing with game rules. ABMs are more constrained in behavior but can be rerun thousands of times with different parameter settings, providing a broad view of possible system states.

**Approach** The case study was approached as follows: a system decomposition of a chemical process industry was performed, and an analogy was created in a different domain - a process chain that produces chocolate bars. With this analogy in mind, a Serious Game was developed that can be used to explore and explain the complexities and interdependencies of an industrial network to players/stakeholders. Using insights from the game and the system decomposition results, an ABM can be constructed to explore the systems behavior under a wide range of conditions. The different elements of the approach will be discussed in the following paragraphs.

**System Decomposition Method** Before we can start modelling a system, we need to describe it. We need to create a system description consistent with the systems perspective and allows for a generativist system description (see Section 2.1.3). The method of creating this description is the System Decomposition Method (SDM). It is a collaborative method in which system components are identified and formalized (Dijkema, 2004; Dijkema et al., 2005; K.H. van Dam and Dijkema, 2006). All important concepts (as chosen by the experts and the modellers), properties and interactions of the chemical process industry are formalized during the method in a formal ontology. Before they are used to create a model, the identified system elements are translated into the domain of chocolate production.

**Technical Analogy** The goal of the case was to study system behavior without being bogged down with details. In order to help domain expert abstract away from the details and to ease the understanding of the overall system behavior, we decided to use an analogy of the petrochemical processing industry. An analogy captures the essence of the petrochemical supply network and chemical conversion processes while reducing detail. Instead of using oil and other chemical feeds as a main input of the petrochemical production chain, we chose to use raw chocolate beans. Instead of ethylene, processed raisins are used; and instead of plastics and other end products, raisin and peanut chocolate bars are produced by the processes. The abstract supply chain of chocolate bar production is the analogy used for petrochemical production.

**Serious Game** The system description resulting from the SDM, after being translated into the analogy, is used to create the concepts of the game. These concepts are used for reasoning and communication in the game. Before a formal computer model is created, all game rules, agent identities, interaction types, etc. are tested in a Serious Game. By creating and playing the game first, the flexibility of human players is used to 'debug' the game by spotting inconsistencies in the system description and game rules.

**Agent Based Model** Once the game has been played and debugged, the game concepts are formalized in an Agent Based Model. Observed player reasoning processes are simplified where necessary and encoded as decision-making rules for the agents. The simulation is examined across different parameter settings to explore possible system states. Based on the insights from the game and simulation, conclusions about the behavior of the petrochemical supply network can be made.

## 6.4.4 Details

### 6.4.4.1 System Decomposition Method

**Main steps** This section describes the practical steps of the System Decomposition Method for evolutionary analysis of complex systems. The theoretical base of the SDM is elaborated in Section 6.5. It is based on an extended, improved and generalized version of the method reported earlier (Nikolic and Dijkema, 2005). The goal is to arrive at a representation of a problem space that enables the analysis of the system's evolutionary patterns.

This means that the system is not only considered as a collection of actors and interactions that exist in the current system configuration, but that the system representation must also allow us to take into account which components might change over time, with system level evolution as a result. The SDM consists of three phases: inventory, structuring and formalization into an ontology. These three phases will be discussed in the following paragraphs.

**Inventory** The aim of the inventory phase is to determine the system boundary and identify the system components relevant for the problem at hand. Given the complexity and scope of problems addressed (see 1), such an inventory cannot be expected to be immediately complete or consistent. In the method, inventory will be readdressed and improved during the iteration step of the structuring phase (see below). The following steps are taken to obtain the inventory:

1. Choose a system to be observed.

2. Define the problem owner and its problems, and determine the questions that the problem owner has.
3. Choose a time frame relevant to the system and problem owner.
4. System and problem space exploration. Domain experts are interviewed, and a structured brainstorm challenges them to explore the system structure and the problem owner's questions. The experts talk about the system in relation to the problem and make an inventory of all concepts, actors, objects, interactions, states, properties, flows, etc. relevant for the problem analysis.

**Structuring** The aim of the structuring phase is to create structure of the identified components, thereby creating a map of the real-world system under consideration. The structuring phase is the core of the system decomposition. The components identified in the inventory phase are first grouped into agents or objects, and the connections between those agents and objects are grouped by interactions. After identification, the objects and agents are linked using the identified interactions.

1. Structuring of agents and interactions: Agents are the units upon which evolution acts. They are recognized by their boundaries. Agents are active, proactive, reactive and able to interact. Objects are entities that have clear boundaries and can interact but are not active or reactive. Interactions are system components that connect agents to objects. The structuring steps are as follows:
  - (a) Look for useful boundaries: physical, organizational and/or functional. Identities identified in this manner are the agents or objects.
  - (b) Within the agents/objects thus identified, search for properties or behaviors that interact with components outside of those boundaries. Within the same boundaries, search for behavior or properties that propagate from these entities and interact with the world around them. These can be codified as interactions.
  - (c) Add hierarchy to the agent components by ordering them in a hierarchical, nested way, as a box within a box.
  - (d) Add hierarchy to the interaction components by classifying them from abstract to concrete.
2. Linking agents and interactions: In this step a connection is made between the agents and interactions so far identified.
  - (a) First only consider the highest level agents.
  - (b) Draw the agents as a box.
  - (c) Add the most abstract interactions as arrows going in and/or out of these boxes.
  - (d) Do not yet connect incoming and outgoing interactions between boxes; this will happen in the simulation of the system. Network structure will follow from behavior.
  - (e) Repeat these steps for lower levels of the agents and interactions hierarchy.

3. Iteration: In the iteration step, one may choose what to include or delete from the initial inventory and what subsequent modifications to structuring this may lead to. Careful review of the results achieved and the original information and knowledge may reveal missing components, agents, interactions or characteristics thereof.
  - (a) Carefully check whether all concepts identified in the inventory phase have been included. If not, complete a new iteration of structuring as follows:
    - i. Decide if any of these should be in the system description.
    - ii. Simplify as much as possible, but record the simplifications made. During the use of the SDM outcomes, users may notice that some essential concepts are missing. In that case, the simplification needs to be reverted.
  - (b) If an agent has no ingoing and/or outgoing interactions, consider whether interactions are missing. Not all agents necessarily need to have both in and out flows, but if an agent has no interactions at all linked to it, it must be considered whether this agent really plays a role.
  - (c) If an interaction is not connected to any agent, decide whether the interaction is relevant, meaning that the agent affected by it is missing, or whether the interactions is superfluous.
4. External world: In this step the world outside the agents is determined by grouping all the system components that cannot be influenced by the other subcomponents. They form the External World.
  - (a) Things that are not influenced by components within the system are part of the External World.
  - (b) Extremely slow processes (relative to the chosen time frame) should be considered to be a part of the External World.

**Formalization into a ontology** The goal of the formalization step is to encode all of the output of the previous two steps. After the inventory and structuring phases of the System Decomposition have been completed, a formal ontology can be created in order to formalize the domain and to enable the system description to be implemented as a model. A body of formally represented knowledge is based on a conceptualization that “the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them” (Genesereth and Nilsson, 1987). An ontology consists of one or more classes and their properties. Each class has a number of properties - characteristics that are specific for this class. Each property is of a certain type that defines how the data for this property should be stored. The formal representation serves as a semantic and functional design description for the development of a simulation model. Software agents use the ontology to exchange messages about certain subjects, and they understand how the concepts in the content of the messages are to be used.

#### 6.4.4.2 Chocolate Game

**Overview of the Chocolate Game** One of the powers of serious games lies in analyzing, experiencing and creating awareness about complex problems (Duke and Geurts, 2004). To

make the game players experience the complexity of the system and to create awareness that system-level thinking increases sustainability, they will be acting as individuals, representing an industrial plant or company. The game leader manages the evolution of infrastructural systems in the External World, in cases where there is a requirement to do so. The actors represented by the players may be hierarchically organized, with explicit cooperation or competition goals.

**Roles** In the Chocolate Game, different player roles are identified: producers of half products, producers of end products, transporters and a world market. The world market is controlled by the game leader and can be used to steer the behavior of the players by changing the prices of products. These roles will be discussed below.

**Producers** Producers can, of course, produce goods from raw materials, be it final products from half products, or half products from unrefined goods. They have a certain technology that allows them to turn products into another type of product (e.g., making a bar of Raisin Dream out of Processed Dark Chocolate and Processed Raisins). All producers can buy any product from the world market and also sell anything to the world market. Producers can also trade products with other producers. This is where part of the fun starts, because in this case contracts and conditions must be negotiated as well as the price. Conditions may concern the duration of the contract and what happens if one of the players does not stick to his or her part of the deal. In the game, players were free to come up with their own conditions as long as they were described using the terminology expressed in and formalized in the game rules.

**Transporters** Transport players are responsible for transporting goods from one player to another and for transporting from and to the world market. In the game, contracts have to be negotiated, as well as a necessary condition to ensure delivery. An important game rule is that no product can leave the tables except in the transport unit of a transport player. That said, transporters are also free to create their own contract conditions as long as they describe them clearly and formally.

**Fun** One of the most important concepts of the game was the fun factor; this was another - if not the main - reason for using chocolate production as an analogy for chemicals production. Players have to enjoy themselves to feel involved in the game (Mayer and Veeneman, 2002). In the Chocolate Game, real ingredients are used that have to be processed by the players. A player buying a batch of 'raw peanuts', for example, receives peanuts that still have to be peeled in the production step, resulting in a number of 'processed peanuts'. The waste resulting from the processing is also a good that must be transported to the garbage bin by the Transporter.

**Game details** The details of the rules and concepts of the Chocolate Game are beyond the scope of this section. The entire set of rules, the game plan and the description of the game materials, as well as impressions of the game sessions are available online <sup>11</sup>.

---

<sup>11</sup><http://wiki.tudelft.nl/Project/ChocolateGame>

### 6.4.4.3 Agent-based Model

**Chocolate Game Model** The Chocolate Game serves as an informal model of the processing network system. Once the SDM is performed and the game is developed and played, a formal model can be developed. With a clear definition of the mechanics of the game established and a formalized description captured in the ontology, we have a good foundation for implementing the chocolate production network in an ABM.

**Model assumptions** The model was developed under three main assumptions.

**No transport agent** The first assumption was that transport agents are absent. This greatly simplified the modelling of contracts, as only bilateral contract needed to be created, and not a tripartite one. The focus was on the negotiations between the producers and the network that results from the decision-making process between them.

**Random prices** The second assumption was made in the modelling of the decision making: each time an agent was asked to sell a certain good, it would simply ask a random price for it from a predetermined range. The model, however, is set up in such a way that we can later easily add more realistic price setting behavior.

**Discrete flows** The final assumption was that the flows between agents are discrete. That is, a flow consists of  $x$  pieces per time steps. This means that the agents need to have a working and limited storage system that stores the correct types of goods.

**Agent reasoning** Agents not only use the ontology in the communication, but the meaning that is defined in the ontology is used to deduce which goods are needed to produce a certain good. An agent starts each turn by examining its technology object, which determines the product that is made. From the formal description of this class, the agent can find out what product is needed in the production step. After that it will look in its own warehouse to see if a product of this class is already in stock or if it needs to do market research to find the best offer. From the one or more offers that the agent receives, it will pick the cheapest one and sign a contract with this party.

**Implementation** With this model, implemented in Repast (N. Collier and North, 2003), a number of experiments were run, varying the number of agents equipped with the technology to produce a certain good. In the initial loading the World Market had a batch of raw products, while the other agents had nothing in stock yet. We kept track of the average price paid for a certain product class (i.e., raw products, half products and end products).

## 6.4.5 Case Study Results

In this section the results from the three parts of the approach - the SDM, serious game and the ABM - will be presented.

### 6.4.5.1 SDM Ontology

The main outcome of the SDM is a system description formalized into a ontology.

**Created ontology** The ontology created during this case study is presented in Figure 6.13.<sup>12</sup> It consists of a general ontology describing the things that agents act upon and an agent ontology defining the agents and their actions. The ontology is implemented as a Java class hierarchy.<sup>13</sup>

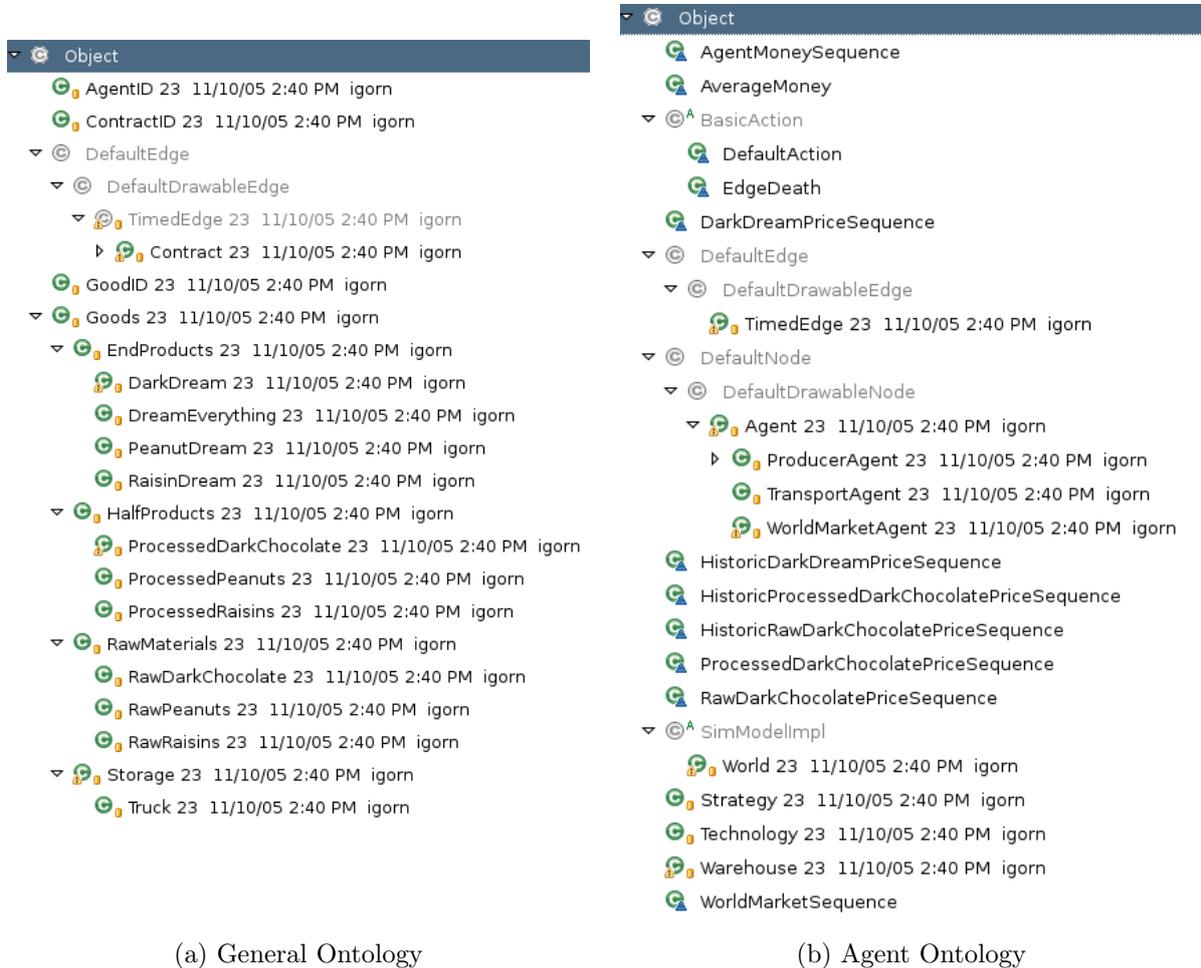


Figure 6.13: Structure of the Chocolate Game Model Ontology

**Verification of the ontology** After the system decomposition has been completed and the concepts have been formalized in an ontology, the next step is to verify its completeness and usefulness. Ontologies, being languages, are meant to enable communication. Therefore, the ontology is used to specify the player roles in the game and formalized their interaction. It also describes the different product classes and their properties as well as concepts such as storage and transport capacities, etc. Because trading goods is a key action in the game, we closely defined contracts used for buying and transporting those goods. The ontology structure provides enough information for the players to be able to understand the relationship between

<sup>12</sup>Ideally, this figure should represent the *has a* relationships as well, not just those of *is a*. Unfortunately, the software system used to formalize the ontology does not have that graphing capability.

<sup>13</sup>see <http://gux.tudelft.nl/svn/ChocolateGameModel/tags/PhdThesisVersion/src/chocolateGameOntology/>

the different goods used in the game. For example, player contract forms <sup>14</sup> are constructed only using the words from the ontology. Any concepts that are found to be missing are added to the ontology.

### 6.4.6 Serious Game

**Gameplay** Before the start of the game, each player received a manual with the specific game rules for his or her role. After a short introduction and the initial loading (chocolate, raisin and peanut distribution), the game was played for 90 minutes with a group of about 15 players. During the game, one scenario was played: after a certain amount of time the prices on the world market of the Raisin Dream bar doubled because of the increase in demand after a new marketing campaign. After the hype was over, the prices dropped again. The goal was to see the change in behavior of the players in this scenario and to make the players realize that they are dependent on each other. Figure 6.14 gives an impression of negotiation about trading goods between actors in the chocolate production during the game.



Figure 6.14: Negotiations during the game

**Game results** After an initial round in which the players were getting used to the concepts and game rules, all players started working on their own strategy for the future, engaging in longer-term contracts. When the world market announced that it would from now on pay a

<sup>14</sup>see <http://wiki.tudelft.nl/Project/GameDesignTradeContract>

much higher price for chocolate bars with raisins, many players immediately tried to adjust their plans to this new situation. The result was that the price of processed raisins on the market between the players also rose, as suddenly all end producers wanted to make Raisin Dream. The producers of half products placed orders on the world market to cope with the increase in demand. As described in the scenario (which was unknown to the players), the world market announced the next turn that the price had dropped back to the old level. The players again tried to adjust their strategies to the new situation, but since they were bound by longer-term engagements they were not able to do this right away. In the debriefing we discussed the game mechanisms with the players who were enthusiastic about the concept and the realization of the game. As predicted, the network of chocolate producers was not capable of reacting in a flexible way to the changes of the world market. Because of this, the players became more aware of the interdependencies between the different actors in the domain, including the link between production and transportation. The players acknowledged this.

#### 6.4.6.1 Agent Based Model

**Model Results** Based on the ontology and the game played, an ABM was developed. The full source code is available online.<sup>15</sup> Figure 6.15 presents the general trading network structure of the model. The ovals represent the agents, and the arrows represent a signed contract. On top is the World Market(s), on the left three Half Product Producers and on the right three End Producers.

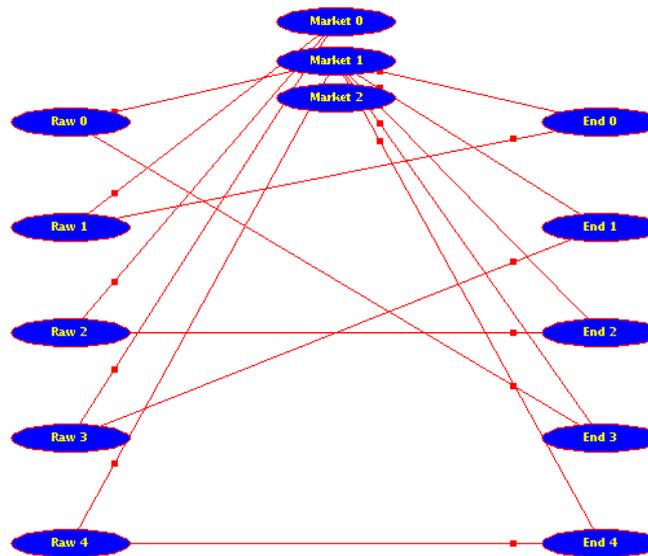


Figure 6.15: Trading structure of the agents

The network structure is emergent and fully determined by agent’s roles. A half product processor must buy raw products that are only sold by the world market agent, and the end producers can only purchase half products produced by the half product processors.

**Price level experiment** It was observed that average price levels varied with the number of agents in each role. In order to examine this effect, the number of agents in each role was

<sup>15</sup><http://gux.tudelft.nl/svn/ChocolateGameModel/trunk/>

varied and the result on the prices observed. The following experiments were performed:

Table 6.3: Overview of the experiment parameters

Experiment	# End producers	# Raw Producers	# World Market
1	30	30	30
2	3	30	30
3	30	3	30
4	30	30	3

The results are presented in Figure 6.16. Large-scale images are available online.<sup>16</sup>

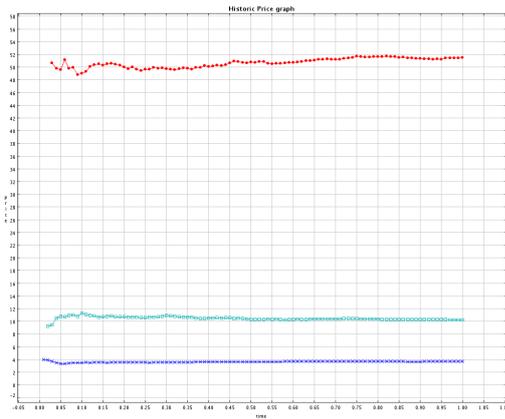
In the first round, the agents that needed the half products asked all other agents if they could buy it from them, but no other agent had any half products in stock yet. However, the producers of half products were able to find somebody (the world market) who sold the raw product needed. They bought a raw product and processed this into a half product. Afterwards, also half products were available for trading. This trading continued until the world market ran out of raw products and the last end products were sold by the end producers. Figure 6.16 plots the results of a run after 100 trading rounds with a certain configuration. Here we can see that different average prices were paid for all three product classes. After a few rounds the averages became more stable. With other configurations we see different values for the prices, but the overall trend of stable price differentiation remains the same.

**Supply and Demand** The stable price levels can be explained by the most basic economics principle: supply and demand. Even though we did not add any market rules directly, a market situation still emerged from the very basic buying strategies we implemented for the agents in this system. Because all agents tried to buy from the cheapest supplier, those who had more choice had a higher chance of finding a supplier with a lower price. Indeed, when we changed the number of agents producing a certain good (thereby indirectly changing supply and demand in the system) we could see that different average prices were paid for each type of good. The position in the network (which is again a result of the negotiation process) determined how rich or poor a certain agent became over time. The results show that commonly observed system level behavior, such as markets prices, can emerge in a model that only includes rules for local decision-making.

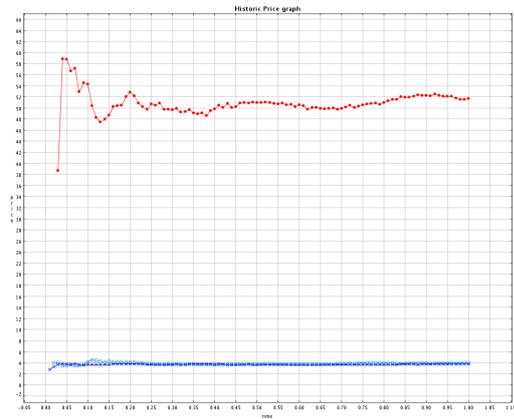
**Proof of concept** This model was designed as a proof of concept that can be easily extended. The modular domain, decision-making and technological components make it possible to extend the model in various directions, making the system more realistic and more complex. It is important to point out that the ABM is based on an analogy of the process industry, which makes this model applicable to this domain without any changes in the behavior of the agents. Using a simple ontology translation, a description of another domain (such as the petrochemical process industry) will allow the agents in this model to reason about this network using the same decision-making objects defined for the chocolate model. This is possible because of the

---

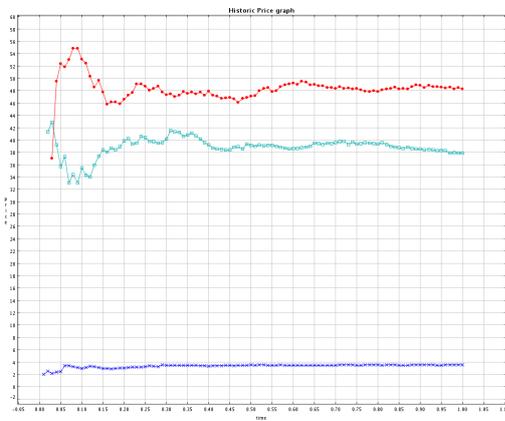
<sup>16</sup><http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ChocolateGameModel/ChocolateGameResults.html>



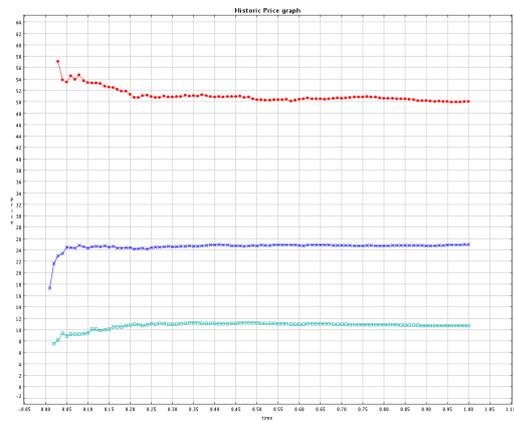
(a) End 30, Raw 30, World 30 (Y: 0 ... 52)



(b) End 3, Raw 30, World 30 (Y: 0 ... 66)



(c) End 30, Raw 3, World 30 (Y: 0 ... 80)



(d) End 30, Raw 30, World 3 (Y: 0 ... 54)

Figure 6.16: Historic chocolate price as a function of trading structure. X axis is time, 0 ... 110 steps. The red line is the price of the end product. The dark blue line is the half product price, and the light blue line is the raw material price.

strong analogy used, which formalizes the knowledge structure and not facts. It is (at this stage) more important to encode the knowledge about flow transformation processes than the identity of the actual flow that is being processed.

### 6.4.7 Domain-specific Insights

**Discrete flow model** The use of the chocolate analogy created one important misinterpretation in the system description. It obscured the fact that most large-scale petrochemical processing networks deal with continuous streams. The analogy caused the model to be built using discrete component streams. While the discrete representation is useful in itself, a continuous representation of streams is also necessary for consistent application of the ontology and the model in different domains. In a future model, this will need to be rectified.

**Emergent prices** Even though all agents in the system are exactly the same except from their technology, the model demonstrated emergent prices for processed and raw materials. The

agents all use the same decision-making component for market research (i.e., asking a random price every time somebody wants to buy one of your goods and when buying yourself always sign a contract with cheapest supplier). Yet from Figure 6.16 we can conclude that there is a difference in *average* price agents pay for their products. The results of the experiment are presented in Table 6.4:

Table 6.4: Overview of the experimental results. (Agents are E: End product producer, R: Raw material producer, W: World market)

Experiment	Agents	Price raw	Price processed raw	Price final product
1	E 30, R 30, W 30	3.8	10.5	50.8
2	E 3, R 30, W 30	3.7	3.7	49.4
3	E 30, R 3, W 30	3.9	36.8	51.4
4	E 30, R 30, W 3	10.7	24.9	50.1

In experiment 1, when there is an equal number of each type of agent present, the price of processed goods is higher than the price of the raw material, and the price of the final product is the highest. This can be explained by the fact that the world market has no negotiation power, since it accepts all and any products offered to it, at the asking price. Both a raw material processor and the final producer have the ability to choose the cheapest contract. This ability to choose the cheapest price causes a lower average price received by the world market and raw producers, respectively.

Experiment 2 show the same downward price pressure for the raw and processed goods. The number of final product producers is lower, and the average processed raw chocolate price is lower, since the end producers will pick only the 3 cheapest products on the market, lowering the average price. Experiment 3 demonstrates the same price-lowering effect. The average price of the processed raw product is much higher here, since many end producers want the product, and all prices offered by the raw processors are accepted, increasing the average price. Experiment 4 demonstrates the same behavior, since a limited supply of a raw material increases its price.

**Analytical validation of prices** As discussed above, there are three interesting prices in the model: those of the raw, processed and final products. The price of the final product is the expected value of a normal distribution between 1 and 100, 50.5. This expected value is reached since the end producers set the price by drawing a number from a uniform distribution between 1 and 100, and the world market accepts all prices set by the sellers without any selection. Both raw and processed have the price which corresponds to a expected minimum value of an informal distribution with 30 draws, namely 3.8. This theoretical value is derived by the following calculation: <sup>17</sup>

Let  $X_1, X_2, \dots, X_N$  be independent random variables, all from the same discrete uniform distribution over the set  $\{a, a + 1, \dots, b\}$ . Let  $W = \min \{X_1, X_2, \dots, X_N\}$ . If  $X$  is a random variable with discrete uniform distribution over  $\{a, a + 1, \dots, b + 1\}$ , then

---

<sup>17</sup>The author gratefully acknowledges the help of Dr. P. Heijnen in deriving the theoretical predictions for price levels.

$$f(x) = \begin{cases} \frac{1}{b-a+1} & a \leq x \leq b \\ 0 & \text{elsewhere} \end{cases} \quad \text{and} \quad F(x) = \begin{cases} 0 & x < a \\ \frac{x-a+1}{b-a+1} & a \leq x \leq b \\ 1 & x > b \end{cases}, \quad x \text{ integer.}$$

$$\begin{aligned} P(W \leq x) &= P(\min(X_1, X_2, \dots, X_N) \leq x) \\ &= 1 - P(\min(X_1, X_2, \dots, X_N) > x) \\ &= 1 - P(X_1 > x) P(X_2 > x) \dots P(X_N > x), \text{ for } a \leq x \leq b. \\ &= 1 - \left(1 - \frac{x-a+1}{b-a+1}\right)^N \end{aligned}$$

$$\begin{aligned} f_W(x) &= F_W(x) - F_W(x-1) \\ &= \left(\frac{b-x+1}{b-a+1}\right)^N - \left(\frac{b-x}{b-a+1}\right)^N \end{aligned}$$

The expected value  $E(W)$ :

$$E(W) = \sum_{x=a}^b x f_W(x)$$

Substituting the values used in the simulation gives a theoretical value of 3.8, which very closely corresponds with the first 3 experiments.

The third interesting price is the price of the processed raw product. This price is dependent on the ratio between the number of raw materials processors (M) and end product producers (N). It is analytically described by the following equation:

For  $i, i \in \{1, 2, \dots, \min(M, N)\}$ , let  $X_{i,1}, X_{i,2}, \dots, X_{i,N-i+1}$  be independent random variables, all from the same discrete uniform distribution over the set  $\{a, a+1, \dots, b\}$ .

Let  $W_i = \min\{X_{i,1}, X_{i,2}, \dots, X_{i,N-i+1}\}$ . The expected value of W is given by

$$E(W_i) = \frac{1}{(b-a+1)^{N-i+1}} \sum_{x=a}^b x \left( (b-x+1)^{N-i+1} - (b-x)^{N-i+1} \right)$$

The expected average value of  $W_i$  equals

$$E\left(\frac{1}{\min(N,M)} \sum_{i=1}^{\min(N,M)} W_i\right) = \frac{1}{\min(N,M)} \sum_{i=1}^{\min(N,M)} E(W_i)$$

If  $M > N$  then the answer does not depend on M. The prediction of experiment 3 is 36.3, which is exactly what the experiment yields. It is interesting to note that with this very simple ABM it is possible to validate the outcomes analytically. As a matter of fact, this validation actually identified an initial logical error in the simulation code that was affecting the price levels. Trying to understand the erroneous outcomes led to development of the analytical solution that pointed to the code error. While this approach is elegant and useful with a simple model, it quickly becomes impossible as the complexity of the agents and their interaction increases.

## 6.4.8 Method Development Conclusions

By using the systems decomposition method and formalizing the domain description in an ontology, a foundation for building a modular ABM is created. The cornerstones of this approach are:

1. System decomposition that allows evolutionary analysis;

2. Formalization of the domain in ontologies;
3. Generalizing the domain descriptions, for example using an analogy; and
4. Implementing the model as an ABM with pluggable domains as well as pluggable decision-making and technological components.

**SDM, Knowledge and fact acquisition** The System Decomposition Method is a knowledge acquisition and social collaboration script. By executing it with a group of experts, relevant knowledge is identified and encoded. It aids groups by preformatting the knowledge gathered, necessary for formalization as an Agent Based Model.

**Ontology use** Ontologies are clearly the way to collect, store and share the collected knowledge. However, the current implementation as a Java class tree built through Eclipse IDE is not easily accessible to non-programmers. Modifications are also relatively time consuming and impractical.

**Game creation** There are two main insights achieved through the use of games. First, a game is relatively quick way to 'model' system behavior, especially when the required agent behavior is very complex. Unfortunately, it is also a very unreliable model, since it is not fully deterministic and repeatable. The second insight is that games are mainly useful as an education tool. Ideally, a complex system would be modelled as an ABM, and the stakeholders should be asked to play a game based on the system to fully understand the interactions involved. The model results can then be communicated to the stakeholder/s with much greater effect.

**Modular model setup** The modular setup of the model promises to be very useful. It requires more initial design and effort to set the model code structure. The approach also requires a more indirect style of programming that makes the code less succinct. On the upside, the payoffs of this approach are expected to be in knowledge and fact reuse in subsequent models, and especially in increased speed of model development in the future.

**Technology and strategy** The initial ontology defined the concept of 'strategy' as something prescribing what the agent will produce. It became clear during the modelling process that this is a misnomer, since what determines the production is the technology, not the strategy. The concept of strategy, however, is a useful concept and will be kept for the future and used as something that modifies the way technology is used.

**Discrete flows** Discrete flow-based simulation is viable. However, using the analogy of chocolates an important misconception crept into the model. All bulk petrochemicals are produced in large quantities, in continuous processes. The concept of a discrete piece is very unwieldy when dealing with streams. In the following cases the description of technology will have to be re-conceptualized for continuous streams. Discrete flows, however, are very relevant for other domains, such as transport.

**World market** The model assumed that the world market consists of many individuals, and was modelled as such. This is incorrect, as the world market is assumed to be an aggregate of many unknown entities. It should be modelled as such. This furthermore greatly simplifies the implementation of the model and its performance, since it dramatically reduces the number of agents.

**Requirements checked** Table 6.5 presents an overview of the modelling requirements and the performance of the model.

Table 6.5: Performance of the case study

Requirement	Score	Explanation
Method		
Open Source	Yes	All computercode, game design deliberations and materials and the ontology are available to the involved social network
Sufficient community diversity	No	Only students and staff from TU Delft involved in ontology, game and model creation.
Organically growing	Yes	Ontology, game and the model are direct conceptual descendents of the first case study, with generic modelling insights from the second case applied.
Recorded history	Yes	The entire modelling process, both formal interaction via code and ontology and informal interaction via wiki are versioned.
Enforceable authorship	Yes	Via authentication in subversion and wiki.
Modular	Yes	Ontology, agents and decision algorithms are designed with modularity and reuse in mind.
Outcome		
Useful	Yes	Provided plenty insights into ontology development and modelling process. Developed a usable system decomposition method.
Testable	Yes	The SDM can be repeated and results compared. Model is repeatable due to versioning and random seed record.

## 6.5 System Decomposition Method

This section presents the theoretical background of the System Decomposition Method (SDM), a social process developed to encode multiple formalisms/domains into the Agent Based Model used to model the evolution of  $\lambda$ -systems. First, the concepts of formalisms, knowledge interfaces and knowledge states are introduced. The System Decomposition Method is presented, consisting of a group modelling exercise where different knowledge domains and formalisms interact and transform knowledge through

interfaces into different states, finally allowing multiple formalisms to be encoded into the states and rules of an Agent Based Model.

### 6.5.1 Background

This section will present the theoretical background of the formalized knowledge collection process that enables the creation of multi-formal and multidisciplinary models and their integration with ABM. Large parts of this section are essentially part of the results of the thesis. The System Decomposition Method (SDM) co-evolved with the technical aspects of the model creation, the knowledge encoded and the facts collected.<sup>18</sup> Even though it is a result of the evolutionary method, it is necessary for the backgrounds to be introduced here, so that the reader can understand the development path of the work with the advantage of hindsight and can follow the vocabulary used.

The starting point for discussing the theoretical backgrounds of the SDM is Mikulecky's definition of complexity, and the insight that one can not study complexity alone but must do it in groups.<sup>19</sup>

**Definition** Knowledge engineering is defined as (Feigenbaum and McCorduck, 1983):

an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise.

Knowledge engineering is a large field, the full review of which would be outside the scope of this thesis. This section presents our (mainstream) perspective on the field.

**Formalisms** As discussed in Chapters 1 and 3, understanding complex systems is not only hindered by systems' intractable evolution, but by the requirement that multiple distinct formalisms be used in describing them. This means that the knowledge required for understanding and modelling  $\lambda$ -systems is based on knowledge and facts from many different knowledge domains and many different people, each with their own formalism. In the section 3.2, formalisms will be discussed and their use explained.

**Knowledge transitions and states in modelling** In Section 6.5.3 the knowledge states are presented. These states are encountered when the modelling method moves from unstructured, unshared and often unexpected knowledge of stakeholders to a formalized, shared and machine-encoded knowledge that can be implemented in an ABM.

**Collaborative research** As already discussed, we are interested not only in understanding, but also shaping, the evolution of  $\lambda$ -systems. In order to do so, one must engage the social networks that shape it and provide them with new insights. This means that the knowledge collected is shared and accepted by all involved stakeholders. This in turn then means that the work must be collaborative. Section 5.2.1 will discuss the main practices, problems and issues of collaborative research.

---

<sup>18</sup>Please refer to Chapter 5 for a description of the modelling method.

<sup>19</sup>This section is developed jointly with dr. P.J. Beers, and resulted in publication (Nikolic et al., 2007).

## 6.5.2 Modelling in Groups

Why model in groups? There are two main reasons. First, combining knowledge of multiple people satisfies the multi-domain and multi-disciplinary knowledge requirement of CAS modelling. Second, if correctly *black-boxed*, group modelling lowers the transaction cost on knowledge exchange (Beers and Bots, 2007). The concept of black-boxing will be discussed below.

**Black-boxing** The modelling group needs to be able to effectively share knowledge with each other and understand each other. They need a shared cognitive frame of reference (Bromme, 2000), whereas they generally have little knowledge in common to begin with (Alpay et al., 1998). We aim for exactly the amount of shared knowledge needed to create interfaces between disciplines and domains, and no more. We call this process *black-boxing each other*. We aim for the smallest sufficient level of shared understanding, that is, enough to edge disciplines and sub-models, but not so much that we would become experts in our mutual disciplinary fields. To give an example from real life, an engineer does not need to know how a micro-economist models an individual's decision process about buying a car. For her it is enough to know that she needs to provide the economist with the parameter values associated with the car, e.g. its maintenance cost and its mileage. Likewise, the economist does not need to know how exactly the design of the car determines maintenance cost and mileage. The only thing the economist and the engineer need is a shared interface between their individual knowledge domains. This means that people cannot know everything, but they can know enough to make collaboration possible. By sharing a common interface, each modeller can stay within the relative comfort of her field, and contribute domain knowledge to the agent-based model without needing to know the details of other fields. The interaction across field thus becomes 'cheaper'. The knowledge transfer transaction costs go down.

**Knowledge transformations** Having explained the importance of modelling in groups, we will proceed by exploring the knowledge processes involved. Let us imagine the process that takes this disciplinary knowledge from a group of individuals and transforms it into an agent-based model.

## 6.5.3 Knowledge Transitions

**Teamwork** This method attempts to bring Complex Adaptive Systems modelling and multi-disciplinary teamwork together. It does so by focusing on the various stages that take knowledge from being unshared and unformalized to becoming formalized in the form of computer models. The route from unshared and unformalized knowledge in persons' heads to formalized and shared knowledge in a model (see Figure 6.17) goes through two intermediate knowledge states (shared unformalized knowledge and formalized shared knowledge). We distinguish three interfaces between these states: soft-to-soft, soft-to-hard and hard-to-hard.

**Interface** An excellent definition of interface is provided by Wikipedia<sup>20</sup>:

---

<sup>20</sup>[http://en.wikipedia.org/wiki/Interface\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Interface_(computer_science)), Accessed 22 February 2007

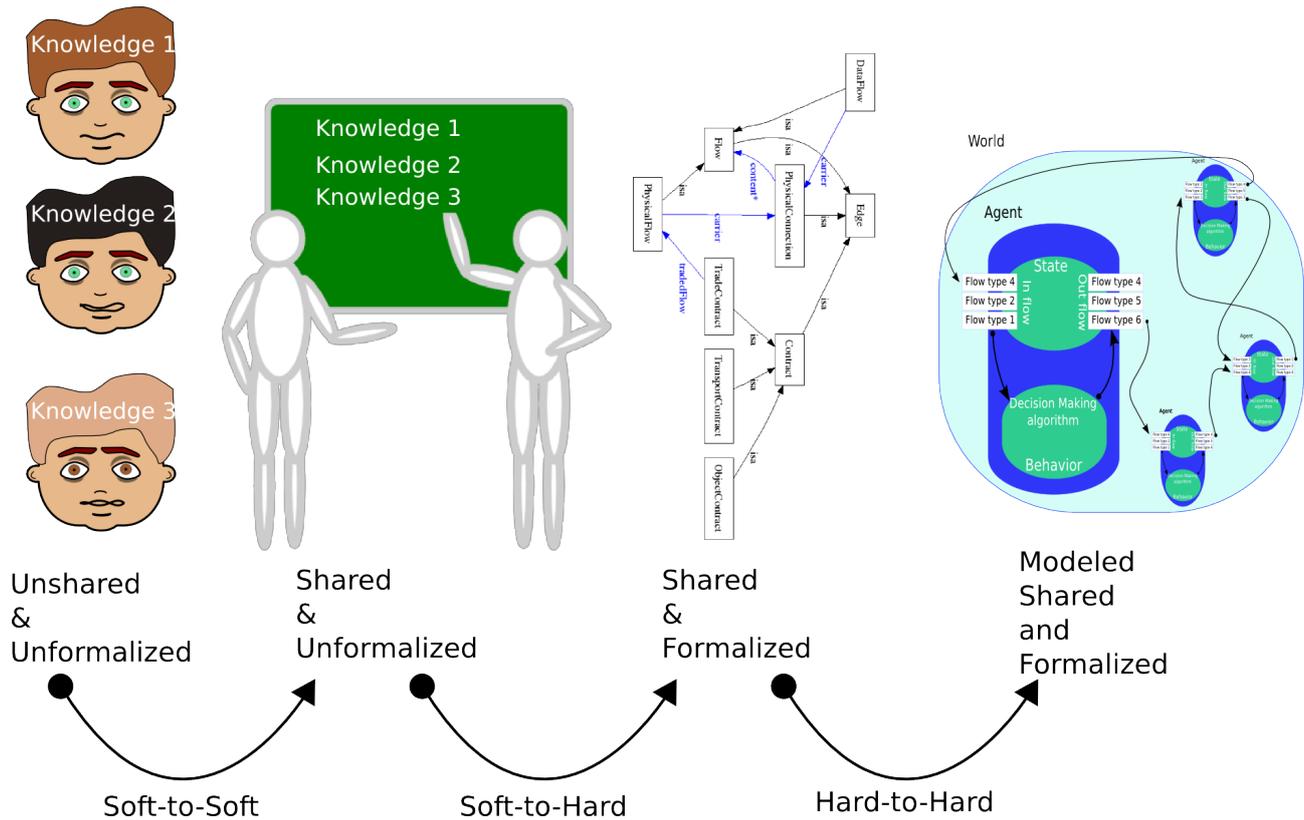


Figure 6.17: Knowledge transitions from unshared & unformalized to shared & formalized

An interface defines the communication boundary between two entities, such as a piece of software, a hardware device or a user. It generally refers to an abstraction that an entity provides of itself to the outside. This separates the methods of external communication from internal operation and allows it to be internally modified without affecting the way outside entities interact with it, as well as provide multiple abstractions of itself. It may also provide a means of translation between entities which do not speak the same language, such as between a human and a computer.

Although this definition is primarily aimed at computer science, it can be easily adapted to our purposes. For instance, the soft-to-soft interface is one between (the minds of) different 'users' or people, the interface itself consisting of represented knowledge. Similarly, the soft-to-hard interface is one between people and ontology, the interface itself consisting of tangible rules for using the ontology itself. The transformation of knowledge from each form to another, passing these interfaces, is accompanied by specific challenges.

**Soft-to-soft** The soft-to-soft interface resides between people, or, more specifically, between the minds of people. It can be seen as a soft communication interface between two or more cognitive networks. Both the way people store their knowledge and the way people communicate influence the nature of this interface. Different people have different distinct ways of storing their knowledge, based on their unique personal (prior) knowledge. Their personal knowledge is cognitively organized in ways specific to their domain and/or disciplinary expertise.

When people collaborate in groups, they try to communicate their knowledge to their collab-

oration partners. A number of conditions needs to be satisfied for their message to successfully reach its intended audience. The audience must have enough prior knowledge to be able to understand the message. However, that is not as easy as it sounds. First, conceptual ambiguities abound when people with different disciplinary backgrounds collaborate (Beers and Bots, 2007). Secondly, experts generally overestimate the availability of their knowledge to the general public and are therefore prone to convey insufficient information for the audience to come to understand their contributions (Bromme et al., 2001). In other words, just telling the other what you know will not be sufficient to achieve a common understanding of each other's knowledge (Bromme, 2000) and (Clark and Schaefer, 1989), because apart from having different knowledge, people also have their own, idiosyncratic ways in which they store, process and communicate their knowledge (Boshuizen and Tabachneck-Schijf, 1998). The associated challenge is how to overcome these representational differences. Effective communication then demands that discussion partners negotiate ways to deal with these representational differences. One of those ways is to use pre-defined representational formats (formalisms) to guide knowledge exchange (Van Bruggen et al., 2002).

**Soft-to-hard** The soft-to-hard interface lies between unformalized and formalized knowledge. It is an interface between people and models. This interface is characterized by conceptualization and formalization. It means that the shared body of knowledge of a group of people is analyzed and defined in terms of the individual concepts that constitute it. On the one hand, this produces a very explicit account of the shared body of knowledge, and on the other hand the yields can be seen as fully formalized, in the sense of being an ontology of that body of knowledge (Gruber, 1993). It should be noted that the word 'ontology' is meant here in the software engineering sense, not in the philosophical sense.

**Challenge** The challenge associated with this interface resides in the difference between cognitive and formal knowledge representations. The human cognitive architecture uses concept networks that operate with stereotypes. This representation is reminiscent of fuzzy logic, where things can to some extent belong to multiple categories. People, especially the non-modelling community (a non-negligible part of our current-day society), are apt to formalize their knowledge up to a level below what is required by computers. Traditionally, modellers step in to fill this gap, but in the case of complex adaptive systems they often lack the necessary domain knowledge. To enhance the interface between people and models, one may use representational formalisms that require a large extent of formalization of content.

The System Decomposition Method facilitates crossing the soft-hard interface by starting with a very small conceptual model that is in line with the modelling technique to be used later on. In our case, we chose to use agent-based modelling and thus used an ontology of an agent as a basis for further conceptualization and formalization. Starting from the agent, the modellers can define instances of agents and build new agent characteristics and behaviors, but all still in connection to the early agent ontology. In that sense, the soft-to-hard interface becomes characterized by a dynamic process of adding, defining, and adding more, without an explicit stopping rule.

**Hard-to-hard** The hard-to-hard interface exists between coherent parts of formalized knowledge, which in this case are agents. This knowledge, by definition, is fully formalized and

computer-readable and -processable. In that sense, the following holds true irrespective of whether it refers to a pen-and-paper ontology or pieces of computer code. Thus, insights from computer science about interfaces between computer programs are fully applicable to the case at hand. In the case where agents constitute the coherent parts of formalized knowledge, computer science teaches us that such agents each have an Application Programming Interface (API). When the interface is specified, communication between components becomes fully explicit and formal. In practice this means that when an agent implements an interface (API), one knows exactly how to communicate with that agent. So the communication between agents becomes fully specified and unambiguous.

**Hard-to-soft** The final interface in the method, which is not presented in Figure 6.17, is the hard-to-soft interface. This interface enables the use of the model and transforms the model outcomes and data to useful, applied knowledge. It is a very complex matter, involving fields like psychology, perception, social science (reasons for why people believe one source and not another), etc., and its exploration is considered to be outside the scope of this thesis.

**Challenge** The main challenge with this interface is bound to its strength. This interface, in order to be functional, needs to be very strict and formal. This severely limits the usability of the interface by modellers, since they need to be able to 'speak' such a formal interface. Computer programming is an abstract skill relatively difficult to master.

**Pitfalls** The interface between agents actually is one between parts of a model, or between sub-models. Our position is that a set of formal models that can communicate and are used in interaction can be regarded as a single model. However, two types of pitfalls can undermine the interoperability of model parts or, in other words, confound the hard-to-hard interface.

The first, less severe type, is the *implementation pitfall*, in which case conceptualizations are compatible but implemented differently. For example, the concept of temperature can be expressed both in degrees Centigrade and degrees Fahrenheit. They mean the same thing, but their different expressions make them non-interpretable. These errors, although practically difficult to solve, are conceptually trivial; solving them is a matter of translation.

The second, more severe type arises from *conceptual mismatch*. This refers to cases where, even with correct individual model parts, modelers are of the impression that they use the same terms for the same concepts in the same way, whereas in practice the formalization is not. It amounts to reading a number to assess temperature, but using a hygrometer to do so. To enhance the interface between sub-models and model parts, one can use ontologies.

## 6.5.4 Knowledge States

The system decomposition method (SDM) is effectively a collaboration script for agent-based model building using a group of experts/stakeholders. While executing the script, each of the interface problems identified earlier are addressed. Earlier versions of the SDM have previously been reported in (Nikolic et al., 2006, 2007, 2009). A schematic representation of the method is presented in Figure 6.18. A full-scale version is available online <sup>21</sup>. The method consists of two parallel paths, the Facilitators' and Modellers' Path, and the Stakeholders' and Domain

---

<sup>21</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/Process\\_Scheme\\_SDM.png](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/Process_Scheme_SDM.png)

Experts' Path. These will be presented below. It is important to notice that in this case study, the Modellers and Facilitators are the author and researchers from the TU Delft, the Groningen Seaports are considered to be the stakeholders, and that diverse domain experts are involved when necessary.

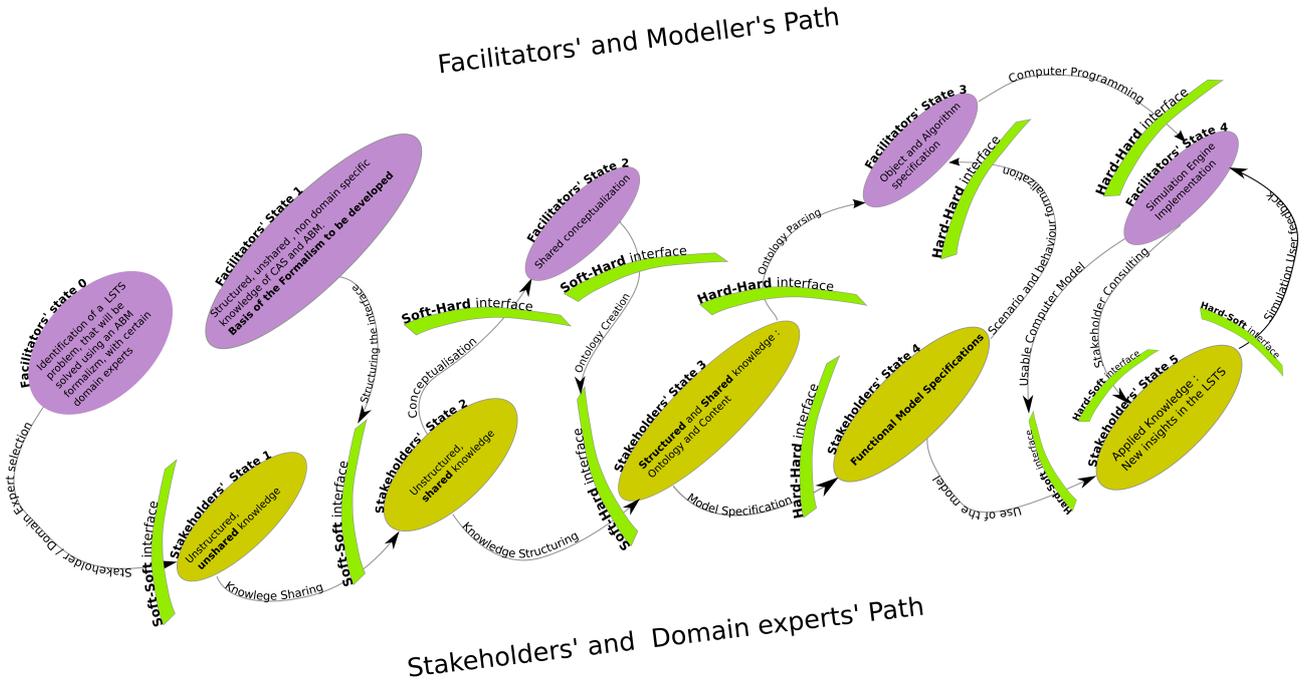


Figure 6.18: System Decomposition Method

In this section, I will present each knowledge state and the activities belonging to both the modeller's and the stakeholder's path. In the Results section the actual results of the SDM application will be presented.

#### 6.5.4.1 Facilitators' and Modellers' Path

The Facilitators' and Modellers' path, from now on referred to as the Facilitators' path, is the 'backstage' process of the SDM. It is the method that allows the stakeholders and domain experts to efficiently formalize and share their knowledge and to eventually increase their understanding of  $\lambda$ -systems.

#### 6.5.4.2 FS 0: Objective and Setting

As can be seen from Figure 6.18, the SDM method starts in the Facilitators' Path, at the Facilitators' State(FS) 0. In order to be able to place the whole SDM in a context, the setting and goals need to be made clear to both the facilitators and stakeholders.

**Objective** As discussed previously, we are about to begin a process of increasing our knowledge about the state and behavior of a  $\lambda$ -system. In order to do so, we are creating a complex Agent-based Model. We have postulated that we must do this by gathering data from stakeholders and domain experts in a structured and systematic manner. We must share and formalize

the knowledge present in many different heads and create a socially accepted, formal computer model. While going through this method, the stakeholders and domain experts will go from Stakeholder Knowledge State 1 to 5, see Figure 6.18, creating an increasingly formal knowledge base. At the same time, the facilitators and modellers will follow a parallel path, supporting the process and creating the computer simulation based on the formalized knowledge.

**Setting** The process setting is as follows. There is a small group of facilitators and modellers (Facilitators), and a group consisting of stakeholders and domain experts (Stakeholders). The facilitators, together with the stakeholder/client involved in a project, determine which specific domain experts need be involved. The such formed Stakeholders group will enter the Stakeholders' / Domain Experts' path. The facilitators create the conditions and prerequisites that the stakeholder group needs in order to move from state to state. It is important to realize that while the stakeholder participants remain the same throughout the process, different domain experts can be involved at different times. On the facilitation side, it is advantageous if the facilitators and the modellers are the same group of people, but it is not strictly necessary. Once enough formalized knowledge has been gathered from the stakeholders group, after Facilitators step 2, modellers can take over the process.

#### 6.5.4.3 FS 1: Structuring the Interface

At the Facilitators State 1, the group of facilitators explicitly defines their knowledge of Agent-based Modelling. This state makes sure that all the facilitators and modellers are 'on the same page' concerning the very basic concepts that are needed in order to structure the soft-soft interface. It is the facilitators' role to introduce the stakeholders group to ABM concepts without explicitly mentioning them. The method involves creating a structured, unshared formalism based on the Agent-based Modelling paradigm and using it to structure the soft-soft interface between Stakeholders Knowledge States 1 and 2.

A schema of the basic agent formalism used in the SDM is presented in Figure 6.19. The agent formalism determines the transition between the Knowledge State 1 to 2, across the soft-soft interface.

**Interacting nodes** Given the basic ABM formalism, agents are considered to be nodes in an interaction network. Nodes are connected by edges, which are abstracted interactions. An agent consists of a number of incoming interactions (In Edges) and a number of outgoing interactions (Out Edges) (Jennings, 2000a; Newman, 2003). Furthermore, an agent consists of a State and Decision-making. The interactions between the State and Decision-Making result in the agent's behavior, manifested as outgoing interactions. Both State and Decision-making consists of Data that are considered to be objective facts in a sense that they are not observer-dependent. For example, a weight or length of something is an objective form of knowledge. Knowledge is defined as data or algorithms that are observer-dependent. It is the codified experience of agents (Stefik, 1995a).

Four concepts form the basic formal ontology:

- Node
- Edge

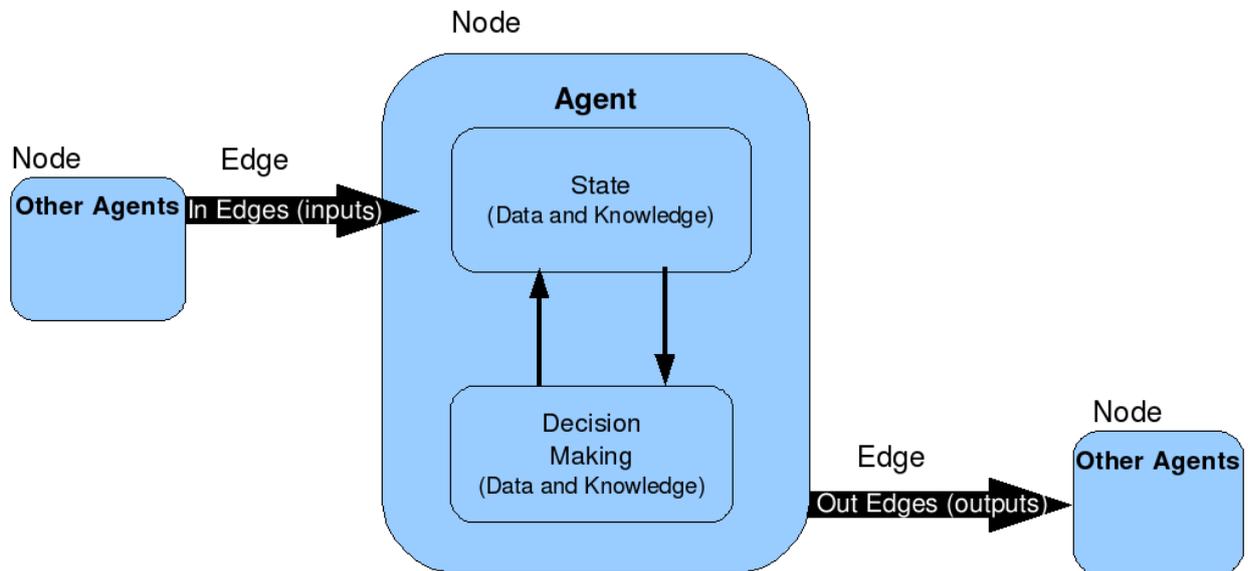


Figure 6.19: Basic agent formalism

- Property (Objective Knowledge)
- Knowledge (Subjective knowledge)

**Node and edge** Their relationships are presented in Figure 6.20. We can see that the Node and the Edge are interdependent. These two form the basic concepts of graph theory (Gross and Yellen, 2004; Newman, 2003). Everything is abstracted to be either a node or an edge. The choice of which an object is abstracted and how is purely a matter of convenience. Data and Knowledge are considered to be two distinct, independent concepts.

**Prestructuring concepts** These concepts are used to pre-structure the Knowledge Sharing process, by setting the implicit vocabulary for the social process. Everything is namely to be expressed in terms of objects or entities (Agents), their interactions (Edges), properties of these two (Data), and their mechanisms/behaviors (Knowledge). To show how the basic agent formalism and the corresponding basic ontology are used, we will present an example agent, a baker.

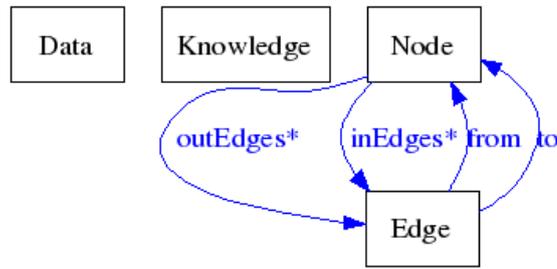


Figure 6.20: The four basic concepts of the formalism. Arrows denote a *has a relationship* with an instance of a class. A star denotes a possibility for multiple instances. So a Node has out Edges that are multiple instances of an Edge.

**Baker as an example** The bakery and its owner are represented as the Agent (Node). The baker has flour and yeast suppliers (In Edges), and has customers (Out Edges). These are the Other Agents (Nodes). The supplying and buying are interactions between agents that are abstracted as Edges. The bakery (Agent) has a State that consists of the amount of flour and yeast in stock (Data) and a number of bread recipes (Knowledge). The Decision-making consists of the amount of bread sold (Data) and the rules the baker follows when setting the price of bread (Knowledge).

**Conceptual bridge** The agent formalism serves as an unambiguous conceptual bridge that structures communication across the soft-soft interface. Its explicitness alleviates problems with conceptual ambiguities in individual knowledge by acting as a shared vocabulary focus point. Given this vocabulary, we can start the SDM method at the first knowledge state SDM1. The facilitators invite the stakeholders to a guided brainstorming session.

#### 6.5.4.4 FS 2: Conceptualization

**Structure and abstract** The next step in the Facilitators' path is to structure and abstract the knowledge, forming an ontology. Initially informal, this knowledge is abstracted and converted into a computer-readable and -processable formal ontology. This ontology contains explicit formal specifications of the terms in the domain and the relations among them (Gruber, 1993). It is effectively a vocabulary of words used *and* the relationships between those words. The problem with this step is that generally, the domain experts do not understand the requirements for creating computer-readable and -processable knowledge, while the computer modellers have no understanding of the domain knowledge involved.

**Frame-based ontology** In practice this means a group effort to create a formal ontology, using ontology editing software, such as Protegé (Gennari et al., 2003). Ontology creation is not an exact science but more of an art. (Noy and McGuinness, 2001) It is a social, iterative method in which the concepts identified in Stakeholders' State 2 are formally described, assigned a hierarchical structure and properties. A Frame-based ontology (Chaudhri et al., 1998), as used by Protegé, is a knowledge representation model that recognizes two types of relationships: *Is a* and *Has a*. It is therefore ideally suited for representing concepts that are object-oriented in nature, such as Agent-based Models.

**Soft-hard interface problem** This step addresses the soft-hard interface problem identified in Section 6.5, by using as a starting point the combination of frame-based and agent-based ontologies that are already computer processable. If pre-structured this way, the soft-hard interface guides the knowledge structuring process in a way that is computer compatible from the start. It is both sufficiently simple and complete for the domain experts to understand and programmable enough for the modellers to use. This is another example of *black-boxing* each other.

**Edge example** In order to illustrate the resulting ontology, we will present the subclasses (the *is a* descendants) of the Edge concept in Figure 6.21. The entire ontology (more than 170 classes and 100 slots) is available online.<sup>22</sup>

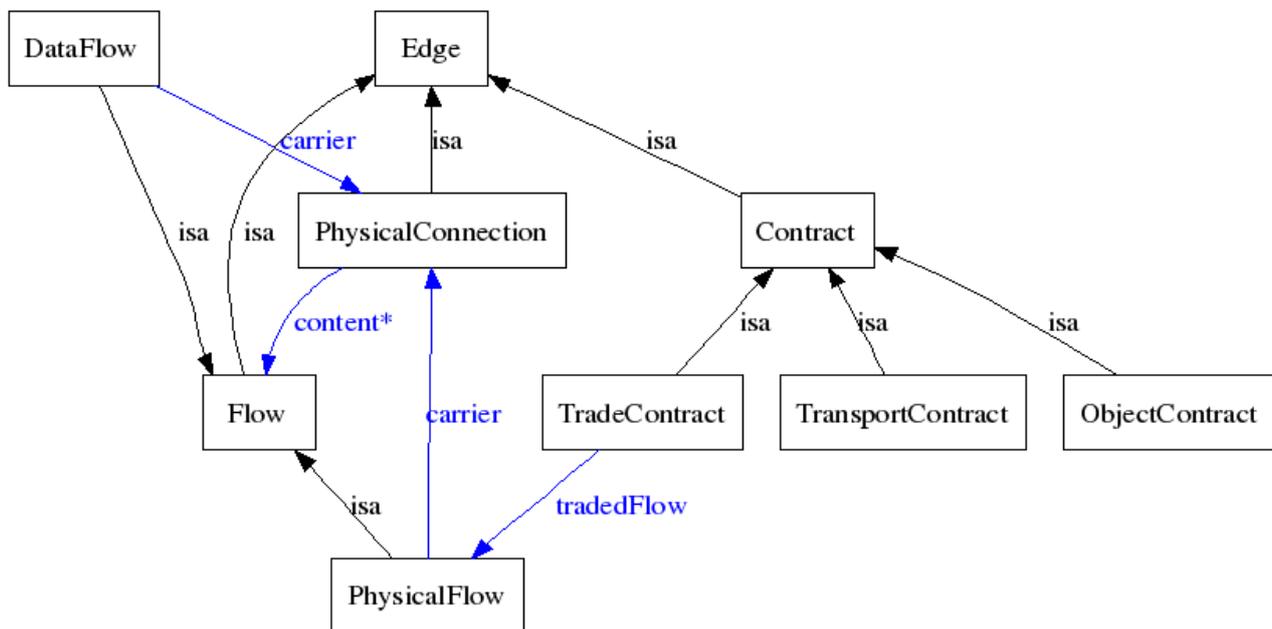


Figure 6.21: Example ontology structure for the concept of Edge

*Is a* relationships denote subclass links. That is, a Physical Flow *is a* Flow, and Flow *is a* Edge. The *has a* relationships describes properties of objects. In Figure 6.21 the *has a* relationships are represented by the name of the property. So a Physical Flow has a *carrier* that is of type Physical Connection, and a Physical Connection has a *content*, which is of type Flow.

**Only what, not how** It is important to note that a frame-based ontology is not able to store behavior (algorithms). It only stores facts. It, as it were, answers the 'what is' question, and not the 'how to' question. Behavior of components is thus not stored in the ontology but in the actual model code, in Facilitators' State 4. Once the ontology is created, it is used as the soft-hard interface that collects and formalizes the stakeholders' knowledge transition from Stakeholders' State 2 to 3.

<sup>22</sup><http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/AgentOntologyExpanded.gif>

#### 6.5.4.5 FS 3: Object and Algorithm Specification

**Java objects** This state further formalizes the knowledge from Stakeholders' States 3 and 4. During this state the modellers in the facilitator group become more active. In this step the ontology created in state 3 is analyzed, and corresponding Java object classes are created. This is necessary because the facts are encoded in Protegés internal data structure, frames and slots, which is not practical for use in the ABM. The structure of the factual knowledge in the ontology is mirrored in the structure of POJOS (Plain Old Java Objects) <sup>23</sup>. The POJOS conform to Java object interaction and communication standards, and are thus usable in any Java program. Using custom software, the GenericKnowledgeBaseReader <sup>24</sup>, instances of these POJOS are created and made available for processing by the decision-making algorithms.

**Behavior extraction** From Stakeholders' State 4, we extract the specification of the Agents' and the environment's behavior. The state 4 tells us exactly which decision-making processes need to be encoded into the agents to give them their behavior, and which specific dynamics behavior should the agents' environment have. The behavior is very tightly tied to the fact-/object, since the decision-making algorithms reason over the facts, that is, instances of the POJOS described above. The environment's set of dynamics is also tightly tied to the agents' decision-making, since its role is to affect the agents and their decision processes. These algorithms are formalized either as plain Java methods (functions) operating on the POJOS, or as rules defined in the inference engine.

**Formalized and shared** At this point, the modellers have access to formalized and shared knowledge, stored in a digitally readable and inferable format. By inferable we mean that a computer, given a suitable inference engine, can generate conclusions (inferences) about facts given, their structure and relationships. As mentioned in Section 6.5.3, knowledge structured this way forms an API. As mentioned above, the knowledge is now contained in a Protegé database, which can automatically be translated into Java (Corporation, 2000) code and made available to the programmer.

**Model description** With these classes and instances, the programmer can start describing the system in terms of an agent-based model that automatically conforms to the created ontology. For example, if agent representing a certain form needs to be described, the programmer will retrieve that agent from the knowledge base, add some explicit behavioral rules, and let the agent interact with others. Of course this implies the presence of suitable ontology database and translation/instantiation software, e.g. the GenericKnowledgeBaseReader class.

**Behavior storage** As mentioned in Section 6.5.4.4, the actual mechanisms of behavior can not be stored in the ontology. They are stored as behavioral rules using a declarative language called JBoss Rules (Forgy, 1982; Proctor et al., 2007), which is processed using a forward chaining inference engine. Since the rules are also based on the ontology, we can create a modular and extensible behavior library.

---

<sup>23</sup>[http://en.wikipedia.org/wiki/Plain\\_Old\\_Java\\_Object](http://en.wikipedia.org/wiki/Plain_Old_Java_Object)

<sup>24</sup><http://gux.tudelft.nl/svn/SimulationGenerics/trunk/src/simulationGenerics/GenericKnowledgeBaseReader.Java>

#### 6.5.4.6 FS 4: Simulation Implementation

**Code writing** Once we have the algorithms in place using the previous step, the simulation is created, building the necessary support computer code, like the scheduler, graph plotting, statistics collection, experiment setups, etc. While this is all standard computer programming, and theoretically trivial, it is not easy. A large amount of time is involved here, since this is a very complicated and error-prone problem. Continuous testing (unit testing, debugging procedures) need to be performed in order to be able to create a verifiable model.

**Computer simulation** An output of this state is a usable computer simulation that can be given to the stakeholder group (see Stakeholders' State 5). From this state we also support and consult the stakeholder group by giving advice and support about the use of the model and adding refinements when necessary.

#### 6.5.4.7 Stakeholders' and Domain Experts' Path

When discussing the Facilitators' path, we examined the 'behind the scenes' processes of the SDM. The stakeholders' and domain experts' path is the front stage of the method, as experienced by the participants. By now, we understanding the 'backstage' processes, and many activities done by the participants will become self-obvious. For brevity, we will refer to this path as the Stakeholders' path, and the states in it as Stakeholder States (SS).

#### 6.5.4.8 SS 1: Unstructured and Unshared Knowledge

The first state of the Stakeholders' path is the unstructured and unshared knowledge state. For the participants in the process, this is perceived as the starting point of the SDM. In this state a group of experts and stakeholders is brought together in a working group. They collectively have a lot of knowledge about the system. At this point, neither the facilitators nor the stakeholders are able to integrate this knowledge so as to increase their understanding of their  $\lambda$ -system for their benefit. The goal is to transform this loose working group into a tightly knit learning social system.

#### 6.5.4.9 SS 1→SS 2: Knowledge Sharing

**Knowledge sharing** By using the Facilitators' State 1 (see 6.18) as an background, a social process of sharing the available knowledge is started across the soft-soft interface. Practically, this guided group brainstorming session can for example be implemented as a Group Decision Room and thinkLet-facilitated collaboration process (Kolfshoten et al., 2006) or as a wiki-mediated knowledge collection process. All of these methods have in common that they require extensive preparation by the facilitators. A generally applicable format that has been applied in a variety of situations will be presented below.

**SDM script** Using the concepts and vocabulary from the Facilitators' State 1, the group is guided to present and exchange knowledge. The group is asked to think about the objects and entities that are relevant and part of the system/problem. They are also asked to think about properties of those objects/entities and the interactions that they have with others. Also, the group is asked to name mechanisms/behaviors of entities. Emphasis is on breadth, not depth.

Through social interaction the group encourages each other to identify more and more entities, properties, interactions and mechanisms. The exact script used to facilitate this process is presented in Section 6.4.4.1.

**Making knowledge explicit** The objective of this method is that the participants share and make their domain knowledge explicit. It is important that when this process is completed the facilitators have sufficient output in order to continue to the Facilitators' State 2, and that the participants experience a degree of satisfaction and achievement. During the session wrap-up, the facilitator should reflect on the large amount of knowledge that was shared and offer a glimpse of the next steps.

**Iteration** This method, if necessary, can be reiterated to make the knowledge pool as complete as possible. Reiteration will further reinforce the soft-soft interface by reducing the problems presented in Section 6.5. The conceptual ambiguity is reduced, since the group discussed many shared concepts at length and sufficient information will have been exchanged.

#### 6.5.4.10 SS 2: Structured and Shared Knowledge

In this state the knowledge shared in the previous step will be structured and shared through a formal ontology. The formal ontology is presented in Facilitators' State 2, Section 6.5.4.4.

Once a shared, unstructured pool of knowledge is assembled at Knowledge State 2, it needs to be structured and formalized. From this state, two activities need to be performed. First is the transition to Facilitators' State 2, the creation of the ontology. This is a process performed by the facilitators, described in Section 6.5.4.4. Second, the created ontology is used to create the soft-hard interface enabling the Knowledge Structuring and transition to Knowledge State 3, a structured and shared state.

#### 6.5.4.11 SS 2→SS 3: Knowledge Structuring

**Knowledge storing** In this method the ontology is used as a repository/database to structure and share the available knowledge. This is facilitated by the ontology editor and its back-end database. Users enter their knowledge about the state of the world through the user interface. This knowledge then becomes available for all users. Because the ontology is created in a participative manner, all users should be able to express all their knowledge in it. In case concepts are still missing, or are ill defined, the conceptualization process can be reiterated.

Because the soft-hard interface, the ontology, is used, we avoid the discrepancy between cognitive and formal representation, see Section 6.5. The cognitive knowledge must be expressed by literally filling out a form in the ontology. The form and the concepts in it are a direct result of the social process and are negotiated. The user wishing to represent her knowledge must use the pre-agreed concepts, therefore removing any ambiguity from the representation.

#### 6.5.4.12 SS 3

**Hierarchy of concepts** The knowledge in the ontology is of two types. First, the ontology contains the relationships between concepts, the class hierarchy. As mentioned in Section 6.5.4.4, class in this context should be understood in an object-oriented sense, as a generic data type. A class is a generic description of a concept that follows simple inheritance rules. For

example, all fruit can be represented by a class Fruit. subclass of Fruit is Apple. If Fruit has certain properties such as color and taste, Apple will inherit those, too. An apple can not be a Vegetable, since a class can only have one parent, one super-class.

**Instances** In addition to class description, the knowledge base contains instances of those classes. That is, we cannot only talk about Apple class things, but also about one specific apple, that has a green color, a sour taste and is in my pocket right now. This apple in my pocket is an instance of a generic class Apple. The result of this state is that we have a repository of relationships between concepts and and a database of actual things that modellers of  $\lambda$ -systems should consider.

**Process end** This process ends in the Knowledge State 3, the structured and shared knowledge. We now have a socially constructed and accepted ontology, and a knowledge base of facts about the system/problem being modelled. Given this shared and formalized knowledge, two different processes are started.

#### 6.5.4.13 SS 3→SS 4: Model Specification

**Ontology parsing** The first step is the parsing of the ontology, through a hard-hard interface. This process is described in Section 6.5.4.5. The instance descriptions are now available to the modeller. However, in order to create a model, a description of the agents' decision-making and the dynamics of the environment must be created.

**Behavioral model** The domain experts are asked to create formal descriptions of these decision-making processes. This can be done using state transition diagrams, causal diagrams, flowcharts or if-then reasoning structures. Such formal descriptions are effectively programmable algorithms for operating on the instances defined in the ontology. Algorithms collected from domain experts are then ready to be formally implemented in Facilitators' State 3.

#### 6.5.4.14 SS 3→FS 4: Ontology Parsing

This process takes place over the hard-hard interface. Is is the process of translating the concepts formally defined in the shared ontology into Java objects. This was presented in Section 6.5.4.5.

#### 6.5.4.15 SS 4: Functional Model Specification

**Model description** At the Stakeholders' State 4, the mode is fully specified. All the knowledge is recorded in the formal ontology, and all the behavior is specified in a formal model. The model is fully specified, in term of what is in it and what it does. Scenarios and other dynamic components are also specified here, as well as the exact metrics that will be used to measure the model. At this moment, there is no actual computer model yet. The knowledge collected at this state is used by facilitators to start the computer model implementation.

**Model implementation** At this moment, the Stakeholder and Domain Experts group's activities are halted for the duration of the computer model implementation. This may take several days to a few months, depending on the size and complexity of the model.

#### 6.5.4.16 SS 5: Applied Knowledge

**Model use** In this state the model is used to increase the insights and knowledge about  $\lambda$ -systems. It consists of feedback workshop(s) between the modellers and the stakeholders. These workshops are ideally an iterative process.

**Iterative process** The iterations are performed over the hard-soft interface. Model outcomes are presented, and the stakeholders react. Stakeholder comments are processed/reimplemented, and the feedback process is repeated.

**Data presentation** Great care must be taken by the modeller when presenting data. Advances in computer graphics and data presentation allow for very visually impressive images. Non-computer experts tend to be overwhelmed by the amount of information presented and tend to assume that the model outcomes are 'the truth'. The modellers have a great responsibility to correctly explain the results, their validity and limitations.

**Two types of stakeholder feedback** When discussing the model outcomes with the modellers, stakeholders will have two main types of feedback, illogical model behavior and surprising behavior.

**Illogical model behavior** Stakeholders observe model outcomes that do not seem to make sense. The behavior cannot be explained by following through on the model logic. This can point toward model artifacts or logic implementation errors. These outcomes lead to a model debugging action by the modellers.

**Surprising behavior** Surprising behavior can be both positive (desired by the stakeholder) and negative (undesired by the stakeholder). Such observations can be explained by examining the model's logic. It leads to questioning of the modelling assumptions. Special care must be taken when the model yields desirable results. Stakeholders tend to be far less critical of 'pretty' model outcomes than 'ugly' ones. Observations of this type lead to a refining of the behavioral model of the agents to changes to the modelling assumptions by the entire team.

## 6.6 Conclusion

**Overview of the co-evolutionary method** This chapter presented the first three generations of the modelling method. The first generation was a deformation of the fitness landscape in the technical dimension. A functioning 'proof of principle' model template was constructed to demonstrate that the necessary abstractions needed to model an evolving  $\lambda$ -system can indeed be implemented as an ABM. The second generation focused on designing a social process to collect relevant domain knowledge. Relevant facts and knowledge were collected. In the third

generation, design problems identified in the social process in the second case were rectified, and an alternative technical implementation was attempted.

Each of these cases represents a learning step for the modelling method - the method progresses via three generations. The intermediate outcome of this evolutionary modelling method is the System Decomposition Method for collecting and formalizing knowledge from multiple stakeholders into an Agent Based Model. The theoretical foundation of the SDM developed in the Chocolate Game case study were discussed in Section 6.5 and provide the reader with a deeper understanding of the case studies to come.

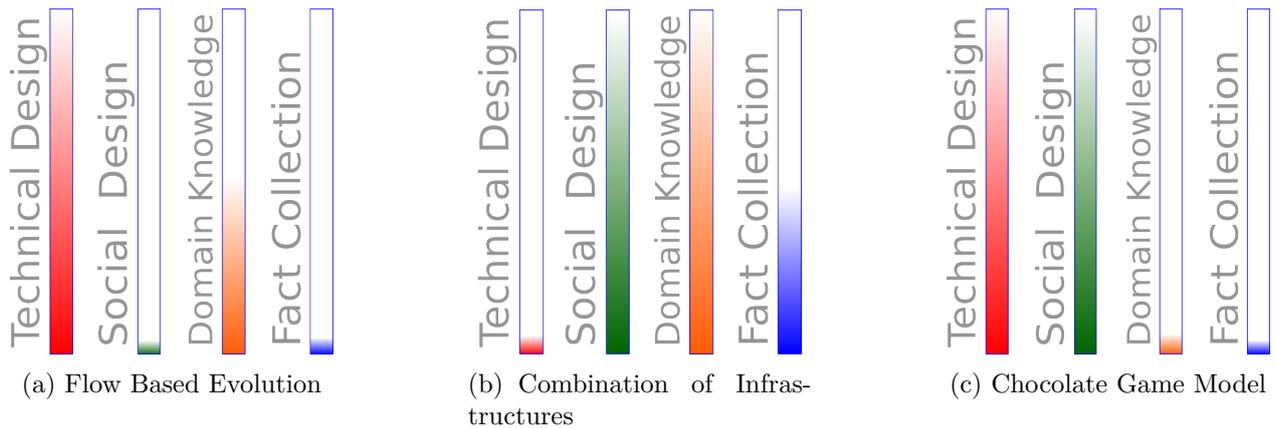


Figure 6.22: Learning case study method development overview

**Next steps** The next step is the Costa Due case study. It is the fourth generation in the co-evolutionary process and a full-scale study, combining all of the insights from model design, social process design, knowledge and fact collection.

## CHAPTER 7

### FULL-SCALE CASE STUDY: COSTADUE

You are an organism.

You need to eat to survive.

If you don't find food, you will die.

You live in an environment.

If the environment changes and you cannot adapt or resist, you will die.

Somebody will try to eat you.

The environment will change.

Welcome To Evolution.

But this is not what you think. This is not Discovery Channel.

This is an Industrial System.

You breathe money, oil, electricity and water.

You eat steel, concrete, plastics and information.

And when you flush CO<sub>2</sub>, dioxins, old car tires and landfills appear...

Welcome To Industrial Network Evolution.

**Case abstract** This chapter presents the first full-scale case study. It implements the complete SDM, presents a full-scale simulation engine and encodes a lot of knowledge and facts on chemical and bioprocesses. The study examines the evolutionary patterns of the transition of the Groningen Seaports region from a chlorine to a biobased cluster. The agents have modular and mass balanced descriptions of technology and realistic economic properties. Basic economic reasoning on contract selection and price determination is implemented. The transition is simulated by adding new biobased technological options, identified by stakeholders, to the simulation. The cluster evolution is studied under different economic selection pressures. The main conclusion is that the transition to a mainly biobased cluster is unlikely, given the current agent behavioral assumptions and gathered data. The evolutionary pattern is very path-dependant, and transition to biobased technologies is only likely for bioelectricity processes, assuming the continuing survival of incumbent industries. The main direction for future model improvement is to examine the model's behavior over much larger parameter settings, including a range of prices on the world market. This poses a technical challenge due to the number of simulation runs needed. Furthermore, more sophistication is needed in the description of agent decision-making, including, for example, the decisions for the agent about when not to join the cluster. A refinement in the cluster management strategies available to the RDA is also needed. At the modelling method level, the quest for encoding new formalisms and new facts continues.

## 7.1 Focus Point

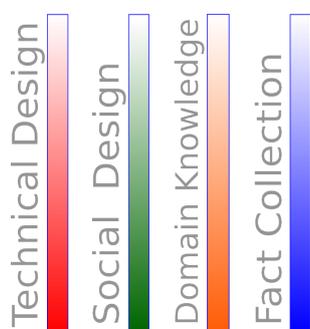


Figure 7.1: Case focus cluster.

**Methodological focus** This case study<sup>1</sup>, will focus on implementing all four aspects of the fitness landscape to the maximum. The social process design is implemented in full, an elaborate technical design of the simulation engine is tested, a large quantity of knowledge is formalized and many facts collected. It is the first comprehensive case study combining all the insights so far, and it creates a solid basis for future development.

**Case focus** The case is focused on modelling and understanding the evolution of a regional industrial cluster in the Dutch province of Groningen. It examines the possible evolutionary pathways for a transition from a petrochemicals and chlorine based cluster to a bio-based

## 7.2 Hypothesis

**First hypothesis** *A the method level, it is possible to combine the lessons learned in the learning cases and create a social process and simulation engine that provides an understanding*

<sup>1</sup> Part of this chapter's results, produced in the context of a MSc thesis study by Blokker (Blokker, 2006), supervised by the author. Parts have been published as (Nikolic and Dijkema, 2007). Parts have been adapted from a working paper "From many minds to modular models: Facilitating the modelling of complex adaptive systems" by I. Nikolic, P.J. Beers, G.P.J. Dijkema and P.Bots

*of the evolutionary patterns of  $\lambda$ -systems.*

**Second hypothesis** *Modelling the patterns of  $\lambda$ -system evolution using multiple formalisms will help the RDA examine possible evolutionary patterns when introducing different technological options in the region.*

## 7.3 Case Description

**Seaport regions** Around the world, seaport regions are hosts to  $\lambda$ -systems. Apart from transport hubs and container terminals comprising harbor infrastructures, these systems also comprise of energy infrastructure and industrial clusters. In many of these regions, Regional Development Agencies (RDAs) have been assigned the responsibility of sustaining the socio-economic and ecological prosperity of their respective regions (Dijkema et al., 2005).

**Effective Management** As discussed in Chapter 4.2, there are several properties of industrial regions, such as seaports and their infrastructures, that make them difficult to manage. Components and subsystems of an industrial network, such as production plant infrastructures, have average lifetimes measured in decades. They are also very capital intensive: the investment for a typical world scale chemical plant ranges from 100 to 1500 M euros. This makes any change to a subsystem very slow and very costly. Furthermore, there are multiple actors/agents that shape a system's evolution. Within companies, private decision-making processes determine which system will be installed where, when and how. Companies experience huge sunk costs because of past investments and are entrenched in a particular system. In addition, the social network may evolve over time when owners and operators enter or leave the scene.

**RDAs' challenge** As already mentioned above and in Chapter 1, managing  $\lambda$ -system evolution represents a formidable task for a Regional Development Agency (RDA). Not only must RDAs timely adapt their policies in a changing world, but also the very economic structure of their regions are shaped by decisions over which the RDAs have no direct control. For example, the EU setting new Directives or a national government completing major infrastructure in competing regions are all outside the scope of control of an RDA but crucial for a region's development. Companies are attracted to a region for the facilities it offers, and the RDA may influence these.

**Objective and setting** Referring to the definition of complexity given in Section 3.3, it is clear that regional industrial networks can be seen as Complex Adaptive Systems. We adopted this view in the RDA's problem definition for the case study: "How can we understand and manage a complex, adaptive, multidimensional network of agent interaction which lacks central control and that by definition requires multiple description formalisms?"

**CostaDue project** The author and the TU Delft research network have been involved with Groningen Seaports, the local RDA, the Groningen provincial government and other partners to explore the possibilities for accelerating the transformation of the harbor and industrial cluster into a bio-materials and bio-energy based cluster (Dijkema and Stikkelman, 2006). The stakeholder and problem owner in the study is clearly the Groningen Seaports, and more

specifically the strategic development managers of the Port Authority. The involved domain experts were from a variety of relevant disciplines.

**Approach** In order to aid the RDA, the approach based on the learning cases was used. Using the SDM led to an ontology of processing industry and to a joint problem formulation between the RDA and the researchers. Furthermore, the RDA and the social network involved in the development of the Groningen Seaports region (Dijkema and Stikkelman, 2006) defined a number of bio-based technological options that in their view are crucial for the future development of the cluster. These technological options were formalized in the ontology, and based on it an Agent Based Model was created. The model examines the patterns of the cluster’s evolution, given the appearance of the different conditions of the identified bio-based options.

## 7.4 Model Details

In this section the details of the constructed model are presented. The forms are modelled to have a mass balanced technology description and basic realistic economic parameters. Decision-making involves contract selection to feed the technology owned by the agents, and pricing mechanism is used to sell the products. The model also involves a World Market agent that acts as a sink/source for all economic flow, and an Environment Agent that acts as a source/sink for environmental flows. The section closes with a discussion of the scenario space available for examination.

This section will only discuss the details of the simulation, its main assumptions and the details of technology and behavior representation within the agents. The social process, in the form of the SDM, was extensively discussed in Section 6.5. The design of the simulation engine is presented in Appendix D.

### 7.4.1 Model Assumptions

There are a number of assumptions underpinning the model. These are:

**No harbor** Harbor activities are not modelled in the simulation, even though the harbor is a part of the Groningen Seaports region. As we are focusing on industrial network evolution, the harbor is only important in its ability to import and export goods.

**Firm as a plant** Agents in the simulation represent processing plants and are assumed to be owned by a firm. The model does not consider firms that own multiple processing plants. The external economic world is aggregated as a World Market agent, and the physical environment is modelled as the Environment agent.

**Scenario** Scenarios are used to describe World Market behavior and the RDA’s strategies. In the base case, we modelled firms appearing in the region at random, since the RDA suggested that it has a rather limited effect on who comes to the region.

**Static prices** The world market prices are assumed to be static.

**Components** There are six main components that build up the model. These are:

- Technology Agent;
- Decision-making;
- Model schedule;
- World Market;
- Environment; and
- Scenario.

These components will be described in more detail below.

## 7.4.2 Technology Agent

As can be seen in Figure 7.2, the Technology Agent consists of its State and its Behavior. These components will be discussed below.

**Agent** As already mentioned, firms are represented as agents. All agents use the ontology to reason about the world and communicate with other agents using the concepts from it. The agents are modelled to have only one goal, namely to survive. In order to survive, agents must have a positive cash flow. Agents are allowed a negative cash flow for a limited period of time, settable by the user. In order to maintain their cash flow, agents must purchase resources needed to feed the technology they own and they must sell the products of their technology. It is assumed that all processes are of such large scale that no stockpiling of products or raw materials is possible.

Two exceptions to the above description of an agent are the Environment and the World Market agents. These are discussed in Sections 7.4.5 and 7.4.4, respectively.

Figure 7.2 presents the layout of the Technology Agent.

As already mentioned in Section 6.5.4.3, an agent consists of a State and Behavior and interacts with others via mass & energy flows and contracts. Agents also ensure that all mass balances are conserved and all money is accounted for. First, the agent's state is discussed, consisting of two parts, Technology and Economics.

**Technology** Technology follows the basic engineering input/output paradigm as discussed in Section 4.3.1. The detailed description of the Technology ontology and the corresponding Java code is available online.<sup>2</sup>

Each technology, in addition to having physical and economic properties, has an Operational Configuration. Operational configuration is a normalized input/output table which, multiplied by the operational scale, gives a mass balanced description of the technology. Operational configuration consists of a set of Component Tuples<sup>3</sup> defining the inputs and outputs. A

---

<sup>2</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Node/PhysicalNode/Technology.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Node/PhysicalNode/Technology.html) and <http://gux.tudelft.nl/svn/SimulationGenerics/trunk/src/simulationOntology/Technology.java>

<sup>3</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Data/DataTuple/ComponentTuple.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Data/DataTuple/ComponentTuple.html)

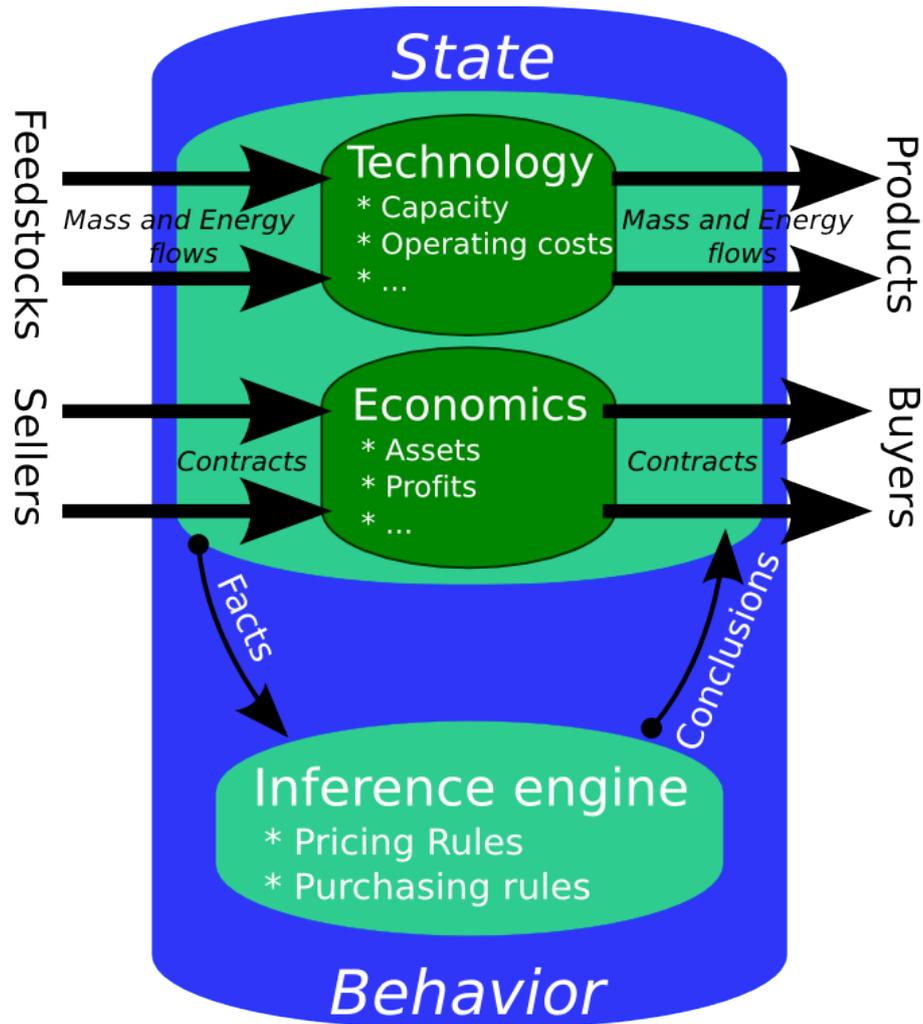


Figure 7.2: Structure of the TechnologyAgent

component tuple is a collection of data fully describing the flow. It is a vector consisting of a Good Name, a Relative Amount, a Unit and a Function Label. An example of an Operational Configuration describing a biomass gasifier is presented in Figure 7.3.

All operational configurations are normalized to the Reference Product. Stream labeling categories are classified using the Functional Label. <sup>4</sup> defined by Dijkema (Dijkema, 2004)

**Economics** The second part of a Technology Agent’s state is the Economics. The agent’s economic properties are defined as a list of Economic Property<sup>5</sup> objects. These are, for example, capital assets, debt, etc. Together with the economic properties of the Technology object, the entire economic state of an agent is described. These states are used as facts in decision-making, for example for setting prices of goods.

<sup>4</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Data/Property/Label/FunctionLabel.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Data/Property/Label/FunctionLabel.html)

<sup>5</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Data/Property/EconomicProperty.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Data/Property/EconomicProperty.html)

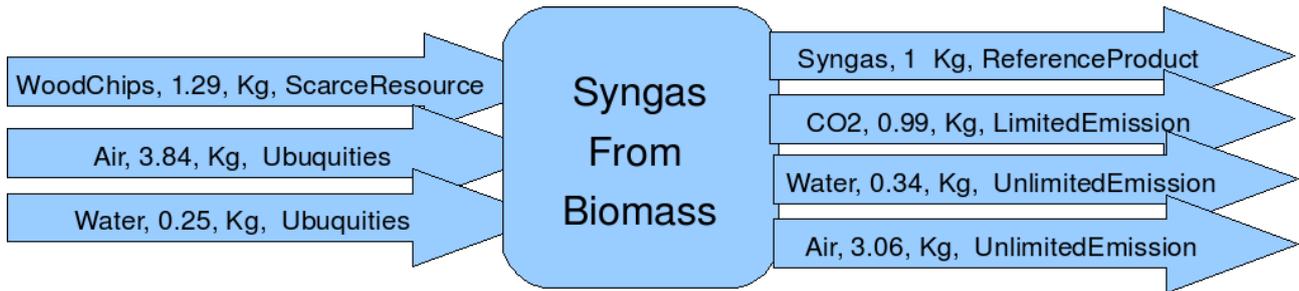


Figure 7.3: Structure of an Operational Configuration, defining a Technology as used by the Agent

**Behavior** An agent’s state is insufficient to fully describe it, as it lacks decision-making processes. We have defined its body and limbs as it were, but not its mind. An agent’s behavior, discussed previously in Section 4.3.1, is the overall emergent behavior resulting from interactions between the agent’s internal decision-making processes. The ‘mind’ of the agent is modelled as a forward chaining inference engine using the Rete algorithm (Forgy, 1982). It is implemented as JBoss Rules (Proctor et al., 2007) objects. A forward chaining inference engine consists of a set of if-then rules that are applied to a set of facts. The engine infers which rules ‘fire’, i.e., are evaluated to be true. Rules can create new facts, which in turn might fire new rules, etc. An inference engine can be compared to a logic net, filtering through a sea of facts, catching those that apply and inferring consequences of those ‘caught’ facts. The decision processes that agents have are proactive and reactive. First, agents make proactive decisions about purchasing resources, and secondly, agents make active pricing decisions when asked to offer their products for sale.

**Purchasing** Purchasing decisions consist of reasoning by the agent about the technology they owned the type and quantity of the resource needed. Once this is done, an agent places a call for offers in the virtual market and receives potential contracts. Once contracts are received, they are treated as Facts, entering the inference engine. Using these facts and the contract selection rules, the agent will select an appropriate contract. The most basic contract selection algorithm chooses the cheapest contract, up to the available amount of what is needed. If the amount of the cheapest contract is insufficient, it will also accept the second cheapest offer, until its mass requirement is met. The purchasing rule set is available online. <sup>6</sup>

**Sales** On the reactive, pricing end, an agent receives a request for a contract from the market, it reasons about the technology it owns to determine whether the type and amount can be supplied and determines the price for the resources. A technology object itself and its economic properties are considered as facts by the reasoning engine. Several mechanisms for price setting are implemented. The most simple one is a cost price plus method, where an agent reasons about its technology, determines the cost price and adds a fixed margin. <sup>7</sup> A more advanced model is the perfect market, a perfect information model in which an agent reasons about the

<sup>6</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/ContractSelectionCheapest.drl>

<sup>7</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/CostPricePlus2.drl>

availability of a certain resource from competition and either offers a price just below the World Market price or attempts to recover operational costs by setting the price just above the cost price. This algorithm is presented in Figure 7.4. The rules used by the inference engine are available online.<sup>8</sup> Finally, an adaptive algorithm was implemented, whereby the agents learn from their past profits and adjust their margins accordingly.<sup>9</sup>

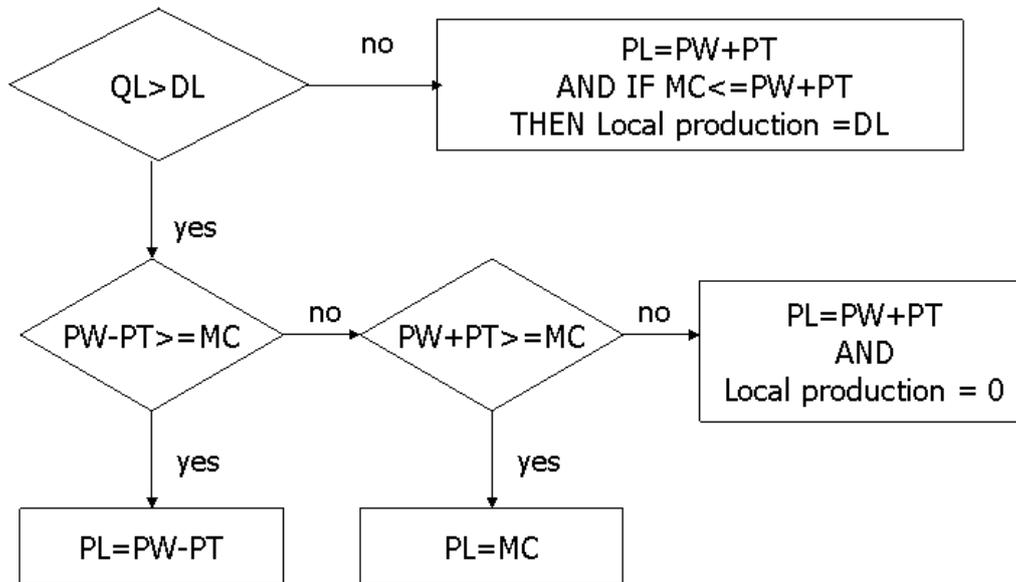


Figure 7.4: Agent’s algorithm for setting a price of a good, agent assuming a perfect competition situation. For meanings of abbreviations, see the paragraph below.

In Figure 7.4, the following abbreviations are used: QL: local production capacity = sum of all local production capacities; DL: local demand = sum of demand of all local agents (considered a scalar, for now; should be a demand function); MC: local marginal costs of production; PW: world price; PT: price of transport from local area to world market; PL: local price.

#### 7.4.2.1 Multidimensional Interaction

The basic agent formalisms described in Section 6.5.4.3 presented Edges as the only interaction between Agents. The Technology Agent presented in Figure 7.2 shows two subclasses of Edges<sup>10</sup> used in interaction. The first is the Physical Edge and its own subclass Physical Connection.<sup>11</sup> This type of edge connects technologies and represents mass or energy flows. Interaction between agents themselves is of the type Social Edge and its subclass Contract. As a result of decision-making, a Contract is created between agents, which in turn causes a Physical Connection to be established, across which a Physical Flow is established. So the network

<sup>8</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/PerfectCompetition3.drl>

<sup>9</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/ChangingMarkup3.drl>

<sup>10</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Edge.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Edge.html)

<sup>11</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/\\_THING/Edge/PhysicalEdge/PhysicalConnection.html](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/_THING/Edge/PhysicalEdge/PhysicalConnection.html)

created by Technology Agents has three dimensions which, while closely correlated, are fully distinct. Just as in the Chocolate Game model, presented in Chapter 6.4, all edges are valid for only one turn. Expansion for long-term contracts and connections is trivial. These physical connections can only be established if a social connection, a Contract, has been established between the agents owning the technologies. The network created by interacting agents is thus multidimensional, with each dimension affecting the others.

### 7.4.3 Model Schedule

Agent Based Models are discrete simulations that attempt to model the inherent parallelism of the real world. This parallelism is achieved by having discrete time steps, between which no time is thought to pass. Everything between two ticks of the clock is assumed to happen at the same time. The Scheduler, an integral part of Repast, is responsible for initiating those activities. Figure 7.5 presents a schematic representation of the schedule used.

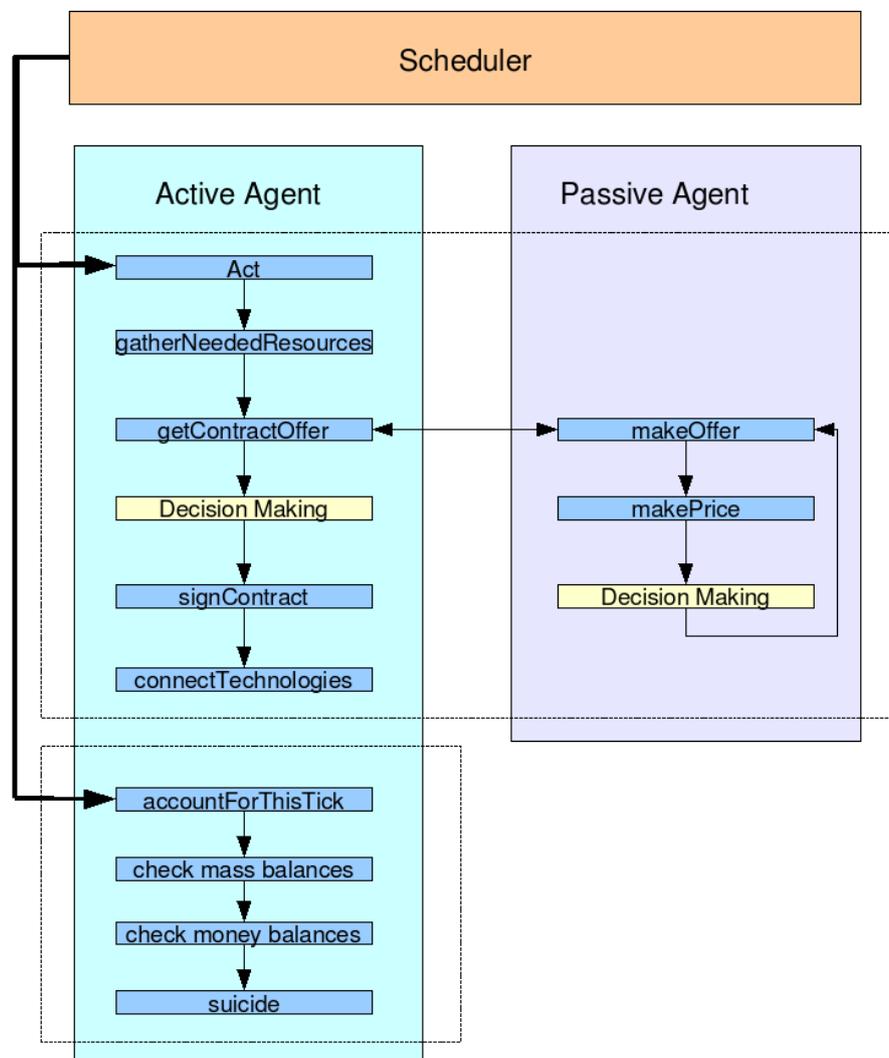


Figure 7.5: Agent activities scheduled by the simulation

**Active and passive** At each time step the scheduler will iterate over the list of agents, making them active and performing their actions. All other agents are passive and only respond to queries by the active agents. The order of iteration is randomized each turn to prevent first-mover advantage artifacts.

**Housekeeping** In the second part of the scheduled actions, the scheduler will cause the necessary housekeeping activities to be performed. These actions check the mass balances, perform economic accounting and, when necessary, remove bankrupt agents.

#### 7.4.4 World Market

As already mentioned, the World Market is a special case of an agent. It is an abstracted aggregate of all other firms in the world that produce and sell goods needed by the cluster. The World Market may have its own internal price dynamics and is not influenced by the agents in the cluster. Depending on the conditions in the cluster, there might be a significant difference in prices. The World Market serves as the infinite economic source and sink for goods that are produced or needed in the cluster.

#### 7.4.5 Environment

The Environment agent is another special case of the Agent. The Environments 'buys' all streams that are labeled as 'waste' in the ontology and that nobody else wants to buy, such as CO<sub>2</sub>. Furthermore, the Environment offers for 'sale' all goods that are labeled as ubiquitous, such as air and water. These environmental emissions and extractions are modelled as physical flows with associated contracts having a price of 0. In this way an accurate mass balance of extractions and emissions to the physical environment is possible. This approach allows complex ecological impact modes to be simply modelled inside the Environment, being dependent on the emissions and extractions of the cluster.

#### 7.4.6 Scenario

**Scenario** The final part of the model is the Scenario, which describes the environment that the agents 'live' in. These scenarios describe the economic environment in the form of world market prices of different goods and their behavior over time. Furthermore, the scenarios describe the type and order of agents added to the growing cluster.

**Scenario components** The Scenario<sup>12</sup> consists of a number of methods (functions) specifying the behavior of the environment the agents live in. Scenario elements are:

**addAgents** Which agents are to be added before each time step?

**agentsToKill** Which agents are to be killed after the step has finished? Economic selection criteria are defined here.

---

<sup>12</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/Scenario.java>

**assertObjectsDecidingOperationalConfiguration** Determines which decision algorithm is to be used by the agents when making decisions about changing technologies or operational configurations (operational decisions)

**assertObjectsForInvestmentDecisionMakingByExistingAgents** Idem for decisions about investments.

**doThisEachTickAfterAllAgentsAct** What should be done after all the Agents have performed their main actions, defined in the `act()` method (function).

**doThisEachTickBeforeAllAgentsAct** What should be done before Agents will start doing their main actions, defined in the `act()` method?

**extraAgentAccounting** All the accounting/post behavior maintenance actions that need to be performed on an agent.

**generateStartingAgents** The agents that should be used to start the simulation.

**getInitialWorldMarketPrices** Setting the world market prices of the goods that exist in the simulation.

**setLocalDemandProfileForGoods** This allows a local demand profile for goods to be set up in the scenario.

**setLocalSupplyProfileForGoods** Idem for supply.

**Scenario space** Because of the way the scenarios have been set up, it is possible to generate a vast scenario space consisting of ranges of model parameters and agent decision-making models. Instead of generating and testing several predetermined scenarios (fixed parameter values), we can explore wide ranges of parameters and observe models responses across this space.

## 7.5 Experiment Details

This section outlines the experiments performed with the model. At the method level, the experiment involves using the SDM created earlier and learning from the method. The simulation engine, developed based on the experiences from the learning cases, is implemented and tested. At the case level the basic scenario of adding bio-based agents to an existing chlorine cluster under different economic selection pressures is tested. The metrics used to measure the multidimensional emergent network are discussed.

**Two levels** Based on the hypotheses presented in Section 7.2, two types of experiments will be performed. The first experiment is at the level of the co-evolutionary modelling method, testing the social process and simulation engine designs, and the second is a case-level experiment.

**Metamodel** The modelling method-level experiment is performed by the development and execution of the SDM and by the design and creation of the simulation engine. This is an experiment that tests the social and technical aspects of the methodology (see Chapter 5. This case presents the first execution of the refined SDM, and where possible learning points for further development will be presented. The case also tests the simulation engine design elaborated in Appendix D.

**Case content level** The second level experiment is a more traditional type. We collect knowledge and facts about the world, create a model and examine the model's behavior. It consists of a series of experiments to test the model's behavior and answer the RDA's questions.

In this section I will focus on the case content level experiments, since the description of the outcomes at the modelling method level are presented in the previous sections.

### 7.5.1 Types of Experiments

**Exploration of futures** We consider an agent's profile and behavior to be fixed and an agent's environment to be variable. This allows us to explore patterns of possible futures by examining an agent's responses when subjected to different scenarios of environmental dynamics. By sweeping the environmental parameters across the full scenario and parameter space and by mapping the system's attractors (Lorenz, 1963), we can get a sense of the range of possible agent cluster behaviors. This give the RDA a 'what if' scenario testing tool and a map of potential future states of the cluster.

**Base scenario** The scenario run is the basic experiment performed with an Agent Based Model. One sets up the dynamics of the environment and observes the reaction of the agents. Part of the scenario definition is the way agents are added to the simulation. RDA experts suggested (Hotsma, 2006) that they have very little control over the order and nature of firms that wish to enter the region. Therefore, the order of agent addition is assumed to be random. We have tested scenarios where 1) a mix of existing technologies and bio-based technologies is allowed to join the cluster, and 2) a purely bio-based mix is allowed to join. In the case of world market prices, we assumed that they either remain stable or rise in a cyclical fashion over time. This reflects the cyclic nature of the prices of base chemicals. In the base case the novel bio-based technologies are added randomly to the cluster over time. Each agent experiences selection pressure, that is, it will die if has negative cash flow for 20 turns. The prices on the world market are assumed to be static.

**Parameter sweeps** The most basic parameter analysis is the one-dimensional parameter sweep, where model outcomes (either individual agent behavior or system level properties) are calculated over time for each parameter value. This gives a time trajectory of the system, depending on the parameter values. It shows the system's dynamic response to parameter changes.

### 7.5.2 Metrics

**Basic network metrics** The limitations of graph theoretical network metrics was discussed previously. However, it is still important to characterize the evolved networks. The most basic

metrics are the absolute size of the network in number of nodes, the number of unconnected nodes and the shortest average path length. Since we are evolving regional clusters, the number of independent clusters within the network is also of interest. The shortest average path length is also measured.

**Agent’s population metrics** This group of metrics gives insight into the population of an agent and its demographics. *NumberOfAgentsAdded* measures how many technology agents have been added throughout the whole simulation up till now. The *NumberOfAgentsTotal* tracks how many of those added agents still exist. *NumberOfUnconnectedAgents* measures how many agents do not have any connections to other agents but are only connected to the world market and the environment. *NumberOfAgentsInLargestCluster* counts the number of agents are in the largest cluster.

**Cluster Focus metrics** The aim of the model is to examine the growth of a cluster. That means, among others, understanding how the cluster is focused. Is the cluster focused internally, mainly trading with each other, or is it externally focused? We measure three aspects: mass fraction, money fraction and contract fraction. Mass is expressed as the percentage of the mass that is flowing internally with respect to the total mass flow, money fraction expresses the percentage of money that is transacted internally, and the contract fraction is the number of contracts between cluster members divided by the total number of contracts in the simulation.

**Cluster performance metrics** The model is fully mass balanced, so we can use mass flow proxies as environmental performance indicators. CO<sub>2</sub> production is the most interesting, since it says a lot about energy intensity and the renewable nature of the network. The CO<sub>2</sub> metrics measure the cluster’s total CO<sub>2</sub> emissions. The Cluster Capital metric monitors the economic performance by looking at the cumulative net amount of money that is made by the cluster. By monitoring these two metrics the tension between environmental and economic goals can be studied.

**Individual agent metrics** In addition to cluster metrics, individual agent metrics are collected. The first metric that can be followed is the margin or marginal costs per unit of production. It gives a sense of whether the agent is running a profitable business. The second individual metric is the profit the agent makes per simulation step per Euro of invested capital. The last one is the cumulative capital of the agent. At the start of the simulation, each agent has to invest the construction cost in order to establish a production facility. Thus the agent’s starting capital is negative due to the construction costs.

## 7.6 Case Study Results

This section presents the results of the experiment at two levels. At the method level, the results are on the application of the SDM and on the implementation of the designed simulation engine. In short, both work ‘good enough’. Several improvements are possible. At the case level the dynamic of the evolving cluster is examined. This dynamic is tested across varying levels of economic selection pressure. It is found that a purely bio-based cluster is unlikely, and that

high economic selection pressure, while disadvantageous in the short run, pays off in the long run.

## 7.6.1 Results of the SDM

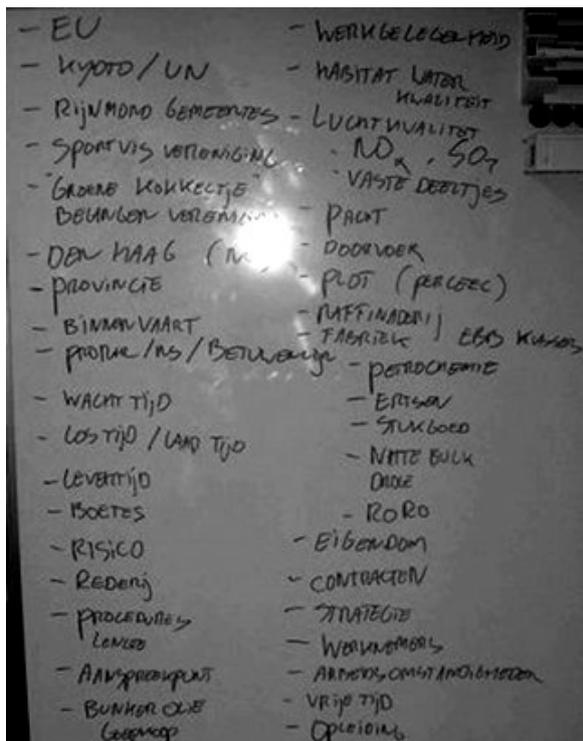
The SDM has been described in detail in Section 6.5. The results of its application are presented in this section. Only the outcomes of Facilitators' States 1 and 2 and the Stakeholders' State 3 will be presented, as they have have interesting, non-trivial outcomes.

### 7.6.1.1 Facilitators' State 1

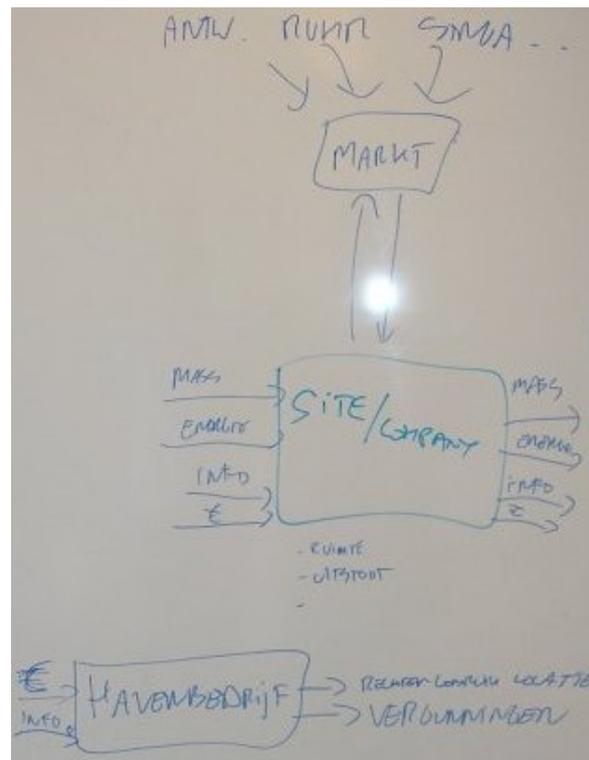
Section 6.5.4.3 discusses the method and the result of creating the facilitators' shared formalism. It is important to realize that this is a very iterative method. It has been repeated several times with smaller projects, in order to reduce the number of basic concepts to the bare minimum while retaining the maximum expressive power. An example of such an earlier iterative step is presented in Chapter 6. The fact that the modellers have extensive previous experience with ABMs greatly increased the speed of development of the basic formalism.

**SDM script** Below we present the results of the script that is executed to in order to facilitate the transition between Knowledge States 1 and 2 and the Facilitator's State 2 (see Figure 6.18). These will be presented per step of the script.

**Inventory** The result of the inventory step is presented in Figures 7.6.



(a) Example output of the inventory step



(b) Example output of the structuring step, Node and Edge identification

Figure 7.6: Example of the inventory and structuring steps

**Structuring** The components identified in the Inventory phase are first grouped in classes: Nodes, Edges, Data and Knowledge. Afterwards they are linked to each other. An example of the structuring output is presented in Figure 7.6.

**Linking Agents and Interactions** Given the created structuring, in this step a connection is made between the Agents and Interactions identified to date. From here the basic structure of the knowledge becomes visible.

**Iteration** In the Iteration step one has to decide if things from the inventory are not important after all. Things that were missing from the initial brainstorming list that resulted from the Inventory phase are discovered here too.

**External world** In this step the world outside of the agents is determined by grouping all the system components that cannot be influenced by the other sub-components. They form the External World. At this moment in the SDM, the stakeholders and domain experts have shared their knowledge in an unformalized way. It is now the facilitator's turn to formalize this knowledge in an ontology and create a soft-hard interface to begin collecting and formalizing this knowledge.

### 7.6.1.2 Facilitators' State 2: Conceptualization and Ontology Creation

Given the shared, unformalized knowledge collected in the previous step, it was possible to start creating the formal ontology. The theoretical background to this method was discussed in Section 6.5.4.4. Here we will present the practical, case-specific issues.

**Domain knowledge** In this step a number of process engineering experts were consulted and their knowledge used to extend the ontology and give it the ability to express concepts related to any technology that can be defined in the input-output fashion. Extensive process stream classification was added from (Dijkema, 2004). Labeling streams as Primary Inputs or Unwanted Waste became possible. Furthermore, economists were interviewed. Based on these interviews, basic economic concepts like price, development, maintenance costs, etc. were added to the ontology. Furthermore, abstract concepts such as decision-making were introduced.

**Outcomes** The outcome of the method, the formalized ontology, is available online in HTML format.<sup>13</sup> As an example of the output of this state, we present the Agents and Technology description through the Protegés user interface in Figure 7.7.

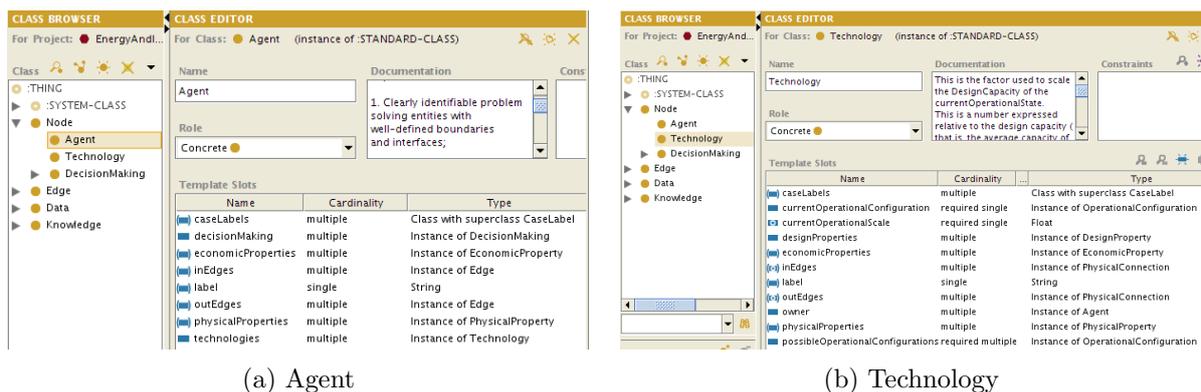


Figure 7.7: Protegé Representation of Agent and Technology

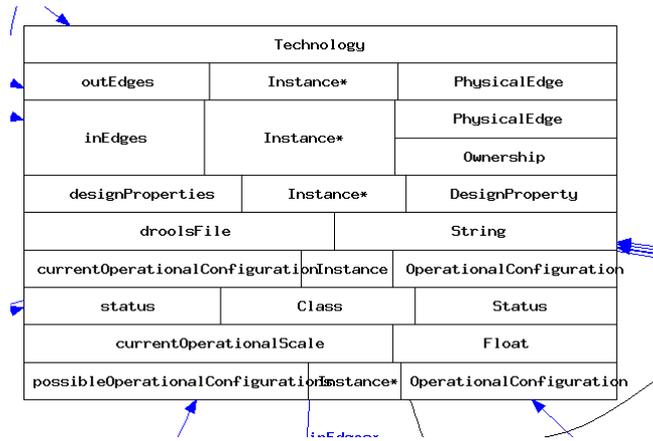
These two object types are fundamental to the CostaDue case, as the most important components of the model - Firms - and their technologies are represented by these two classes.

**Size and complexity** While Figure 7.7 shows the structure of the relationship between the Agent and Technology concepts, it gives us no information about the structure of the subclasses, nor does it give a sense of the overall size and complexity of the ontology. In order to provide a sense of scale, Figure 7.6.1.2 presents the ontology of Technology and its place within the overall structure. Note that the red circle marks the Technology class description to give a sense of scale. A full-scale image is available online.<sup>14</sup>

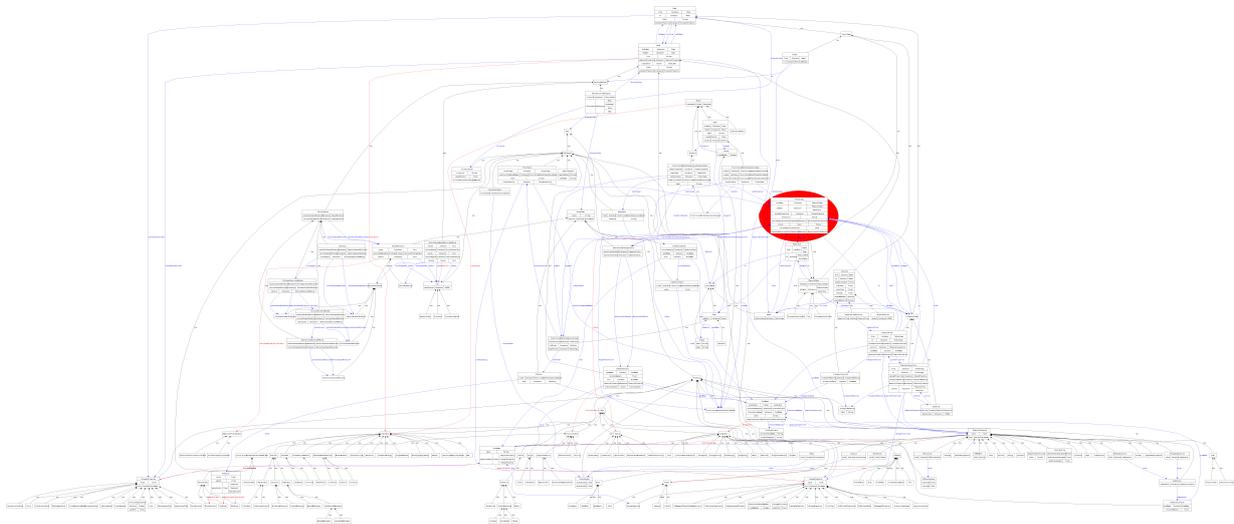
At this stage, the ontology was fairly complete, but did not yet contain any instances. In the following steps, the necessary instance of Agent and Technology classes are identified and encoded.

<sup>13</sup><http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/ontology/index.html>

<sup>14</sup><http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/AgentOntologyExpanded.png>



(a) Ontology of Technology



(b) Entire ontology

### 7.6.1.3 Stakeholders' State 3

**Scenario space** Facilitating the transition to a bio-energy and bio-materials based cluster is the goal of the CostaDue project. In order to visualize the transition paths and identify concrete steps that must be taken, a “Crosslinking conference” was organized by the RDA and the Groningen provincial government (Dijkema and Stikkelman, 2006). The conference was meant to increase stakeholders’ interaction and create a list of technologies that the group believes would form a successful bio-based cluster. It was the stakeholders’ belief that a mix of these technologies would form a stable bio-based cluster. This mix creates a part of the scenario space, namely the types of agents that are to be added during the simulation. The identified technological options are presented in Figure 7.8.

**Future technology options** These technology options are encoded as instances in the ontology and will be subsequently used in the simulation to explore the cluster growth. It is important to note that the group focused on what technologies would be good to have and not necessarily on what is realistic. We have taken these technologies, and using public data we created corresponding agents in the ontology. Sources were mainly process engineering literature

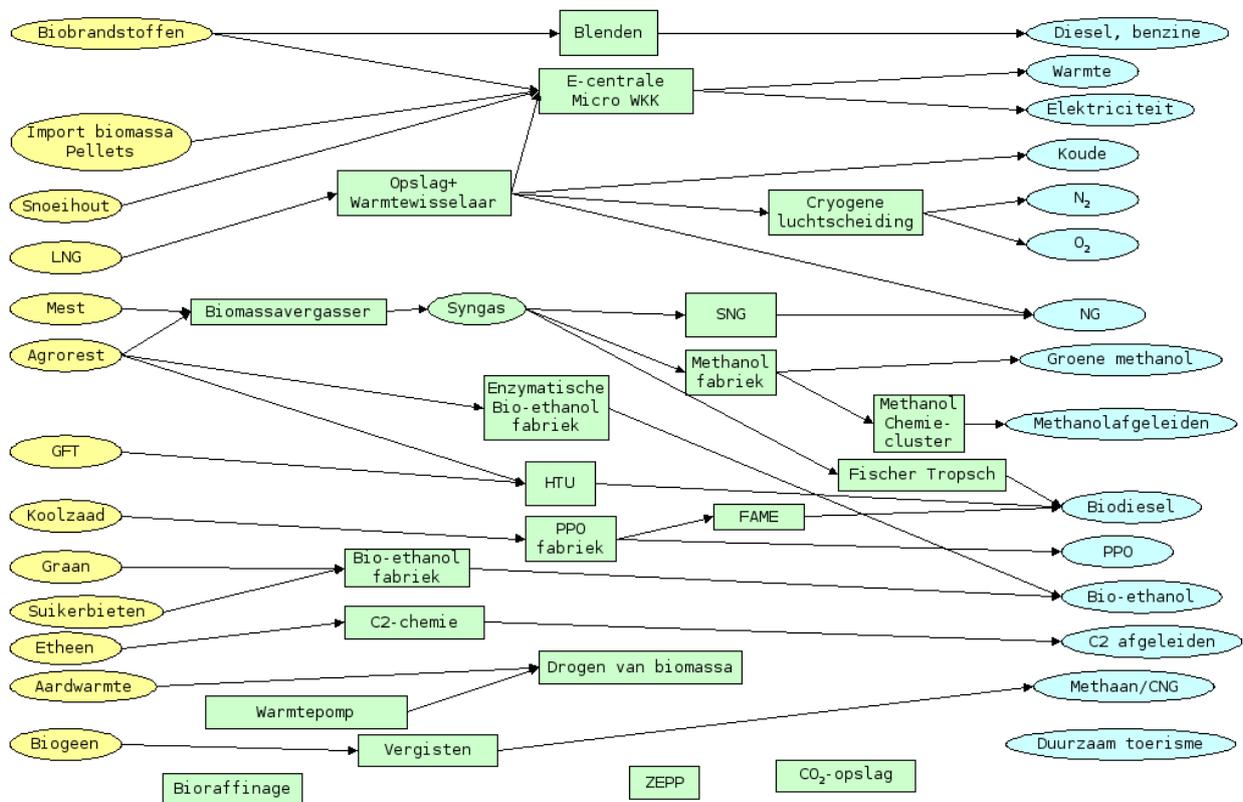


Figure 7.8: Options for cluster development, as identified by the stakeholders at the “Crosslinking conference”. (Dijkema and Stikkelman, 2006)

and environmental permits of the Groningen province.

**Limitations** There are two important limitations to this approach. First, a number of technologies does not exist yet, and their operating parameters, in and out flows had to be estimated. While these estimates are rough, engineering domain experts are able to estimate the figures within an order of magnitude. The second major problem is with the economic parameters. These are either proprietary information, and thus unavailable, or simply unknown. The lack of economic sanity checks, analogous to mass and energy balances, makes it very hard to estimate economic parameters. We estimated them either through using published values on similar technologies or by consulting experts.

## 7.6.2 Simulation Engine

The engine used evolved from a one-off simulation presented in Section 6.2, via a modular but conceptually inadequate model in Section 6.4 to a full-scale engine, designed from reusable, open source components from the hardware through knowledge creation. The design and the associated requirements are presented in Appendix D.

## 7.6.3 Simulation Results

In this section an example of a typical simulation run will be presented. The experiment setup was presented in Section 7.5. Additional results, not discussed in the thesis, are available online.<sup>15</sup>

### 7.6.3.1 Single Run

In order to help the RDA, a number of different scenario runs were performed. The most important one is the need to understand the viability of a bio-based cluster. In this scenario, the simulation is initialized with the agents already present in the cluster, and agents representing the identified bio-based options are randomly added. The modelled world market prices rise cyclically. If an agent is unable to keep positive margins for 20 turns (each turn corresponding to a quarter) it will die. The total model run is 200 steps, or some 50 years.

**Cluster self-organization** In order to verify whether the model is able to represent the structure of the industrial cluster reliably, agents that exist in the current cluster are added and allowed to self-organize. This is presented in Figure 7.9. If we compare the emergent structure with the linkages in Delfzijl, we can see that they are the same. This indicates that both the technology is accurately modelled and the economic conditions are accurately described, since the agents choose to connect in the same structure.

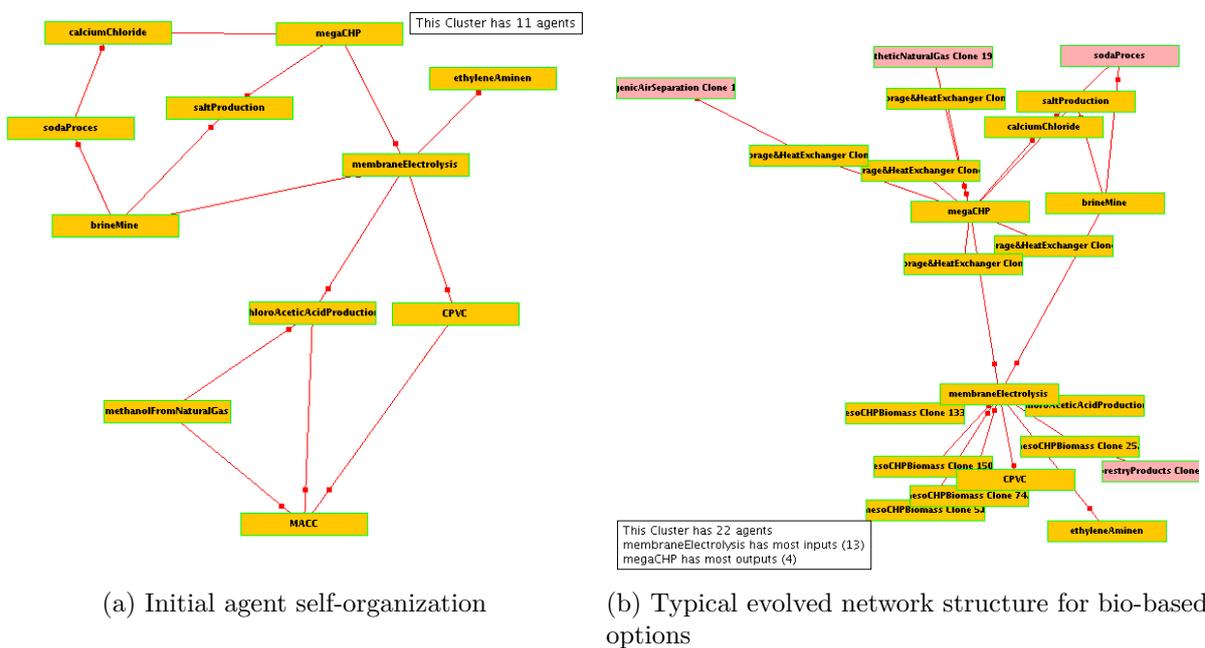


Figure 7.9: Initial and typical network structure

**Emergent structure** After the cluster is initialized, the bio-based options are randomly and repeatedly added to the cluster. Over time, this yields a cluster structure laid out using the

<sup>15</sup><http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/data/CostaDueData/index.html>

Fruchterman-Rheingold algorithm (Fruchterman and Reingold, 1991) as presented in Figure 7.9. We observed that the bio-based options alone are not able to form a stable cluster, not even upon repeated addition of agents of the same type. Only very few agents are able to survive. The ones that do are all bio-energy producers from biomass, or suppliers of biomass to bio-energy plants. It is important to note that the MACC and Soda Process die very early on in the simulation, which greatly affects the network structure.

**Example statistics** During cluster evolution a large number of statistics is collected. Below, the cluster's total CO<sub>2</sub> emissions and its mass and money balances as well as the network diameter are presented in Figure 7.10<sup>16</sup>.

**CO<sub>2</sub> increase** In Figure 7.10 a, we observe a sharp increase in total CO<sub>2</sub> production, followed by a rapid decrease and finally a leveling out. This is caused by the addition of several large energy producers and their subsequent removal. The leveling of the CO<sub>2</sub> emission is caused by the addition of mainly CO<sub>2</sub> neutral, bio-based energy sources.

**Flow fractions** In Figure 7.10 the evolution of the interconnectedness of the cluster has been visualized. To this end, the fractions of money, mass and contracts, respectively, that flow internally have been tracked. Internal flow is defined as not flowing to or from the World Market or the Environment. The most interesting result is that even while the number of agents is steadily increasing, the fraction of internal flows remains roughly constant. This demonstrates that agents must find local suppliers and clients in order to survive in the cluster.

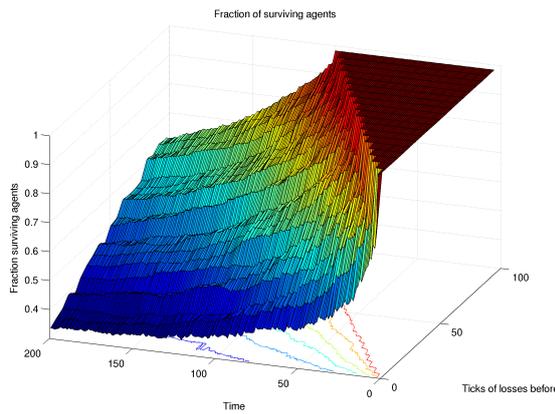
**Network size** Furthermore, we have tracked the shortest average path length for the cluster using Dijkstra's algorithm (Dijkstra, 1959). The metric is stable between 1.5 and 1.8, which is relatively large for a network of such small size.

### 7.6.3.2 Economic Selection Pressure

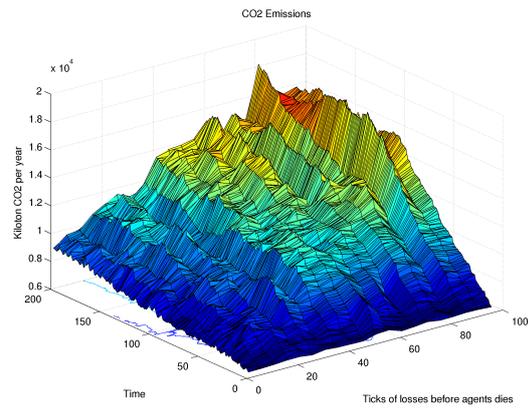
The main question the RDA has is to understand whether subsidies can support the development of a bio-based cluster by allowing the new entrants to the cluster to be non-profitable in their initial years? This experiment examines the role of economic selection pressure on the network's evolution. The selection pressure is expressed by the parameter "HowManyTicksOfLossesBeforeDie". If an agent has negative margins for a period of time longer than specified by the parameter, it will be removed from the simulation and "die". The model has been run for 200 steps, with "HowManyTicksOfLossesBeforeDie" varying from 1 to 100 time steps. The results are presented in Figures 7.11 and 7.6.3.2.

---

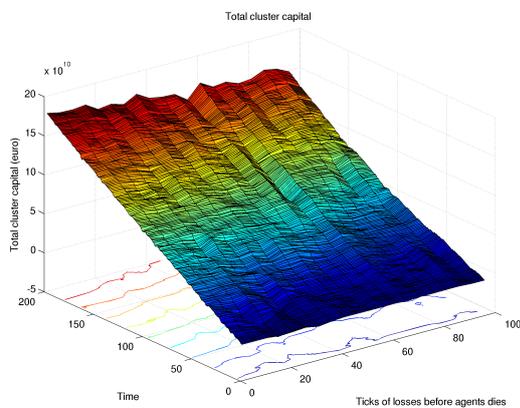
<sup>16</sup>For typographical reasons the figure is located at the end of the chapter.



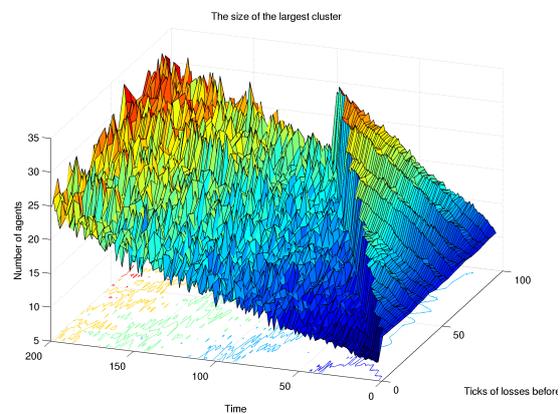
(a) Fraction of surviving agents



(b) CO<sub>2</sub> emissions



(c) Total cluster capital



(d) Largest cluster size

Figure 7.11: Metrics as a function of the economic selection pressure. Part I

**Survival fraction** There are several interesting results to observe. Figure 7.11 a shows a clear correlation between the initial length of agent survival and the economic selection pressure. After the initial removal of unprofitable agents, that selection pressure is far less pronounced, since most of the agents that survived the initial selection are profitable and remain in the cluster. Since the metric is a fraction of the total, it will stabilize at a limit value.

**CO<sub>2</sub>** The CO<sub>2</sub> graph shows an interesting attractor change. At severe economic selection (low number of loss ticks before death) the CO<sub>2</sub> emissions remain low over time. As the economic conditions are relaxed, the CO<sub>2</sub> emissions rise stepwise, although not very strongly. This is caused by the increased survival chances of bio-based agents that form a larger portion of the cluster but contribute disproportionately little to overall CO<sub>2</sub> emissions.

**Cluster capital** Cluster capital presents an interesting outcome. It seems that the overall cluster performance is only weakly related to the economic selection pressure. At low selection pressures many bio-based agents survive, but their economic added value is relatively low. The cluster capital rises steadily with time. This can be explained by the fact that weak agents

that are unable to make a profit contribute very little to the overall cluster, and their removal is hardly noticed.

**Largest cluster** Increasing economic selection pressure clearly causes damage to cluster size. We can see that as the selection pressure reduces, the largest cluster is free to grow to ever increasing size, until the economic 'grim reaper' removes all the weak firms. However, the overall trend is still rising over time, suggesting that even under strong selection pressure the cluster will rise, albeit remaining poorer in diversity. We can also observe that the largest clusters are formed under the least stringent economic selection pressures.

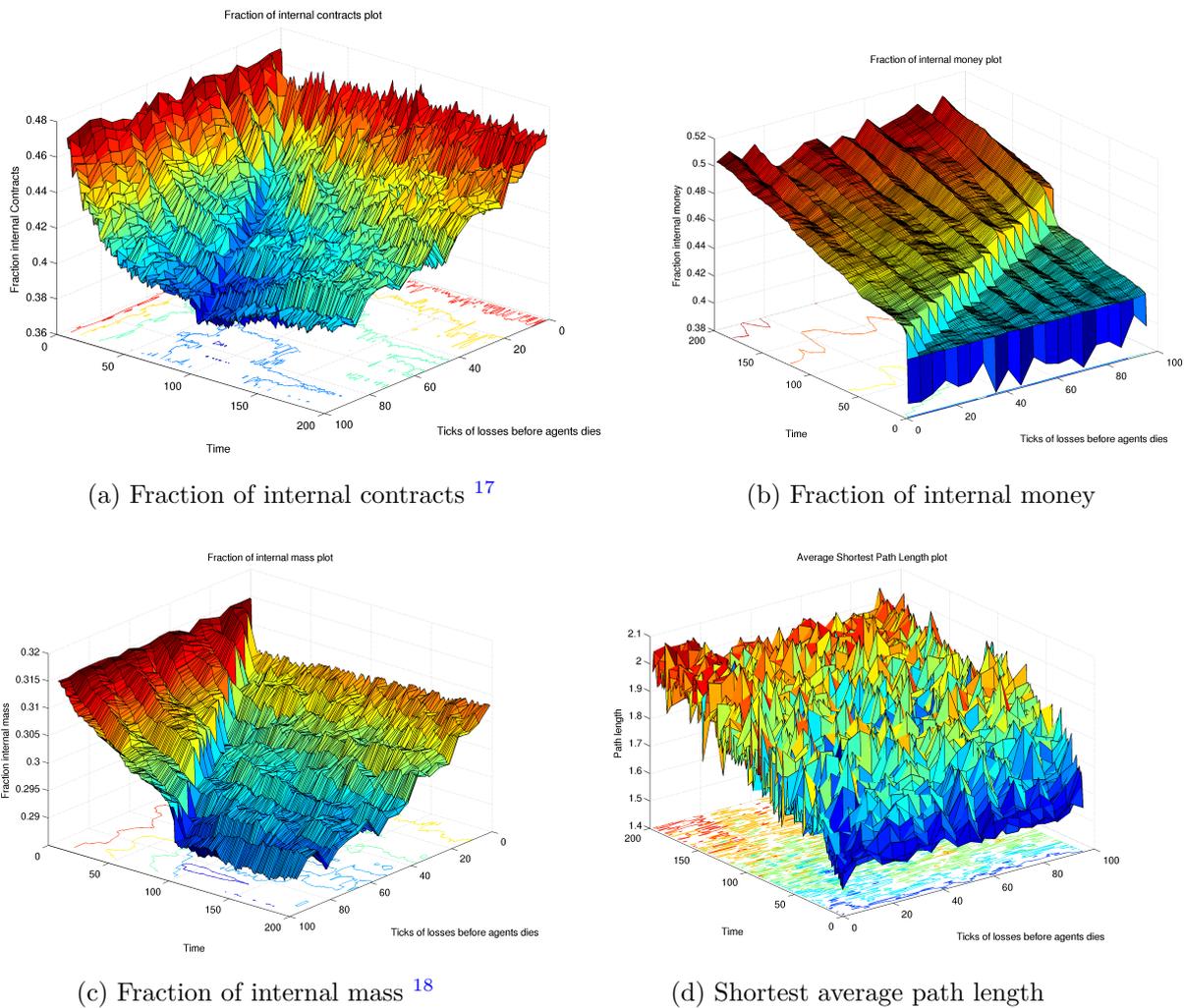


Figure 7.12: Metrics as a function of the economic selection pressure. Part II

**Contracts** Please note a rotation of axes of Figure 7.6.3.2 a in order to improve its legibility.

We can see that at the higher selection pressures, fewer and fewer external contracts are formed over time and the cluster becomes more internally focused. At low selection pressures, the cluster is more externally oriented, as all the 'weak' agents that cannot be well integrated with the cluster trade mainly with the world market. Looking at the money fraction graph in

Figure 7.6.3.2 b, we can see that at high selection pressures the internal money fraction rises, meaning that the external contracts lost are low value ones.

**Money** We can observe a phase change when the selection mechanism is activated. When the selection is first applied, agents that trade mainly with the external world are removed, increasing the internal orientation of the cluster. The apparent contradiction with the previous graph is explained that most contracts with the external world are of low value, so even if the cluster trades more with the World Market, the total amount of money within the cluster slowly rises.

**Mass** The mass flow correlates with previous findings. High value, low volume mass goods remain within the cluster. This is mainly caused by the electricity producers clustering around large incumbent consumers. The low value, high mass bio-fuels are sourced from outside the cluster, and the high-value product without mass - electricity - stays inside.

**Network diameter** The network diameter remains relatively unaffected over all values of economic selection pressure, since the cluster develops preferential attachment of bio-energy production to electricity users, not increasing the overall diameter of the network. This is confirmed by Figure 7.9 b, where we see the salt electrolysis plant being directly supplied by a large number of (small) bio-electricity firms.

## 7.7 Domain-Specific Insights

In this section the insights specific to this case study will be discussed. In summary, the emergence of a pure bio-based cluster is unlikely. A bio-energy cluster can develop only if the incumbent energy users remain in the cluster. As the RDAs have very little control over the decision of what firms settle in the region, a proactive, bottom-up approach to region “gardening” is proposed. Finally, the notion of an optimal cluster design is dismissed as impossible, due to the intractable nature of evolving  $\lambda$ -systems.

**Main conclusions** Current bio-based technological options, as identified by the stakeholders, do not appear to lead to a diverse bio-materials based cluster in Groningen, even under very low economic selection conditions. An enrichment of the existing cluster with bio-energy options is possible. This outcome is dependent on the assumption of the survival of the energy-intensive incumbent industry in the region. The importance of path dependency in cluster development is demonstrated, as is the very limited power that the RDAs have in controlling this evolutionary process due to the lack of control of which company appears when in the region.

**General guidelines** The following general guidelines and advice can be given regarding the RDAs’ problem of managing the evolution of a regional industrial network:

**Beware of path dependency** The order of appearance of firms matters strongly. The RDA must be very careful to build up a portfolio of firms that can incrementally feed off each other. The RDA must be aware of the fact that it is “gardening” the region, not

running a campsite. Planting many different flowers at the right time will attract many beautiful butterflies.

**Have faith in Chaotic Attractors** The once established network structure is relatively robust. Once a solid seed of a cluster is started, the chaotic nature of the evolutionary process will tend to keep stable by amplifying the initial success. On the other hand, the RDA must be very aware of large-scale external world changes that can disrupt that stable situation and cause system collapse.

**Context Dependency is king** Equally important to the firms in a cluster is its environment. The social, legal, institutional, regulatory environment can make or break a cluster, even if the right firm and technology mix is present. Again, the RDA must garden the cluster's environment as much as making sure the right mix of plants are in the garden.

**Always retain ability to control** Given the importance of past decisions and the chaotic nature of the evolutionary process, it is inevitable that mistakes will be made. Incompatible forms will be attracted. It is therefore very important to retain control of the land that is given out. Selling land is a one-way street to losing control of the cluster.

**Evolution is both Top-Down and Bottom-Up** In order to evolve a successful cluster, balance is needed between having not too much top-down control and sufficient bottom-up diversity.

**Ghost of optimality** An important final conclusion is that the inherent need to managers and designers to create the 'perfect' or 'optimal' situation is dangerous. In a complex, multi-formal evolving  $\lambda$ -system the 'best / optimal' situation cannot be objectively determined. It is a context-dependent societal decision that evolves over time. One must focus on evolving a *good enough* condition of a  $\lambda$ -system. This means allowing the system the freedom to adapt to the given situation without striving to overcontrol it. Overcontrolled or overdetermined systems become very unpredictable, as they 'seek' to relieve the constraint, often resulting in very unexpected and mostly unwanted side effects. One must be freed from the 'ghost of optimality' and stop desiring the perfect. 'Good enough' is just that. Good enough.

## 7.8 Method Development Conclusions

### 7.8.1 General

**Main advantage** The main advantage of the presented approach are that it provides a sense of the development potential of the  $\lambda$ -system relatively easily and quickly. It requires little data, all of which is publicly available. Furthermore, the developed ABM has its conceptual layout based in a social process, the SDM. This means that the generated insights will less likely be perceived as 'wisdom descending from the academic ivory tower'. Through structured interaction with the stakeholders, a shared understanding of the outcomes and their limitation is achieved.

**Main limitation** The main limitation is that it is very difficult to achieve a quantitative prediction. Since the approach is not based on a 'first principles' model, it is extremely sensitive to the quality of the data. It follows the so-called GIGO (Garbage In - Garbage Out) principle. High quality, detailed data are mostly non-existent, and when present tend to be proprietary. Verification and validation present additional problems. These will be presented below in Sections 7.8.3 and 7.8.4.

## 7.8.2 Ontology Development

**Correctness vs ease** A major effort was dedicated to ontology creation. One important aspect is the balancing of correctness and implementation ease. A more correct ontology corresponds more to the language used. On the other hand, language is often very ambiguous and makes implementation rather cumbersome. For example, a correct way to represent the concept of Physical Property is to define it as having a Value and a Unit. When implementing the Value, one assumes that it will be a number. This makes implementation straightforward. On the other hand, there are number of properties that one cannot express numerically, such as a Physical State. Implementation exceptions must be made to accommodate the possibility to classify a Physical State as Gas, Liquid or Solid. This greatly complicates implementation. In this particular case, the modellers chose for a more complex implementation, due to the obvious advantage of being able to describe the Physical State of something.

**Overspecifications** Another pitfall during Conceptualization step is the danger of overspecifying the ontology. On the one hand, one wishes to have a very complete and very specific ontology, since this allows computers to reason about it to a great extent. However, the more specific the ontology becomes, the more narrow its expressiveness becomes. In a very specific ontology, many extra concepts must be added in order to accommodate concepts that do not quite fall within the definition of the already present one. An example of this is the Design Property called Building Time. Right now, we assume that a Technology, or any other Node, can have a Physical Property called Building Time that denotes the time from the initial decision to create that Technology to the moment it is operational. During discussions with the stakeholders, the wish was expressed to add the concept of Planning Time as an extra Design Property. This would make expressiveness more complete and formal, but at the same time overspecifies the ontology and makes data collection more cumbersome.

## 7.8.3 Verification

Verification is an essential part of modelling. The verification process should demonstrate that the computer model indeed does what the modellers wanted. In other words, since a computer does exactly what one tells it to do, verification is about checking whether we told it to do what we wanted. There are two types of verification: code based and knowledge based. Code based verification consists of Compile, Run and Debug cycles and of Unit Testing. Knowledge based verification consists of Business as Usual scenarios and Extreme Parameter Setups.

**Compile, Run and Debug** The most obvious verification of the program's correctness is the compiler itself. If we made syntax and programming errors, such as using non-existing variable, the compiler will not be able to create an executable file. Some errors cannot be

caught by the compiler and will be detected during run time as program crashes. This process of coding, compiling, running and debugging is the traditional approach to software development. The repetitive process can somewhat be alleviated by using debuggers, which most IDEs are equipped with. Advanced debuggers, such as the one incorporated with Eclipse IDE, allow data and code modifications while the program is running.

**Unit testing** The next step in verification of the code is unit testing. A unit test is a formal 'contract' that the piece of computer code must satisfy. In computer programming, unit testing is a procedure used to validate that individual units of source code are working properly (Board, 1999; Wikipedia, 2007). A unit is the smallest testable part of an application. In the case of object-oriented programming, that is a class and its methods. Each component (class) of the computer program ideally has an associated unit test that specifies the desired outcome of that code, given specific parameters. For unit testing, Java JUnit <sup>19</sup> is the best known Open Source framework available.

**Baseline run** In a Baseline run, we simply run the model with parameters deemed to be most likely and common. This establishes the normal mode of operation for the model. Each outcome of such a run should be a logical consequence of agent interactions. Basically, we should be able to explain each outcome. In other words, this serves as a sanity check, to identify any behavioral artifacts caused by logic errors.

**Extreme parameter settings** To further validate the programmed logic, one can perform extreme parameter setups. Model behavior are extreme parameters and should be predictable from agent and environment logic. For example, setting prices at infinity or 0 should have predictable consequences. By systematically examining the edges of the parameter space, one can be confident that the model is consistent.

## 7.8.4 Validation

Contrary to verification, validation is very problematic. This is especially true when one models socio-technical systems. Problems occur because of:

- Inability to perform real world experiments;
- Limited historical data records; and/or
- Confidentiality or lack of current data.

**Real world experiments** In large-scale socio-technical systems, real world experimentation is impossible. One cannot say, "let's disturb the global air transport network just to see if new policy works out". This fundamentally limits model outcome validation.

---

<sup>19</sup><http://www.junit.org/>

**Historic data** Historic replays are often suggested as a way to validate the models. If a model can replay a part of the history of the system, the assumption is that it is valid and can possibly be used in prediction.<sup>20</sup> The main problem with historic replays is that data on decisions that agents made is very incomplete. History prefers winners, and the 'stupid' decisions are not recorded, even though they may be key in modelling the system's evolution.

**Current data unavailability** Even if we can have the historic data, current data can be problematic as well. When dealing with social actors, their behavioral statistics are often unavailable. Companies are mostly very confidential with performance data and the decision processes inside the company. Often, the data needed to validate the model are simply not collected.

**Future challenges** There are several important future challenges. So far, the model was used under the assumption that a current economic situation or a future development scenario is given, and the agents/clusters response to it is observed. However, in some cases it is interesting to examine the opposite. Given a desired cluster, what would the economic situation need to look like in order to make it viable? This gives a sense of how realistic the desired cluster is. The second challenge is to directly aid the RDA in cluster evolution. If we assume a rational RDA, what would be the best strategy to evolve a viable cluster? This requires that a model of the RDA be created, so that the stakeholder can reflect her own behavior in the model. Finally, an agent's decision-making needs a longer-term perspective than just direct survival. Currently, the agent's decision-making is at the operational level. It is important to add longer-term, tactical and strategic decision-making processes. Soft issues arise here, trust, regulatory stability (Vries, 2004), long-term strategic goals of companies. These challenges will be tackled in the next chapter, in the next evolutionary step.

### 7.8.5 Requirements Checked

Table 7.1 presents an overview of the modelling requirements and the performance of the model. It is clear that all the conditions are met within this case study.

---

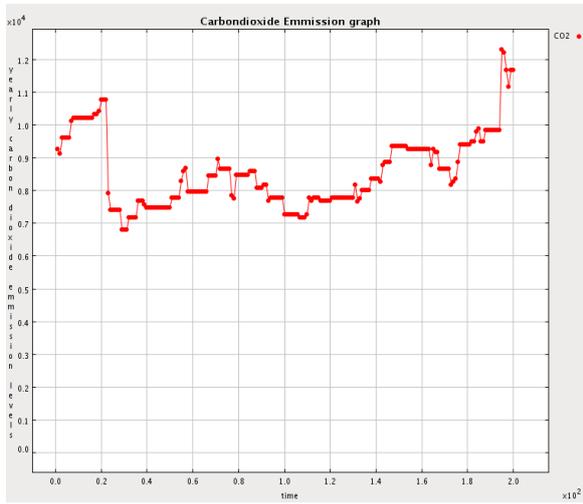
<sup>20</sup>From a more traditional perspective this is already inadvisable. Never extrapolate outside the measured points.

Table 7.1: Performance of the case study

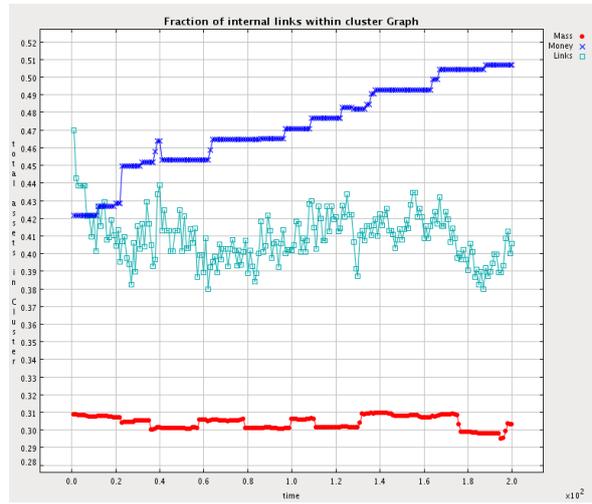
Requirement	Score	Explanation
Method		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	Yes	A wide range of domain experts and stakeholders were involved. While this is good a even greater diversity would contributed to better insights.
Organically growing	Yes	The SDM is based on the previously developed approach. The simulation engine was designed with insights obtained from the learning cases but did not include any computer code. It will serve as a basis for future organic growth.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	The Simulation engine is designed to be modular, and Simulation Generic are extensively used to encourage reuse of computer code.
Outcome		
Useful	Yes	The model has yielded both useful case level insights as well as provided a good bases for further method development. It can be considered a resounding success.
Testable	Yes	More than any other model so far, every component can be examined and tested because of the use of versioning.

## 7.9 Conclusion

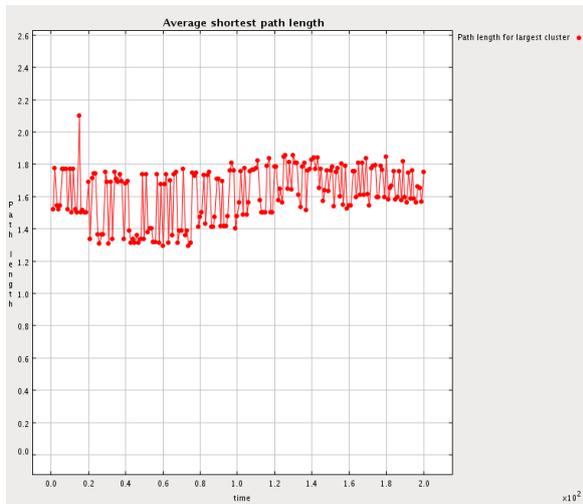
This section presented the first full-scale case study, in which all four dimensions of the co-evolutionary modelling method were extended. In the technical design dimension, a full-scale software stack design was implemented. In the social dimension, the SDM was performed, formalizing a lot of domain knowledge and encoding many facts on process industry. Such a full-scale study effectively resets the fitness landscape. In the next chapter this landscape will be deformed in different directions again, as new case studies are performed, technical elements are added to the design and new knowledge domains and facts are collected.



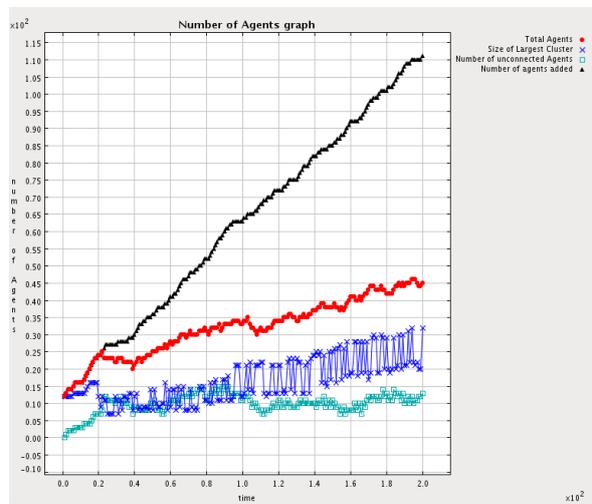
(a) Clusters CO2 production over time



(b) Fraction internal mass, contracts and money cluster. Blue: money, Red: mass, Light blue: contracts.



(c) Average shortest Path Length in the largest cluster



(d) Number of agents. Red: total agents in the simulation. Blue: size of the largest cluster. Light blue: number of unconnected agents. Black: total number of agents added

Figure 7.10: Cluster metrics of a single base case run



## CHAPTER 8

# METHOD VERIFICATION: EVOLUTION OF THREE CASE STUDIES

### 8.1 Introduction

In this chapter three case studies are presented that further evolve the model design created in Chapter 7. They are meant as a verification of the evolutionary method, demonstrating that the process of modeling aspect co-evolution can continue even after a full-scale case study has been performed and all aspects of the fitness landscape have been maximized. The direction of the model development evolution lies mainly in the technical design, knowledge formalization and fact collection dimensions.

Over the course of three case studies, we expand the types of knowledge formalized. The Bulk Biochemicals study adds Multi Criteria Analysis as a decision model for the Regional Development Agency (RDA) to guide the creation of the cluster. The Metals network case study encodes longer-term economic reasoning using Net Present Value and Internal Rate of Return into the agent's decision-making processes on whether to join a cluster or not. The case study also introduces a much improved dynamic of the World Market agent. Finally, the Bioelectricity case study adds environmental consciousness to the agents through the integration of Life Cycle Assessment into Agent-Based Modelling.

These new formalisms increase the size of the model's parameter space considerably. As a part of the technical evolution, a new parameter sweep technique using Latin Hypercube Sampling will allow us to deal with this problem.

The main outcome of the co-evolutionary method is the ability to ask and answer types of questions that we could not before. Our modelling ability and domain knowledge on  $\lambda$ -systems evolution is growing.

The precise direction of the future work is not precisely known, as evolution is a random, intractable process. The path dependency on past developments, however, hints at a number of issues that will be discussed in Chapter 11.

## 8.2 Bulk Biochemicals

**Case abstract** This case study examines the performance and evolution of a biorefinery cluster. A biorefinery is envisioned as a distributed production facility consisting of many processes, tightly integrating and acting as a single unit. As its technical configuration requires a large diversity of resources and products, sourced and sold in a diversity of markets, the economic conditions under which a biorefinery cluster can be profitable will be examined. As it consists of a number of processes, possibly owned by different firms, a rational strategy for evolving a biorefinery will be examined.

Latin hypercube sampling is examined as a technique for examining a very large parameter space of the economic environment in order to examine the cluster's fitness under different conditions. Multi Criteria Assessment (MCA) is examined as a method to rationalize the RDA's decision-making process when selecting which firms to invite to join the cluster. The main result from the economic environment exploration is that the cluster, if developed at all at one time, is likely to be successful in the majority of economic conditions examined. In the case of rational decision-making, it is found that increased rationality of the RDA through an MCA does not improve the performance of the cluster. This is mainly caused by the limited number of technological options available to the RDA in this case study. Next steps will examine a dynamic economic environment, attempt a greater diversity of agents and a more sophisticated economic reasoning by the agents.

### 8.2.1 Focus Point

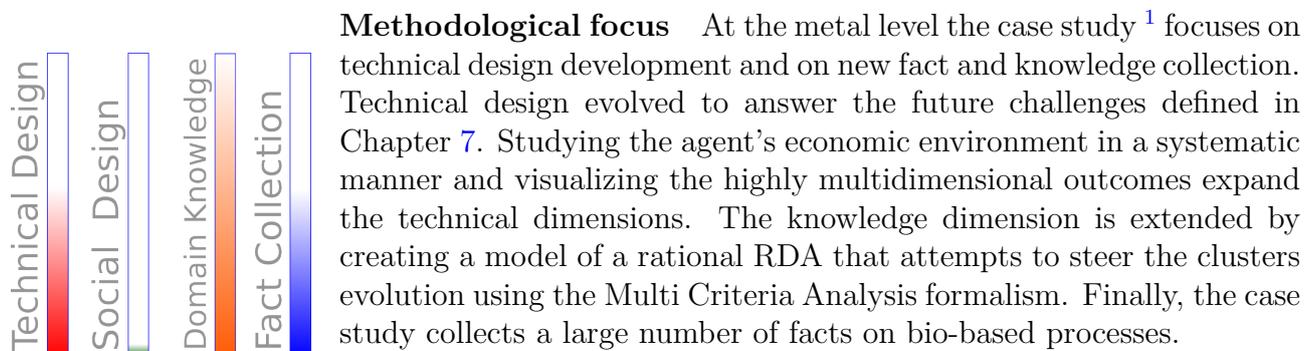


Figure 8.1: Case focus

**Case focus** At the case level, this study has two parts. The first part examines the economic environment of biomaterials based cluster, asking the question “if we want this cluster to exist and be successful, what would the economic situation need to be?” It explores the “distance to reality” of different economic situations that the cluster could be in. The second part answers the question of whether a more rational approach is possible other than the “campsite owner” method identified in Chapter 7. This part examines the suitability of Multi Criteria decision approach for RDAs in evolving a bio-based cluster. The two parts form the case study hypotheses.

<sup>1</sup>The practical case content presented in this section is performed as a part of the MSc thesis by Roos van Krevelen (van Krevelen, 2007)<sup>2</sup> supervised by the author.

## 8.2.2 Hypothesis

**Environment mapping** *It is possible to map the relevant economic environment of a cluster, finding regions where the performance is 'optimal'. The distance between the 'optimal' and current economic situations gives insight into the degree of likelihood of the creation of such a cluster. Statistical design of experiments is a viable approach for such environmental mapping.*

**Rational RDA** *The RDA can be modelled as a rational agent using factual information to determine which firm is best suited for sustainable cluster growth. Multi Criteria optimization can be used as a practical and simple tool for the task.*

## 8.2.3 Case description

### 8.2.3.1 Industrial network as a refinery

**Cluster as a biorefinery** According to the Dutch Biorefinery Network<sup>3</sup> a biorefinery consists of several processes that are clustered, making products such as chemicals, fuels, power, heat, materials, food and feed, with the highest possible added value and at lowest possible cost because they should be self-supporting with regard to heat and preferably also electricity (van Ree and Annevelink, 2007). It is important to realize that biorefineries are envisioned as clusters of many companies and not as a single economic entity.

**Competition with fossil fuels** The proposed biorefinery clusters are based on the idea that biofuel production in itself cannot compete with fossil fuel because of the feedstock costs. However, by combining biofuel production with production of valuable specialty biochemical (by)products, the overall economic performance of the cluster can improve dramatically.

**Two main questions** By definition, a biorefinery is an industrial network. There are two main questions that concern such a network. The first is whether a biorefinery cluster is a viable economic entity, given that its structure is dictated by technology. Under which economic condition will such a fixed structure perform well? The second question asks whether a rational approach by the RDA to evolving the cluster will be more successful than the random growth, "campsite owner" strategy. The proposed rational approach is Multi Criteria Analysis.

### 8.2.3.2 Mapping the Economic Environment

**ABM upside-down** ABMs are traditionally designed with a relatively fixed environment that contains a collection of adaptive agents. The goal is to observe the emergent properties of agents within that environment. In a sense, environment mapping is an upside-down use of an ABM. A fixed network of agents is taken, and the environment around them changes in order to meet some criterion. This can be done in a single simulation, letting the environment evolve and adapt itself so that it accommodates the agent network as well as possible. Such system would effectively try explore the fitness landscape and find its global maximum.

---

<sup>3</sup><http://www.biorefinery.nl/>

**Attractor map** While the global maximum is interesting, it is also rather limited. A more thorough approach would be to consider environmental variables as simulation parameter settings and sweep across a parameter space with a fixed agent configuration. The cluster is in essence a probe that maps the topology of a highly multidimensional space. This probe, while having a fixed composition, does not have a fixed structure. The internal contract and mass flow structure responds to the given environment and adapts to it. So we are examining the attractor space of the economic fitness of the cluster by taking it across the environmental parameters.

**Smallest cluster** Van Krevelen ([van Krevelen, 2007](#)) identified three distinct types of bio-refinery clusters possible in the Netherlands. The smallest cluster, consisting of 6 agents and 14 goods, will be used in this case study. Since the goal is to explore the method, the small cluster size reduces the dimensionality of the parameter space of the economic environment and increases the speed of simulation.

**Wheat straw Bio-ethanol** The cluster under examination mainly produces bio-ethanol, along with various other chemicals. It is based on the processing of lignocellulosic biomass such as wheat straw. This cluster produces many different products, starting with ethanol, but also including organic acids and solvents, methyl tetrahydrofuran (MTHF) and other aliphatic chemicals ([van Krevelen, 2007](#)).

### 8.2.3.3 Rational Gardener

The previous chapter identified “campsite owner” as an important strategy for an RDA when developing a cluster. In effect the RDA promotes the region and attempts to make it more attractive, but ultimately accepts anybody who has interest in settling in it. In effect, clusters grow randomly. As  $\lambda$ -systems are heavily path dependant, it is important to be careful about the choices one makes. In order to improve this random strategy, we have introduced a model of a rational RDA that uses a multicriteria analysis when selecting which firms to accept into the regional cluster. The expectation is that by choosing better fitting companies, a better cluster will emerge.

**Criteria** A Multi Criteria Assessment is based on the existence of criteria on which a firm can be scored and on relative weights that the RDA agent gives to each criterion. The criteria and the used weights are presented in [Table 8.1](#). Since we are concerned with the suitability of the method and not the specifics, the details will not be presented here. The exact definitions of the criteria and the rationale for choosing the weights is given by Krevelen ([van Krevelen, 2007](#)). Three types of RDA agent strategies have been identified: the economic, environmental and neutral agent. These are presented below.

**Economic Agent** The Economic Agent is an RDA that cares more than average about economic issues. In addition to firms’ economic performance, this agent values CO<sub>2</sub> emissions highly as they are an important political aspect. Firms’ technical properties are relevant, and the agent has a preference for large capacity processes and high product diversity. A positive and safe products image is produced and the policy supporting the products and raw materials

is also valued quite highly since the Economic Agent feels responsible for the wellbeing of its population. The agent has a preference is for labor intensive processes.

**Green Agent** The Green Agent RDA highly values the environmental impact and accepts lower economic performance. This agent also highly values public acceptance. Technical aspects are valued neutrally. The production efficiency, on the other hand, is highly valued as is lower energy use and low waste production. A large number of possible raw materials is highly valued since this eliminates the food competition problem of biofuels.

**Neutral Agent** This is a control RDA strategy that gives each criterion the same weight.

## 8.2.4 Details

In this section the implementation details and the experimental setup will be presented for both experiments.

### 8.2.4.1 Economic Environment Mapping

**Parameter space** The goal of this experiment is to examine the economic environment of the cluster. This means examining the cluster’s behavior across a large space of possible world market prices. The test cluster consists of 6 agents, trading in 14 goods (Water, Enzyme, Sulfuric acid, Corn steep liquor, Lime, Wheat straw, Levulinic acid, Cellulose, Hemicellulose, Lignin, Ethanol, Furfural, MTHF and Pyrolysis oil). Assuming that each price can vary by a factor of 10 up or down, we can create a 14-dimensional logarithmic parameter space that represents all the possible economic environments that the cluster could be in. Ideally, one would perform an experiment at every point in this space and observe the clusters response, effectively “mapping the environment”.

**Computational infeasibility** If we assume that we test each price at pseudo-logarithmic intervals of 0.1, 0.2,..., 0.8, 0.9, 1.0, 2.0, 3.0,..., 8.0, 9.0 and 10.0 across all 14 prices, generating a list of all possible combinations would mean making  $20^{14}$  combinations. Assuming an (unrealistically short) simulation time of 1 second, this means that simulating all these possibilities would take  $20^{14}$  seconds ( $5.2 \cdot 10^{10}$  years) which is more than 10 times longer than the estimated age of planet Earth ( $4.6 \cdot 10^9$  years). A full factorial experiment is clearly infeasible.

**Latin Hypercube Sampling** In order to examine a parameter space of this size we must revert to taking samples from it. The technique of Latin Hypercube Sampling (LHS) was initially developed to generate a distribution of plausible collections of parameter values from a multidimensional distribution (Iman et al., 1981; McKay et al., 2000; Santner et al., 2003). The sampling method is often applied in the design of experiments, were the main question is: “I can afford X experiment to test the behavior of the system. At which parameter values should I measure in order to make sure that I looked everywhere?” Given X, LHS will generate parameter sets that are guaranteed to be evenly distributed across the parameter space. While the LHS technique is very computationally expensive for large experiment numbers, it is preferable to the random sampling method alternative. See Appendix E for details.

**Experiment setup** Using Matlab, 10000 samples over 14 dimensions are created. <sup>4</sup> Taking the average current prices, the samples are taken from a range of  $[0.1 \times AveragePrice - 10 \times AveragePrice]$  for each price. The simulation scenario <sup>5</sup> has been adapted to accept a `LHSSampleNumber` parameter, which loads the sample price vectors <sup>6</sup>, sets the prices and performs the simulation. The cluster is constructed from the agent data stored in the ontology, labeled as *BiobasedBulkChemicals* and *Cluster2*. The output data is collected and processed using Matlab. <sup>7</sup> Since the cluster composition is static, the simulation rapidly stabilizes. Each simulation is therefore run for 5 time steps. This allows a larger number of experiments to be performed.

#### 8.2.4.2 Rational Gardener

**Experiment goal** The experiments attempt to examine the effect a rational, MCA based, Regional Development Agent has on the evolution of a clusters. The assumption is that different RDA rationalities will effect which firms are chosen and thus affect the development path of a cluster.

**Experimental setup** In order to give the RDA something to choose from, firms/technologies identified in all three clusters are used (van Krevelen, 2007). Each style of RDA agent is allowed to evolve 10 clusters, all starting with a different random seed. Environmental selection is set at 10 time steps. The values of cluster metrics are averaged. The total simulation time is set at 100 time steps each so that there is enough time for the cluster to develop and all technologies of the three clusters are initiated at the start-up. When the RDA agent fails to make a decision, a random agent will be added to the cluster.

**RDAs actions** It is assumed that the RDA's goal is to maximize internal use of goods produced in the cluster. After each time step, the RDA observes the goods flowing out of the cluster. Whenever a good leaves the cluster that is not labeled a Product, the RDA agent will look for a technology that can process this good into a final product. It will use the MCA to choose between all processes that could use this intermediate as a raw material. As control, a 'campsite owner' RDA strategy as defined in Chapter 7. That is, under this RDA's regime, agents are added randomly from the pool of available technologies. The weights used to define the three different RDA styles are presented in Table 8.1.

---

<sup>4</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/tags/BulkBiochemicalsCase/LHSExperimentPrices/LHSdesign.m>

<sup>5</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/tags/BulkBiochemicalsCase/LHSExperimentPrices/src/simulationScenarios/ScenarioBiobasedBulkChemicalsLHSExperimentSmallestCluster.java>

<sup>6</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/branches/Experiments/ Roos/LHSExperimentPrices/LHSPricesRoosExperiment.txt>

<sup>7</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/tags/BulkBiochemicalsCase/LHSExperimentPrices/>

Table 8.1: Weights for the criteria for various styles of RDA (scale 1-5). For definition of criteria see (van Krevelen, 2007)

Code	Criterion name	Economic Agent	Green Agent	Neutral Agent
EC1	Investment costs	5	1	1
EC2	Gross profit	5	1	1
EC3	% Profit from by-products	5	1	1
EC4	Added value of products	5	1	1
EC5	Pay-back time	5	1	1
EC6	Average impact on global market	5	1	1
EN1	Materials used	1	5	1
EN2	% Recycled materials used	1	5	1
EN3	Direct energy use	1	5	1
EN4	Indirect energy use	1	5	1
EN5	Water use	1	5	1
CO <sub>2</sub>	from production	3	5	1
TE1	Production efficiency	5	2.5	1
TE2	Total capacity	5	1	1
TE3	By-product to total product ratio	5	1	1
TE4	# of different products	5	1	1
TE5	# of possible raw materials	5	2.5	1
TE6	Process complexity	5	1	1
SO1	Labour provided	3	5	1
SO2	Awareness of product	2	5	1
SO3	Policy stimulation	2	5	1
SO4	Process risk	4	5	1
SO5	Image of products	2	5	1
SO6	Transport impact	2	5	1

## 8.2.5 Case study results

### 8.2.5.1 Environmental Mapping

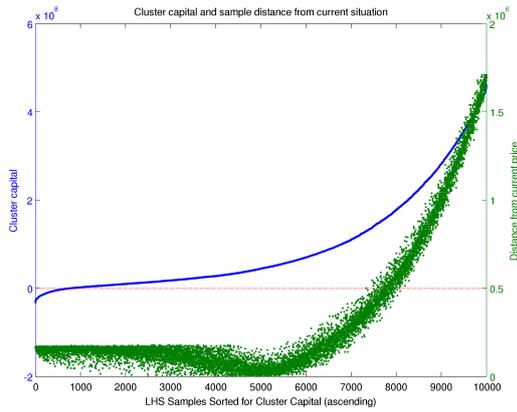
The case study data and code is available online. <sup>8</sup>

**14-dimensional space** There are 10,000 samples taken from a 14-dimensional, logarithmic space. At each sample point in this space the usual cluster metrics are measured. Since it is very difficult to visualize data from such a high-dimensional parameter space, the result plots need to be clarified.

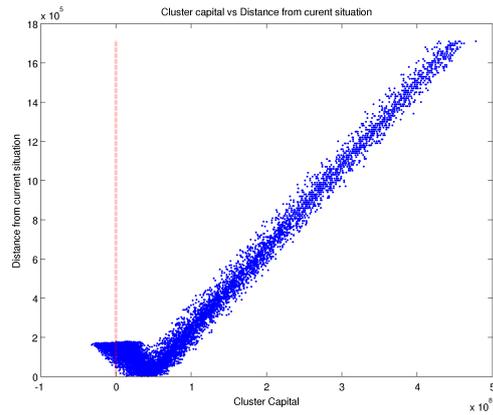
**Plot explanation** Figures 8.2 and 8.3 present two types of plots per cluster metric. Left hand figures plot the value of the cluster metric at a given LHS sample. The data are sorted in ascending order. So the blue line in Figure 8.2 (a) shows the 10,000 measured cluster capital values, sorted from the lowest value of cluster capital to the highest. The green points are the Root Mean Square (RMS) distance from the sample's price coordinate to the current price coordinate. It is the absolute value of the length of the straight line through 14-dimensional space from the current LHS sample point to the current price point. The smaller the value, the closer is the given LHS sample to the current situation. This metric can be viewed as a sample's *distance from reality*. The right hand image presents the same data, but now the cluster metric is plotted against the corresponding RMS distance.

**Cluster capital** Figure 8.2 (a) shows the sorted cluster capital and the RMS distances. The dashed red line denotes the 0 cluster capital. We observe a negative region, a long, slowly rising plateau and a steeply rising tail. Roughly half of the samples are in the mildly positive region,

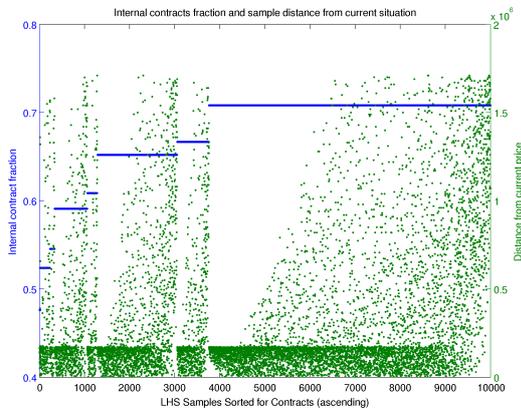
<sup>8</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/tags/BulkBiochemicalsCase>



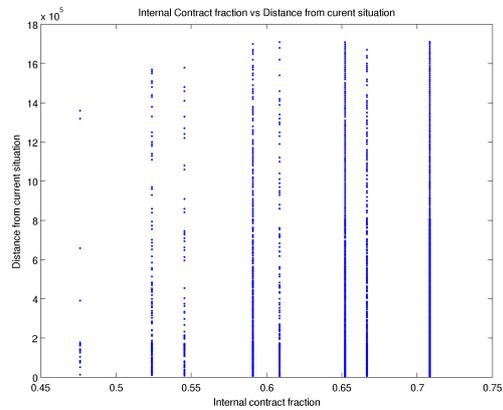
(a) Distance to average price and Cluster capital



(b) Distance vs Cluster Capital



(c) Distance to average price and Internal contract fraction



(d) Distance vs Internal contract fraction

Figure 8.2: Distance from current situation plots for cluster capital and fraction internal contracts

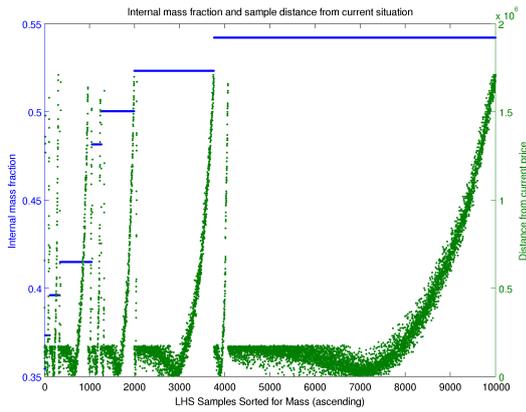
and a very small fraction is below 0. This tells us that most of the price space is favorable for the cluster's existence. The corresponding RMS distances show a region between samples 4,000 and 6,000 where the distance from reality is minimal. This shows that the cluster is likely to be profitable under the current economic conditions and nearby variations of it. It is interesting to note that as the cluster capital increases, so does the distance from reality. The maximal cluster capital values also have the maximal distance from reality. This behavior is confirmed by Figure 8.2 (b) where we see that only a small part of the samples leads to negative cluster capital values, and that there is a lot of opportunity for positive cluster capital close to the current situation. The extreme increase in cluster capital and distance from reality can be explained by variation in the price of a single product, MTHF, whose price per ton is several orders of magnitude higher than other goods (see Table 8.2). A sample that picks a very high (10 times average) MTHF price is both very successful and in absolute terms very far from the real situation. A 10-fold increase in water price has a much smaller effect on the RMS distance than a 10-fold increase in MTHF price.

**Fraction of internal contracts** Figure 8.2 (c) reveals the existence of 8 attractors in the cluster structure. We see that many diverse price settings lead to only 8 discrete contract fractions. This is especially visible in Figure 8.2 (d). This has two causes. First, the physical nature of the technology limits who can trade with whom, meaning that this metric can not assume a continuous value. There are only so many contracts that can be made inside the cluster. Second, buying from the world market always commands a higher price of transport costs. The two extreme fractions are easily explained. The high internal fraction (above 0.7) represents the situation where the cluster makes maximal use of all goods that are internally produced, and nothing that does not need to be imported is imported. Because of the capacity differences and the unavailability of producers inside the cluster for some goods, there is always some export and import activity. The other extreme is the situation where the world market prices are very low, causing most products produced internally to be sold externally. Since the transport premium and the cost structure of agents' economic model, there are products that can be produced cheaper internally, cheaper than the world market at its lowest point in the parameter space. This causes the internal fraction to be very low, but not zero. It is likely that an even broader parameter space will produce a zero fraction.

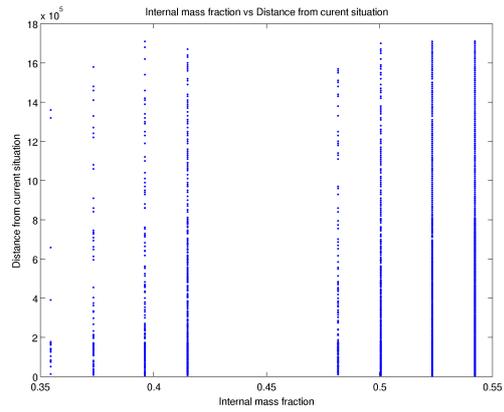
**Fraction of internal mass** The mass metric, Figures 8.3 (a) and (b), show a similar pattern to the contract metrics, since all contracts are tied to mass flows. The location of the attractors is different, since there is no fixed contract-mass ratio, i.e., a single contract can be for many tons, or just for a few kilograms. We also observe that the distance increases as we approach the edge of the attractor, see 8.3 (a). This demonstrates that the certain cluster mass fraction is retained even at an economic situation far from the current one.

**Fraction of internal money** Unlike the discrete attractor structure displayed by the contract and money metrics, the money fraction is continuous across the economic environment. This is caused by the fact that prices of goods are a continuous function, giving rise to a smooth metric distribution. We observe the effect of a single, very expensive good, observed in the cluster capital metric in an exaggerated fashion in the money fraction plots (Figure 8.3 (a) and (b)). The metric value varies between 0 and 1 asymptotically. We observe an almost stepwise increase in money fraction: as the distance from reality increases, the money fraction nears 1. So the more economically unrealistic a cluster gets, the more it is internally oriented. At the same time, a large number of samples have very low distance from reality at the same high fraction. It seems that the cluster structure and the economic environment do not favor extremely outwardly oriented clusters, where the majority of the money flow is toward the outside of the cluster.

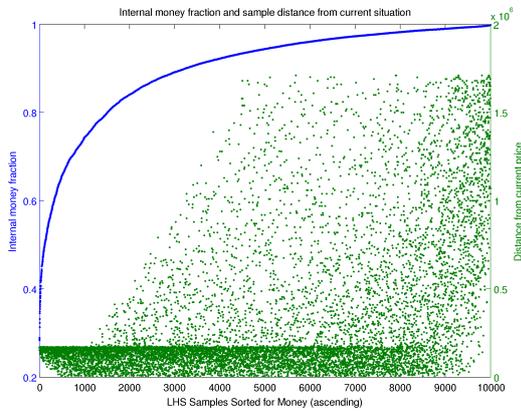
**Minimum and maximum cluster capital** As a final verification of the outcomes, the price coordinate of samples with the highest and lowest cluster capitals are presented in Table 8.2. We can observe that the highest capital indeed occurs when all the products are very high and the inputs are very low, near 10 times and 0.1 times the average price, and the cluster capital is the lowest at the exact opposite side of the price space. Obviously the cluster is most successful when the most expensive product is sold at the highest possible price and the inputs are at their lowest.



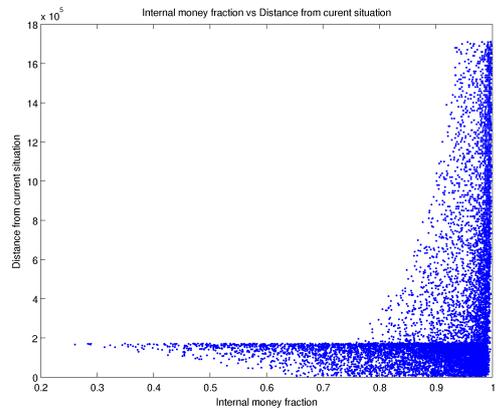
(a) Distance to average price and Internal mass fraction



(b) Distance vs Internal mass fraction



(c) Distance to average price and Internal money fraction



(d) Distance vs Internal money fraction

Figure 8.3: Distance from current situation plots for fraction internal mass and fraction internal money

### 8.2.5.2 Rational Gardener

**RDAs impact** Figure 8.4 presents the evolutionary patterns of the biorefinery cluster as a function of the RDA style. The usual metrics are presented. The values plotted are averages of 10 runs with different random seeds.

**Little difference** The main observation is that there seems to be very little difference between different RDA styles on the overall cluster capital. All RDAs perform equally poorly, considering that the cluster consisting of all identified technologies does not manage to create positive cluster capital. Examining the other metrics, the same pattern is repeated. No metric shows a clear differentiation between different styles of cluster development. We do observe large variations over time, caused by the cluster's path-dependent growth, showing that different styles do choose different firms at different times, creating a different historical development path. It is, however, impossible to say which style of decision-making has a better long-term performance, even when compared to random firm addition.

Table 8.2: Prices used and prices set for the run with maximum Cluster Capital

Good Name	Type of Good	Price at <i>max</i> capital (€)	Ratio to average real price	Price at <i>min</i> capital	Ratio to average real price
Water	Input	0.62	0.62	0.13	0.13
Enzyme	Input	13.68	0.15	346.05	3.84
Sulphuric acid	Input	262.04	4.37	323.84	5.40
Corn steep liquor	Input	127.34	0.98	21.11	0.16
Lime	Input	52.65	0.88	15.91	0.27
Wheat straw	Input	13.65	0.23	589.98	9.83
Levulinic acid	Intermediate	49,461.14	7.61	4,412.93	0.68
Cellulose	Intermediate	576.88	4.81	975.97	8.13
Hemicellulose	Intermediate	509.86	4.25	26.43	0.22
Lignin	Intermediate	92.84	1.33	20.06	0.29
Ethanol	Product	3,991.20	9.98	1,447.49	3.62
Furfural	Product	87.95	0.12	196.35	0.27
MTHF	Product	1,898,916.20	9.99	24,764.90	0.13
Pyrolysis oil	Product	1,140.19	1.19	198.41	0.21

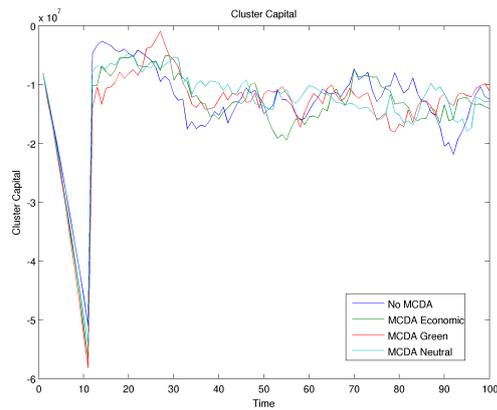
**Limited diversity** The most likely reason for the lack of difference between different RDA styles is the limited diversity of technologies that the RDA can chose from. The RDA agent attempts to find different technological options for reducing the number of intermediate products leaving the cluster. As there are very few options to do so, the agent can often not make that decision and so defaults to adding a random technology.

### 8.2.6 Domain-specific Insights

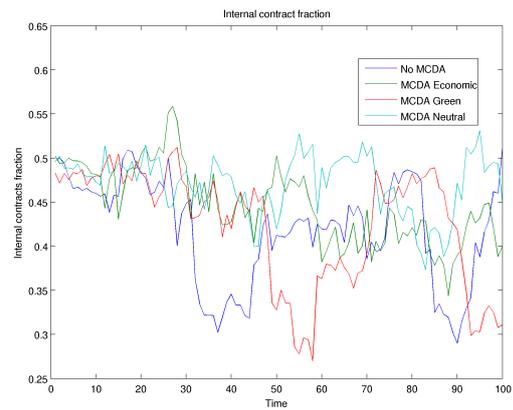
This case study mainly aimed to explore the methodological possibilities, and thus did not concern itself with domain specific aspects. Yet it is still possible to draw a few domain-specific insights.

**Environmental mapping** The main insight from the environmental mapping experiment is that the examined cluster is likely to be economically successful. At the current prices, and in the immediate surrounding in the price space, the cluster makes a net profit. Whether this positive capital is enough to recuperate investment costs and provide an acceptable rate of return is impossible to say, since the agent’s economic model is too simple.

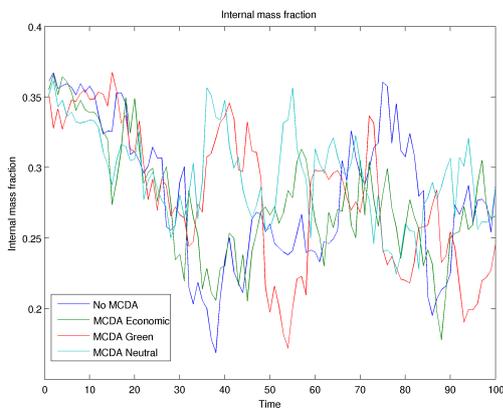
**Rational gardener** The experiments suggest that there is little difference between the proposed styles for evolving clusters. There are two main reason for this. First, the pool of potential technologies is small. The RDA agent often has very little or nothing to choose from. Second, by using a MCA to choose which firm and technology to add, the RDA only looks a single step forward. In reality, a RDA that is interested in evolving a successful cluster will take a much more long-term view.



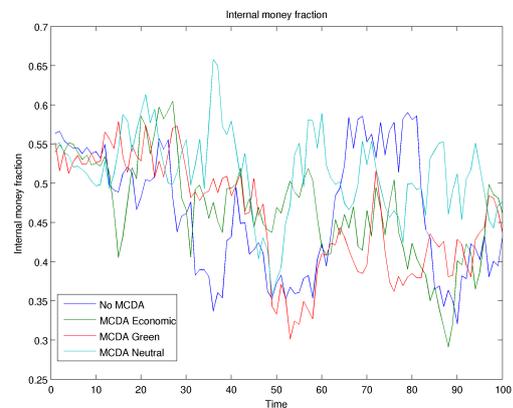
(a) Cluster capital



(b) Contract fraction



(c) Mass fraction



(d) Money fraction

Figure 8.4: Cluster metrics as a function of different RDA strategies

## 8.2.7 Method Development Conclusions

**LHS works** The main methodological conclusion is that the first hypothesis is confirmed. Not only is it possible to map the clusters economic environment, it also provides insights in the cluster's internal dynamics. LHS is a very viable method for exploring extremely large parameter spaces, even if data interpretation is difficult. The concept of *distance from reality* is a useful abstraction for understanding the cluster's behavior across the economic environment.

**MCA does work** MCA has been formalized within the ontology and implemented as an agent decision model. It is a relatively straightforward way of formalizing complex decisions, and has the advantage that it is relatively widely known and used. Furthermore, its implementation is rather simple.

**MCA only does not work** MCA does not work on RDA level unless there is a substantial number of options to choose from. Using MCA with a limited choice of technologies does not add much advantage, since the cluster's technical design space is very limited to start with.

**Too short term** Furthermore, MCA in its current implementation is too limited as the only mechanism for making cluster evolution choices, since its focus is very short term. The RDA agent using MCA only looks at the next time step. While this is better than nothing, evolution of a clusters is a long-term, path-dependant process that sometimes goes through local minima. Short-term rational optimization during such intractable processes is very difficult, if not impossible. It is important to add that this conclusion only holds true for the current implementation of MCA inside the agents. If MCA criteria would include future expectations and predictions, it might be useful.

**Future questions** Given the results and insights, several questions for further investigation can be formulated. A better agent investment economics model is needed. Currently, agents cannot decide to join a cluster, they are just added. There is no consideration of the future prospects. Return On Investment (ROI) and Net Present Value(NPV) models would improve this. Furthermore, the economic landscape, mainly the world market, is static. A more dynamic landscape would be much more realistic in evolving robust and adaptive clusters. Finally, besides NPV and ROI, agents are currently oblivious to any and all risks. Additional realism may be achieved with introduction of some sort of risk calculations.

**Requirements checked** Table 8.3 presents an overview of the modelling requirements and the performance of the model.

Table 8.3: Performance of the case study

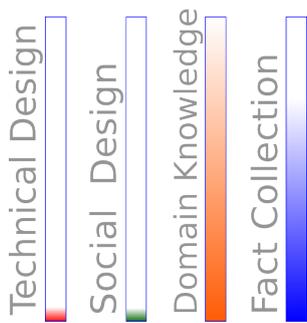
Requirement	Score	Explanation
Method		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	No	The case study is performed as a technological development step and no stakeholder was involved.
Organically growing	Yes	The case study is entirely based on the previous one.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful	Yes	Useful as a proof of principle for both LHS environmental mapping and in implementing MCA.
Testable	Yes	All outcomes are fully testable.

## 8.3 Metals Production Network

**Case abstract** This case study examines the evolution of a global aluminum and copper production network. It attempts to examine the effect that the agent's investment policies will have on the emergent network structure and its properties. The case extends the existing case studies by formalizing knowledge in the metals processing domain. At the modelling method level, the case extends the agent's reasoning with Net Present Value and Internal Rate of Return calculations, and adds a dynamic world market with global interest rate developments.

The main outcome is that the agent's investment policies have very little effect on the development of the cluster, as the economic reasoning and limited diversity of agents severely constrain the network's ability to evolve. Future work will include the refinement of the economic reasoning and a refinement in the environmental performance metrics.

### 8.3.1 Focus point



**Methodological focus** The case study<sup>9 10</sup>, will focus on fact and knowledge collection. It addresses the future questions defined in the previous case study. In terms of knowledge, the agent's economic reasoning and the world market dynamics will be made more realistic. Agents' strategic decision-making is expanded, allowing them to decide whether or not to join a cluster. The agents also acquire the ability to temporarily stop production when economic conditions are unfavorable. The interest rate and dynamic price developments are modelled, making the World Market agent more realistic. Dynamics is modelled using the System Dynamics modelling formalism, creating the initial steps towards a hybrid modelling approach, combining

System Dynamics and ABM aspects. The fact aspect of the case study focuses on collecting metallurgical process descriptions.

**Case focus** The model describes the evolution of a global network of copper and aluminum production over a 50-year period under different global development scenarios. The focal point of the model is the evolution of production network under different agent decision styles, different global economic conditions and different metals use cases. The main evaluation criterion for the evolved production networks is their environmental profile, mainly energy use, virgin materials use and total generated waste.

### 8.3.2 Hypothesis

**MCA with future expectations** Multi Criteria Assessment, combined with estimates of future economic performance of technologies is a useful mechanism for agents to use when deciding whether to join the cluster.

---

<sup>9</sup> The practical case content described in this section is performed as a part of an MSc thesis by Krittaya Sakamornsnguan (Sakamornsnguan, 2007), supervised by the author

<sup>10</sup> <http://wiki.tudelft.nl/Research/NuksNotes>

### 8.3.3 Case Description

**Metals are important** Metals are one of the basic types of materials that human civilization is built upon. From the first bronze axe used to fell a deer to the rare-earth superconducting magnets used in a particle accelerator, metals are an essential part of human society. Their production network, being a  $\lambda$ -system, warrants a closer examination.

**Metals networks** Metals networks generally consist of 4 stages, including extraction, production, manufacture, and recovery (Verhoef et al., 2004). During the extraction stage, virgin resources such as metal-bearing ores are mined from the environment. These ores are refined, producing pure metals and metal alloys. Materials produced in this way are used in various applications and are often recovered at the end of the life cycle.

**Interesting characteristics** The metals production network has several interesting characteristics. As with any  $\lambda$ -system, the metals production network consists of a many social and technical components. These components are very widely distributed geographically. The metals production network is very resource-intensive and has a large environmental impact. Many metals are functional substitutes for each other, making the already complex production network even more interdependent. These aspects are discussed in greater detail below.

**Global network** The metals production is a global business. Ores are extracted all over the globe, refining and manufacture often taking place at very different locations. Because of the size of investments necessary, companies involved are very large multinationals, operating across many economic and political environments.

**Resource intensive** Metals production is very resource intensive. Mining often happens in extremely large open pit mines, covering tens of square kilometers, involving shifting enormous amount of overburden and large amounts of toxic chemicals. Refining involves large amounts of water and energy.

**Co-production** Most metals are not produced in single product processes, but are co-produced with other metals. This means that any change in production volume of a certain metal will affect other, co produced metals, so demand decrease for a carrier metal such as copper has a direct effect on the availability of the co-produced metals such as gold.

**Functional substitution** Metals often have functional substitutes. Electricity, for example, can be transmitted through copper or aluminum cables. These production and use linkages further increase the complexity of the metals network.

**Copper and aluminum** In the case study the production networks of copper and aluminum will be examined. These two metals are each other's functional substitutes in a number of applications, mainly electrical, and both have large resource costs involved in their production. Both metals are recycled to a high degree, mainly due to their cost. In order to keep the case study manageable, no other metals will be considered.

**Model setup** The model setup is very similar to previous models. Starting agents are added to the simulation, where they must survive by producing products and by trading with each other and the world market. The main difference is that there is a pool of alive but inactive agents that observe the world market conditions and the cluster. When sufficient demand is present, the inactive agents can decide to invest with a suitable technology. When the conditions worsen, they can decide to stop production. All other behavior is identical to the previous models. The agents' extended economic decision-making power and the world market dynamics are discussed in greater detail in the following section.

### 8.3.4 Details

**Main assumption** The main simplifying assumption used in the model is to include only the extraction and production stages of copper and aluminum life cycles. The remaining production stages (i.e., manufacture & consumption as well as disposal & recovery), international and national institutions, and the dynamics of the physical environment are not taken into consideration. The model detail will be discussed in two sections: the investment decisions and world market dynamics.

#### 8.3.4.1 Investment Decisions

**Sufficient demand** The agents that want to invest search for potential products by calculating the ratio between total supply (S) and the total demand (D) of each reference product of every existing technology. The potential demand of products is defined by the S/D value of less than 0.7. In the case that multiple technologies are possible, an agent performs an MCA analysis to decide which technology to choose.

**MCA** The technologies that are deemed profitable because of a good S/D ratio are compared using six criteria: net present value (NPV), internal rate of return (IRR), use of secondary material, generated wastes, generated emissions and energy use. The technology with the highest score is selected to be invested in by the new agent. A simplified investment decision algorithm is presented in Algorithm 8.1.

**Underlying assumptions** There are three main assumptions affecting the agents' investment decision-making. These are:

**Constant scale** First, the scale of each technology is constant and set at its maximum. The operational scale is not adjusted during operation.

**No lead time** Second, the model does not consider the long lead time associated with building the production plants. As a result, agents are not able to delay or cancel an investment once a decision has been made.

**Capital availability** Third, capital funding is assumed to be readily available. Agents do not experience any financial constraint when deciding to invest.

---

**Algorithm 8.1** Simplified investment decision algorithm

---

```
for all technology do
  for all reference product do
    if S/D ratio < 0.7 then
      add good name into needed product list
    end if
  end for
end for
for all needed products in needed product list do
  Calculate: S/D ratio
  if S/D ratio < 0.7 then
    for all technology do
      if reference product = needed product then
        put technology name to alternative list
        Calculate: NPV of the technology
        if NPV > 0, using interest rate set in the world market agent then
          put technology name to NPV-screened alternative list
        end if
        use MCA with the NPV-screened alternative list {choose the best score using the
        weight factor in scenario}
        calculate: S = S - operational scale
      end if
    end for
  end if
end if
end for
```

---

**NPV and IRR calculation** As presented above, an agent needs to calculate the Net Present Value (NPV) and Internal Rate of Return (IRR) of the technology it operates in order to make an investment decision. NPV and IRR are calculated using the technology's lifetime, expected cost and expected revenue of a certain technology, as well as interest rate, which is changed by the world economic growth rate. The expected cost of a technology includes construction and operational costs. Algorithm 8.2 presents the NPV calculation and algorithm 8.3 the IRR calculation.

---

**Algorithm 8.2** Simplified Net Present Value calculation

---

**for** all year within the life time **do**  
    **Calculate:** expected metal payment  
    **Calculate:** expected revenue  
    **Calculate:** expected profit<sub>year n</sub> = expected revenue - operational cost  
    **Calculate:** discounted expected profit =  $\frac{\text{expected profit}_{\text{year } n}}{(1+\text{interest rate}_{\text{year } n})^n}$   
    **Calculate:**  $NPV = \sum \text{discounted expected profit} - \text{ConstructionCost}$   
**end for**

---



---

**Algorithm 8.3** Simplified Internal Rate of Return calculation

---

**for** interest rate = 0 to interest rate making NPV > 0 **do**  
    **Calculate:** NPV  
    **if** NPV > 0 **then**  
        IRR = interest rate {IRR is the first value that makes NPV > 0}  
    **end if**  
**end for**

---

**Suspended animation** In previous models the agents had no choice but to go on losing money with unprofitable technology until they went bankrupt. In this model the firms can decide to suspend their operation in order to prevent losing too much money when the economic conditions are unfavorable. They are also able to restart operations when the conditions improve. Typically, a firm starts the production of an good when profitable and suspends production when profits are low. Values used in the algorithm are defined as: Current profit = margin - operational cost; Accumulated profit =  $\sum$  (current profit) - construction cost; and Expected profit = expected revenue - operational cost. The simplified logic of suspending and resuming production is presented in Algorithms 8.4 and 8.5.

---

**Algorithm 8.4** Production suspension algorithm

---

isTemporarilyStopped = **false**  
**if** accumulated profit > 0 AND current profit < 0 AND expected profit < 0 **then**  
    isTemporarilyStopped = **true**  
**end if**

---

When an agents production is suspended, no contract will be made and the operational cost is assumed to be only 10% of general operational cost.

---

**Algorithm 8.5** Production resumption algorithm

---

```
if expected profit > 0 AND isTemporarilyStopped = true then  
  isTemporarilyStopped = false  
end if
```

---

### 8.3.4.2 World Market Scenarios

In previous models, the world market was no more than a static infinite source and sink for goods. In this case study, the world market has evolved into a dynamic actor with its own logic. It is a hybrid entity, both being an agent and having system dynamics based behavior. Its behavioral components are presented below.

**Raw materials** The model does not describe ore extraction from the environment. The mining process is too diverse and complex to be included without stretching the model scope too far. The world market agent supplies ores to the metals production agents.

**Metals recycling** The waste management firms which collect and recover metal-bearing products are abstracted as metal scrap goods. The main assumption is that the scraps are always available and can be supplied to the system when required. Furthermore, it is assumed that the quality of these scraps is uniform.

**Consumers** The world market agent represents an aggregate of the global metal consumption as well as performs the role of exchange firms such as the London Metal Exchange (LME). The world market sets the demand and market price of goods in the simulation. The demand function (D) of metals is a function of time, economic growth and resource efficiency. Equations 8.4 and 8.5 describe the behavior.

**Global economic growth** The growth of metal demands strongly correlates with world economic growth. It is assumed that the demand of every good grows at the same rate and the rate is proportional to the global GDP growth.

**Interest rate** The interest rate used in both calculations is defined as a function of both time and the global economic growth rate ( $G_e$ ), since economic growth has an influence on the fluidity of the monetary system. That is, when economic growth is high, the interest rate will be raised to avoid inflation; and when the growth is low, the interest will be lowered to allow more circulation in the system. Interest rate calculation is presented in Algorithm 8.6.

**Efficiency factor** The efficiency factor represents the progress towards less resource consumption per unit of product. The increase in efficiency is driven by technological and design developments and by legislation. These aspects are considered to be external to the model. The efficiency factor is assumed to be constant within the demand function, and its value drives a reduction in metal demand. For example, an efficiency factor equal to 3% means the metal demand decrease by 3% per year.

---

**Algorithm 8.6** Simplified interest rate calculation

---

```
 $i = f(t, G_e)$   
if  $G_e \leq 2$  then  
   $i = 12$  (%)  
end if  
if  $G_e > 10$  then  
   $i = 24 + G_e/2$   
else  
   $i = 12 + 2(G_e - 2)$   
end if
```

---

**Substitution rate** For the case of copper and aluminum, substitution is an important factor to their similar functional properties. The substitution factor applied to the demand function of copper and aluminum represents net replacement between both metals over time, and it is assumed that the substitution between copper (or aluminum) and other materials (such as paper and plastic) is negligible. The factor is set in the form of a sinusoidal graph. The positive value represents the substitution of aluminum over copper while the negative one is the reverse of this. One period of substitution is assumed to cover 15 years. The maximum and minimum values are proportional, based on total amounts of copper and aluminum. The demand functions of copper and aluminum ( $D_{Cu}$  and  $D_{Al}$ ) are shown below:

$$D_{Cu} \text{ and } D_{Al} = f(t, G_m(G_e), S, E) \quad (8.1)$$

$$G_e = a * G_m \quad (8.2)$$

$$a = \text{constant} \quad (8.3)$$

$$D_{Cu, \text{yeart}} = D_{Cu, \text{yeart-1}} * (1 - 0.01E_t) * (1 - 0.01S_t) * (1 + (0.01G_{m,t})) \quad (8.4)$$

$$D_{Al, \text{yeart}} = D_{Al, \text{yeart-1}} * (1 - 0.01E_t) * (1 + 0.01S_t) * (1 + (0.01G_{m,t})) \quad (8.5)$$

where :

$D_{Cu}$  = demand of copper

$D_{Al}$  = demand of aluminum

$t$  = time

$G_m$  = metal demand growth rate

$G_e$  = world economic growth rate

$S$  = substitution factor between copper and aluminum

$E$  = resource efficiency stages of the supply chains

**World market price** Metals prices are set by exchange firms and are driven by their inventories and demands (Joseph and Kundig, 1999). Therefore, the price functions of copper and aluminum ( $P_{Cu}$ ,  $P_{Al}$ ) are set to respond to overall demand and overall supply of the system by using a multiplier. The multiplier (M) is a ratio of demand and supply which increases the price when demand exceeds supply and decreases the price when supply exceeds demand. The price functions (and multipliers) of copper and aluminum are shown in Algorithm 8.7.

---

**Algorithm 8.7** World market price calculation

---

$$P_{Cu} = f(t, S_{Cu}, D_{Cu})$$
$$P_{Cu,t} = P_{Cu,t-1} * multiplier(M_{Cu})$$
$$Ratio_{Cu} = \frac{D_{Cu,cluster,t-1} + D_{Cu,worldmarket,t-1}}{S_{Cu,cluster,t-1}}$$
**if**  $Ratio_{Cu} \leq 0$  **then**  
     $Ratio_{Cu} = 1$ **end if**  
**if**  $Ratio_{Cu} > 1$  **then**  
     $M_{Cu} = 1 + (0.01 * Ratio_{Cu})$ **else if**  $Ratio_{Cu} \leq 1$  **then**  
     $M_{Cu} = 1 - (0.01 * Ratio_{Cu})$ **end if**

---

### 8.3.4.3 Agent Policies

**Overview of policies** In order to test the hypothesis that an MCA is a useful tool for agents in deciding when to join a cluster, four investment policies have been defined. See Table 8.4.

**Policy 1: Towards sustainability** Strong environmental regulation, high promotion of recycling and energy-saving. In this scenario, global scale environmental problems are high on the agenda. Thus, the national governments implement stricter environmental regulations, especially regarding emissions controls. Furthermore, public awareness on hazardous and contaminated wastes results in stricter control of production processes and resources and energy conservation. Reuse and recycling markets and technologies are stimulated by subsidies. As a result, the use of secondary materials is preferred by firms. The use of renewable energy is preferred, as are efficiency improvement measures. Consequently, weight factors of generated wastes and emissions have to be higher. In addition, subsidies for high energy efficiency and recycling technologies attract the interest of the investor. At the same time, these subsidies can compensate money losses. Therefore, the use of secondary materials and energy criteria are more important than the economic issues.

Table 8.4: Overview of agent investment policies and their associated MCA weights

Criteria	Weight factors			
	Policy 1	Policy 2	Policy 3	Policy 4
NPV	1	10	10	1
IRR	1	10	10	1
Use of secondary materials	10	1	10	5
Generated emissions	10	1	10	1
Generated wastes	10	1	10	1
Energy use	10	1	10	10

**Policy 2: Laissez faire** Weak environmental regulation, low promotion of recycling and energy savings. In this case, environmental problems appear periodically, but there is no leader group to drive the policies and regulations towards a certain issue. The firms have to push themselves towards less energy consumption, more use of secondary materials or fewer emissions and wastes without any support. As a result, the main parameter in deciding if they should invest in any technologies is the potential profit from the investment. Since the environmental regulations are weak and there is no promotion regarding resource and energy consumption, the investors pursue their main goal: maximizing profit. As a consequence, they concentrate only on economic factors and pay less attention on the environmental criteria.

**Policy 3: Cleaner is better** Strong environmental regulation, low promotion of recycling and energy savings. In this scenario, the effects of global environmental problems, like global warming, become more apparent over time, raising social awareness on emissions issues. As a result, stricter environmental regulations are put in place to control the emissions. Waste management becomes a topic of concern, tightening waste regulation. Subsidies are provided to the cleaner technologies and treatment system. Unlike environmental regulations, the promotion of recycling and energy saving is low. Since there is no subsidy, the investor has to be more concerned regarding her investment and profit. At the same time, she is automatically forced to become cleaner by increasing social awareness and stricter regulations.

**Policy 4: Towards less extraction** Weak environmental regulation, high promotion of recycling and energy savings. In this scenario, the rapid economic growth of past years has driven metal supplies to the very limit of production capacity, resulting in high concerns about future resource supplies. Therefore, development is directed towards the reduction of virgin material extractions. This becomes a good opportunity for waste and by-product materials, and governmental and private sectors try to promote this due to environmental and economic benefits. Besides the resources, energy is another significant issue, since its consumption relates directly to resource production. The drastic increase of resource demand results in substantial energy consumption and thus social awareness on energy sources and supplies. Subsidies are given to the energy-saving project as well as the one that reuses or recycles. However, environmental regulations are kept steady at a low standard, making the use of wastes and by-product less interesting. As a result, all the criteria are weighted equally at 1, except for the use of secondary materials and energy. However, the weight for the energy criterion is higher because the environmental regulations are weak, resulting in lower costs of waste management. Therefore, the investors are less interested in using scrap, due to its low availability.

#### 8.3.4.4 Experimental Setup

Compared to the previous case study, the main difference is the absence of an RDA. There is no central agency attempting to organize the cluster.

**Experiment** Multi Criteria analysis is used as a decision mechanism for Agents in deciding when to join the simulation. Agents can use the four different policies defined above. Policies are tested across the world market parameter space. The world market demand is reduced 100 fold from its actual value, in order to keep the number of agents and thus simulation time manageable. The economic selection pressure is set at 5 time steps.

**Parameter space** Each policy is tested against the parameter space spanned by the economic growth rate, efficiency factor and aluminum substitution factor. All three parameters are varied from 0 to 10 %. This range of values is estimated to be realistic. 100 LHS samples are taken from this 3-dimensional parameter space.

**Metrics** Unlike the previous cases where we were interested in studying the network structure, the focus is on the cluster's overall performance. The metrics used in the previous cases are relatively uninformative, as the supply chain is relatively fixed and they do not offer insights into environmental effects specific to metals processing. Therefore, the following metrics will be used in this case study:

**Cluster capital** The same metric as used in previous models.

**Total energy used** This metric tracks the total energy used. It is the combined energy in different types of liquid fuels used by the agents plus the total electricity used.

**Virgin material use** It is the total sum of bauxite, copper oxide and copper sulfide used by the cluster.

**Total waste generated** The sum of Overburden, Bauxite Residue, RedMud, Copper Tailing and Copper Gangue.

### 8.3.5 Case Study Results

The raw data and data processing scripts are available online <sup>11</sup>.

**Reading the graphs** The figure 8.6 presents an example of the results. It presents the time trajectories of 100 LHS experiments, summarized in a 'box and whiskers' plot. The box and whiskers are calculated at each time step. The blue box bounds the upper and lower quartile of the data at each given time step, with the red dash in the middle denoting the median. The whiskers span the extremes of data. The red crosses outside the whiskers represent the outliers. The graphs represent the average values and the spread of trajectories of all simulations over time.

**Cluster capital** Figure 8.6 presents the cluster capital development over time and across the parameter space, under different policies. We observe that all simulations start in a very similar fashion, with almost no spread in the trajectories up to timestep 20. From there on, the spread across parameter space increases somewhat. The main result is that the agent policies no significant effect on the cluster development. In all 4 policy cases the averages of endstates and their spread are similar.

**Total virgin material use and total waste generated** Figure 8.7(a) presents the total virgin materials used. The figure 8.7(b) shows the total generated waste. We see that these metrics demonstrate the same lack of differentiation between the policies, and that the general trend follows cluster capital.

---

<sup>11</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/tags/MetalsNetwork/resultAndGraph/>

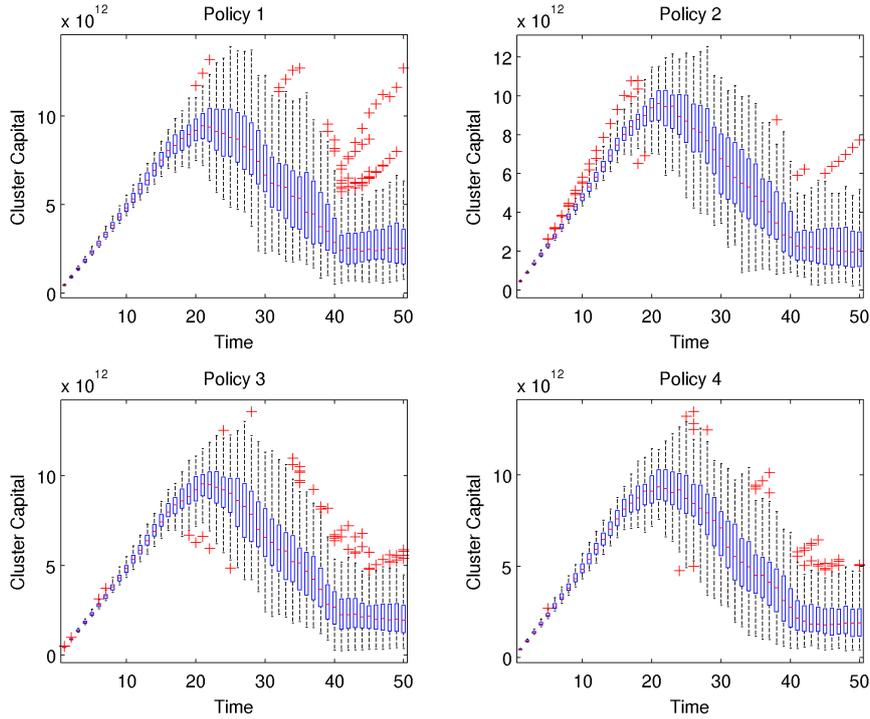
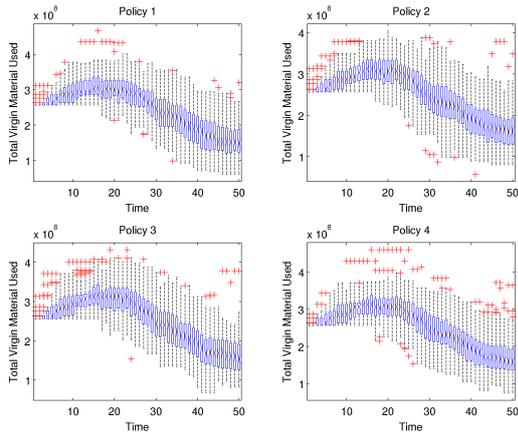


Figure 8.6: Cluster capital per Agent policy, all scenarios, time series

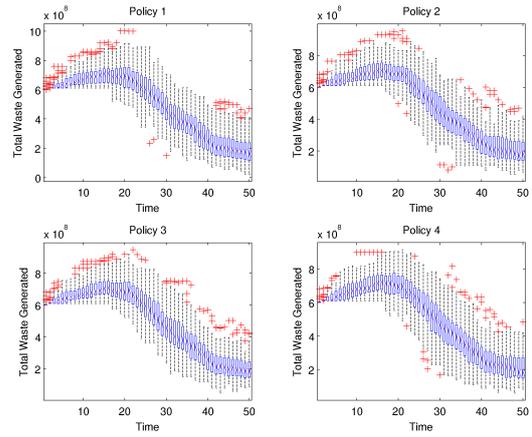
**Total energy and scrap metal used** Figure 8.8(a) plots the networks total energy used, over time, per policy, across the parameter space. Figure 8.8(b) presents the total scrap metal used by the network. Again, the main observation is that there is very little difference between different policies. The average of the scrap metal metrics has a relatively flat profile, except for a number of outliers, occurring at certain parameter settings, in which a large amount of scrap is used. This variability across the parameter space will be examined in more detail below.

**Parameter space** The agent policies show very little difference in the results. However, the spread of the results within a policy is relatively large, especially for scrap metals used. In order to examine the cause of spread, we examined the outcome space of policy 3 (equal weights for all aspects) across the parameter space, looking for patterns. Figure 8.9(a) presents the simulation end state at the 50th tick of the simulation runs, across all LHS samples. The size of the circle represents the relative size of the cluster capital. Figure 8.9(b) presents the scrap metal used, again with the circle size corresponding to the value of the metric.

As already observed, the spread of values is not very large for the cluster capital metric. Examination of the parameter space reveals no obvious patterns. There is no correlation between the samples position in the parameter space and the value of the cluster capital. The scrap metal use shows a limited number of runs that have much higher metric values than average. No clear pattern is observable in the location of the large value samples.

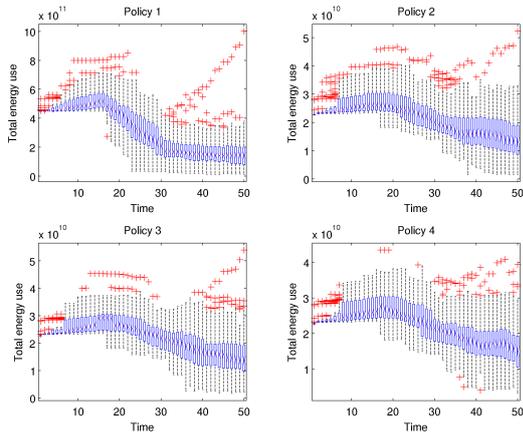


(a) Total virgin material used

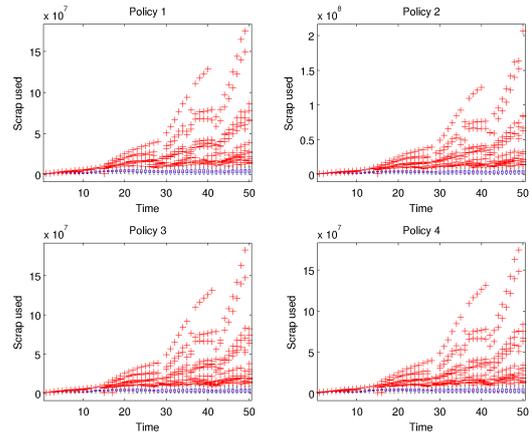


(b) Total waste generated

Figure 8.7: Waste generated and virgin materials use per Agent policy, all scenarios, time series



(a) Total Energy use



(b) Scrap metal used

Figure 8.8: Total energy and total scrap metal used, per Agent policy, all scenarios, time series

### 8.3.6 Domain-Specific Insights

The goal of this case study was to expand the modelling method with new types of knowledge and facts on metals networks. The goal was not to explore the model domain in detail. However, several case-specific insights can be presented.

**Robust system** The future trends are very robust across the tested world market scenarios. We do not observe radical changes over time, only outcome drift caused by the randomness of the model. There are two main issues constraining the evolutionary patterns of the model. First is the economic shortsightedness of the agents, and the second is the limited diversity of technologies available.

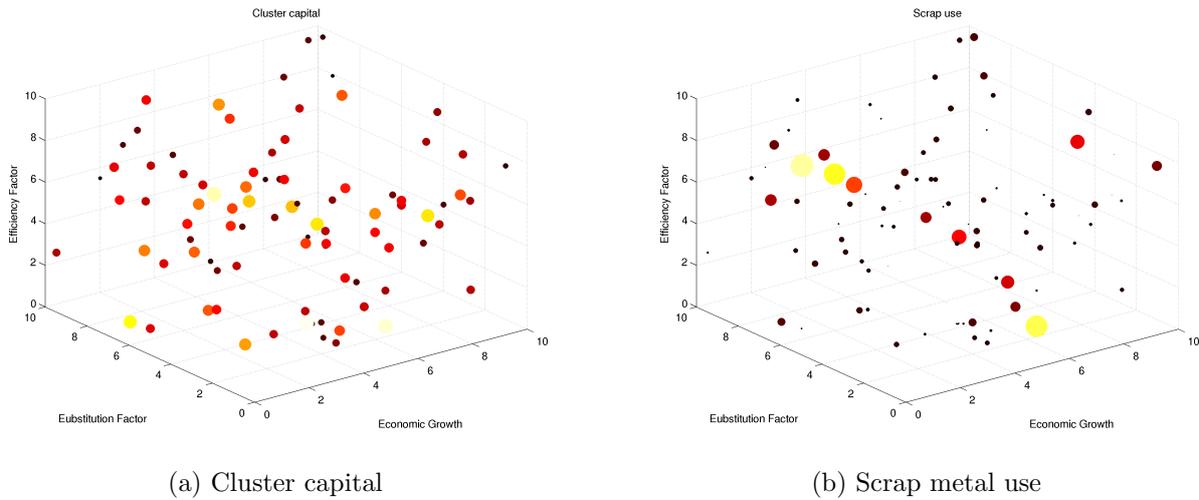


Figure 8.9: Overview of the parameter space

**Economic shortsightedness** The cluster’s capital metric demonstrates the cluster’s collapse most tellingly. After an rapid increase in the cluster’s ‘biomass’, the system starts to shrink, almost returning to its starting point. As the market prices increase over time (see Section 8.3.4.2 for a description of world dynamics), and because of the policy bias towards using recycled metal, the cluster starts evolving away from primary production. Expensive primary producers go extinct. This affects the total cluster capital negatively. The relatively strict and short-term NPV/IRR driven decision-making does not allow the agents to reinvest in expensive primary production, but instead they keep on investing in recyclers. This causes the overall collapse of the network. In order to avoid this lock in, the agents would need to have decision-making processes that are able to examine much longer time scales and are able to avoid local minima. Using NPV/IRR is already an improvement over previous models, in terms of longer time planning, but it is not enough in a world where the market prices are volatile and investments are very large.

**Limited diversity** This case highlights the same problem we faced in the previous case. The diversity of technologies to invest in is rather limited, so no major shift in network structure is possible. Such a shift would allow the cluster to adapt to new environmental conditions and survive. Given the current diversity, that is not possible, and the cluster collapses.

### 8.3.7 Method Development Conclusions

The hypothesis of this chapter was stated as: “MCA combined with estimates of future is a useful mechanism for Agents to use as a for decisions whether to join the cluster”. We can conclude that at the method level the extension is useful, as it enriches the economic reasoning of the agents in a significant manner and sets the stage for the next case study. We can furthermore conclude that the extension of the ontology with metals related domain knowledge was a success, as is the elaboration of the dynamics of the world market.

**Future questions** The issues surrounding the limited diversity and shortsightedness of the economic reasoning have been discussed in the previous section. These issues will be addressed in the future cases. Furthermore, the model also allowed us to enrich our understanding of the cluster’s environmental impact. It is, however, not finely grained in comparison to traditional environmental impact metrics such as Life Cycle Analysis (LCA), and it is limited to bulk metrics such as scrap metals use, total waste generated and energy used. In the future a more life cycle oriented approach would provide a better sense of the cluster’s impact.

**Requirements checked** Table 8.5 presents an overview of the modelling requirements and the performance of the model.

Table 8.5: Performance of the case study

Requirement	Score	Explanation
Method		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	No	The case study is performed as a technological development step and no stakeholder was involved.
Organically growing	Yes	The case study is entirely based on the previous one.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful		Useful as a proof of principle for both encoding metal processing knowledge refinement of economic reasoning.
Testable	Yes	All outcomes are fully testable.

## 8.4 Bioelectricity

### Case abstract

The final case study in this thesis adds the formalism of life cycle environmental assessment to the simulation engine. The case study examines the evolution of the Dutch bioelectricity production network under different CO<sub>2</sub> tax levels and under different agent reasoning strategies.

The incorporation of Life Cycle Assessment (LCA) enables agents to reason about their environmental impact across their supply chain. Incorporation of the EcoInvent LCA database enables the World Market to provide goods and environmental impacts from over 3,000 different production processes. The case solves a number of very complex algorithmic and computational challenges in combining a static analysis tool (LCA) with a dynamic pattern generation tool (ABM).

The main methodological outcome is a practical way to combine LCA with ABM. Furthermore, that case study shows that decision-making (pro profit vs pro environment) has very little effect on the overall emissions of the bioelectricity cluster. It also shows that the high levels of CO<sub>2</sub> taxation allow for structural change in the way bioelectricity production is organized.

The case study opens up the avenue to extending the model with other types of static metrics and tools, like economic Input-Output tables and economic Computable General Equilibria models.

### 8.4.1 Focus point

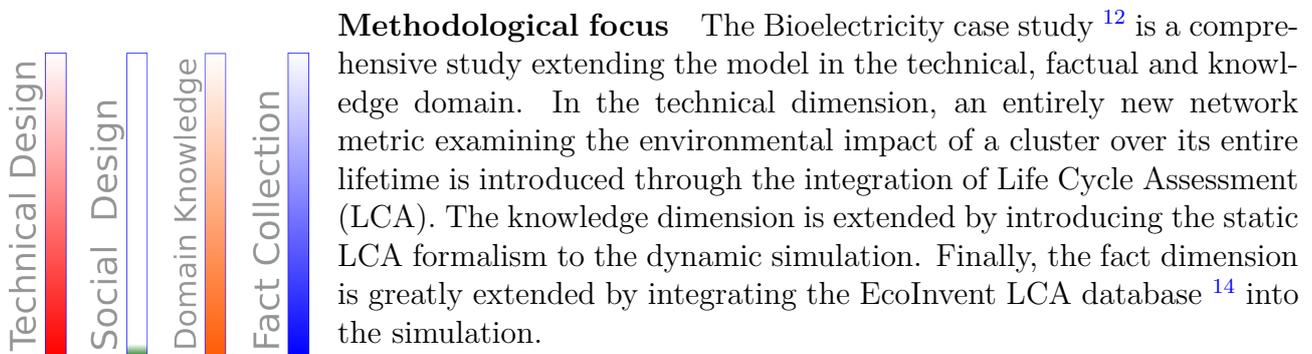


Figure 8.10: Case focus

**Case focus** The Bioelectricity case will focus on studying the changes in the Dutch electricity production from fossil fuels to bio-based fuels.

The environmental impact of the biofuel is of special interest, since not all biofuels are created equal. The impact of CO<sub>2</sub> tax on the total CO<sub>2</sub> emissions of evolved bioelectricity clusters will be examined. The case study is performed as an extension of the existing Dutch government project project (Cramer, 2006).

<sup>12</sup>This section is based on the MSc thesis by Chris Davis (Davis, 2007)<sup>13</sup> supervised by the author and on the paper “Integrating Life Cycle Analysis with Agent Based Modelling: Deciding on Bio-Electricity” by C.B. Davis, I. Nikolic and G.P.J. Dijkema (Davis et al., 2008).

<sup>14</sup>It holds environmental profiles of almost 3,000 production processes

## 8.4.2 Hypothesis

**Combine LCA and ABM** *It is possible to combine a static LCA tool with the dynamic ABM approach to create a model providing comprehensive insights into the network's environmental impact.*

**Environmentally conscious Agents** *Making agents environmentally conscious through integration of LCA will affect the evolution of a bioelectricity production network.*

## 8.4.3 Case description

**Climate change** Concerns about climate change and the security of future energy supplies have led to an increased focus on utilizing biomass as a feedstock for energy production. While many want to see biomass succeed for all the right reasons, it has become clear that not all biomass is suitable, and we have to be conscious about not creating new problems as we work to solve old ones. This has become evident in the recent controversy over the co-firing of palm oil in Dutch coal plants ([Junginger and Faaij, 2005](#)). Increased food scarcity is affecting developing countries worldwide. The destruction of Southeast Asian tropical virgin forests into palm oil plantations ([Fargione et al., 2008](#)) is just one other example of a new problem arising. Even without these complications, there are already questions about how effectively we can reduce greenhouse gas emissions when the biomass being used is shipped from the other side of the world. With interest now growing in using wood byproducts from Canada ([Damen and Faaij, 2006](#)), answering this question becomes even more critical. Next to the type of fuel used, the technology used to produce bioelectricity is important.

**Bioelectricity technologies investigated** The feedstocks and technologies investigated are shown in Table 8.6. What is interesting is that each feedstock and production method has different strengths and weaknesses. Each of them has a defined lifetime and may also only be viable at certain scales. Based on these characteristics, we may find the electricity production system resembling an ecosystem with different types of organisms that thrive in certain niches but may face stiff competition in others.

Table 8.6: Biomass Feedstocks and Bio-electricity Production Methods Investigated

Biomass feedstocks	Bio-electricity production methods
Demolition wood	Co-firing with coal
Wood Pellets	Gasification
Wood Chips	Combined Heat and Power
Refuse Derived Fuel	Anaerobic Digestion
Manure	
Palm Oil	
Rapeseed Oil	

**Systems perspective** Analyzing this from a large-scale systems perspective is important due to the global impact of increasing greenhouse gas emissions. For a country to achieve a meaningful reduction in greenhouse gas emissions, it cannot reduce emissions inside its borders while its economic activity leads to increasing emissions outside its borders. Addressing this means that at every step in the energy supply chain, there needs to be a record of the environmental and economic flows taking the form of materials, energy, emissions, and other environmental impacts. Using the Life Cycle Assessment (LCA) method, these results can then be aggregated to show the total flows and impacts resulting from the production of a functional unit such as 1 kilowatt-hour of electricity.

**LCA** LCA is a tool used to analyze “the environmental burden of products at all stages in their life cycle - from the extraction of resources, through the production of materials, product parts and the product itself, and the use of the product to the management after it is discarded, either by reuse, recycling or final disposal.”(Guinee, 2002) This comes from a recognition that environmental problems are systematic in nature and that by choosing one particular product or service, we are indirectly supporting environmental impacts that may occur several stages away from us in the supply chain.

**Attributing responsibility** From another perspective, LCA can be seen as a means of attributing responsibility for environmental impacts among actors in a supply chain. The owner of a coal-fired power plant may complain because his output is only in response to the demand from consumers and other industries (Heijungs, 2001). While such a perspective can easily lead to actors trying to blame others for environmental impacts, this underlying problem is very important for understanding sustainability. We need to remember that our technical systems have grown through the push and pull of supply and demand. These networks have emerged through the accumulation of choices made by a multitude of actors.

**Network tool** While it may not be immediately apparent, LCA is at its heart a type of network metric, or a means of calculating characteristics of a network. Specifically, it is a way of understanding a network’s structure from the viewpoint of one node within a network that represents the functional flow or reference product. Instead of trying to find the shortest path like Dijkstra’s algorithm (Dijkstra, 1959), it is trying to find all the upstream environmental interventions that can be attributed to the process represented by the root node. It keeps exploring the network until it can’t find any more emissions.

**Decision making aid** At a high level, LCA can be seen as a tool capable of either framing questions or directly answering questions already raised. For instance, framing may be needed when the nature of environmental problems are not be known. Once the questions are known, more detailed analysis can be conducted to locate the source of the problems of interest. On a practical level an LCA can be used for product and process development, by showing how to improve existing products or create new ones. It may also be used for labeling as a means to provide information to aid purchasing or design decisions(Ehrenfeld, 1997a).

**Limitations** The idea behind LCA is very powerful, although as we try to tackle larger and more complex problems, its limitations become more apparent. In its current implementation, it

views the world as being composed of static connections between technologies. It is also a linear model in that if production of one good is increased, the flows of the upstream technologies are scaled proportionally as well. LCA is commendable in that it addresses a very nontrivial problem. As will be argued further below, its current limitations can be overcome to an extent by using it within a new framework.

**Combining with ABM** From these examples, the main point to remember is that sustainability is the emergent property of a network, which LCA to an extent inherently recognizes. Sustainability cannot be measured at the level of individuals, it can only be measured at the level of the total system itself. However, LCA only examines environmental interventions in systems. The often cited economic and social concerns are left out of this analysis. On the other hand, an ABM is capable of capturing components of all of these aspects. By combining these two tools, a much richer analysis tool could be created.

#### 8.4.4 Details

**LCA data structure** The issues involving the combination of these tools can be better understood by comparing the data structures used by LCA and ABM. A typical data structure for LCA is illustrated in Figure 8.11. Most of the data is stored in two matrices. The technology matrix has information about economic flows between technologies, and the emissions matrix has information about the amount of pollution produced by each technology.

**ABM data structure** Figure 8.12 shows an example portion of the ontology (data structure) used in the ABM. The Technology Agent represents the entity (i.e., the 'company') that operates the technology. As seen, there are different design, economic, and physical properties that can be assigned to the technology. The Operational Configuration is the equivalent of an input-output table, showing the amount of goods needed for manufacturing, and the amount of goods produced. For a detailed description of the ontology please see Chapter 7. The flexible network structure between agents is facilitated by the In Edges and Out Edges that represent the contracts that an agent makes when it buys and sells goods. Since the other agents have the same type of data structure, we can see the contracts as facilitating the creation of a network of knowledge networks. This network is able to grow from bottom-up interactions, and algorithms can be applied to examine the structure that results.

**Hybrid approach** While an LCA could technically be done just on the agents in an ABM with limited system boundaries, the challenge is to create a hybrid approach, where the network is represented by both simulated agents and technologies defined in an LCA database. In short, an agent representing the "World Market" also has a connection to the LCA database. Whenever an agent buys goods from the World Market agent, it is actually obtaining those goods from the static supply chain defined in the database. This approach creates a simulation where there are dynamic supply chains in the foreground, with static supply chains in the background. In other words, we have a 'lens' of dynamic agent behavior in an otherwise static world defined by a database. This approach is illustrated in Figure 8.13.

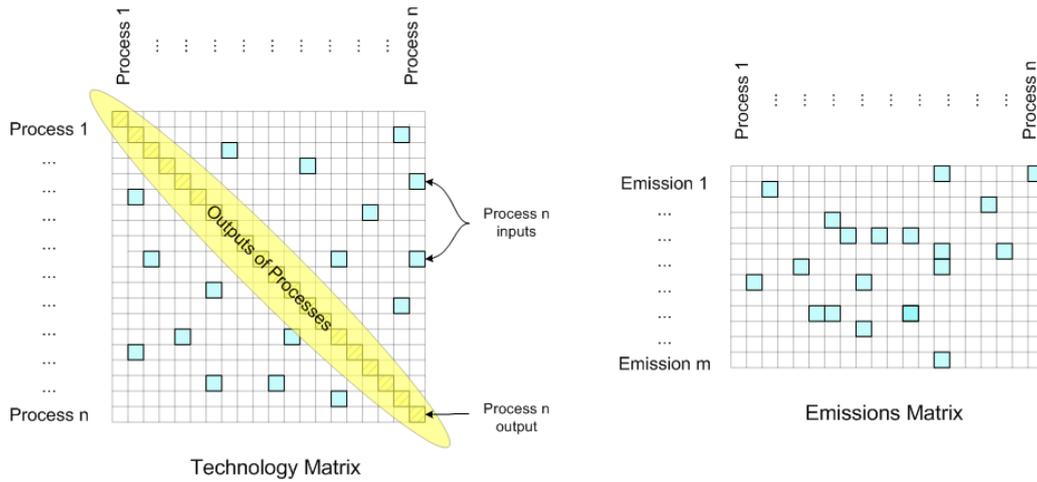


Figure 8.11: Illustration of LCA data structure, squares represent non-zero matrix components

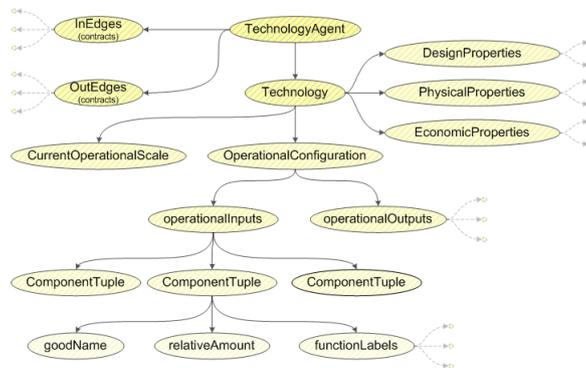


Figure 8.12: Partial illustration of ontology used for ABM data structure

**Combination** Figure 8.14 illustrates the data structure used to combine LCA within ABM. The simulation is paused at every time step, and all the information about inputs and outputs (i.e., traded flows and emissions) is retrieved from the simulation’s data structure. This information is then processed and placed into a technology and an emissions matrix where calculations can begin. At this stage, economic allocation will occur for technologies with multiple outputs (multifunctional processes). In both of the matrices, there is one section that is composed of information from the LCA database. This portion stays constant throughout the simulation. Additional rows and columns are added to represent the agents themselves. Since part of the matrix is composed of a dynamic system, the matrix will grow and shrink over time in accordance with the number of active agents present at each time step.

**Linking processes** Figure 8.14 also illustrates how the links are made between different processes within the technology matrix. The diagonal represents the total output of each process. By reading down a column, we can see the different amounts of inputs used for a single process. By reading across a row, we can see how one process supplies several different processes. The top left quadrant is composed solely of the LCA database, while the top right quadrant contains information where an agent buys goods from the World Market, which is connected to the LCA database. The bottom right quadrant shows trading between individual

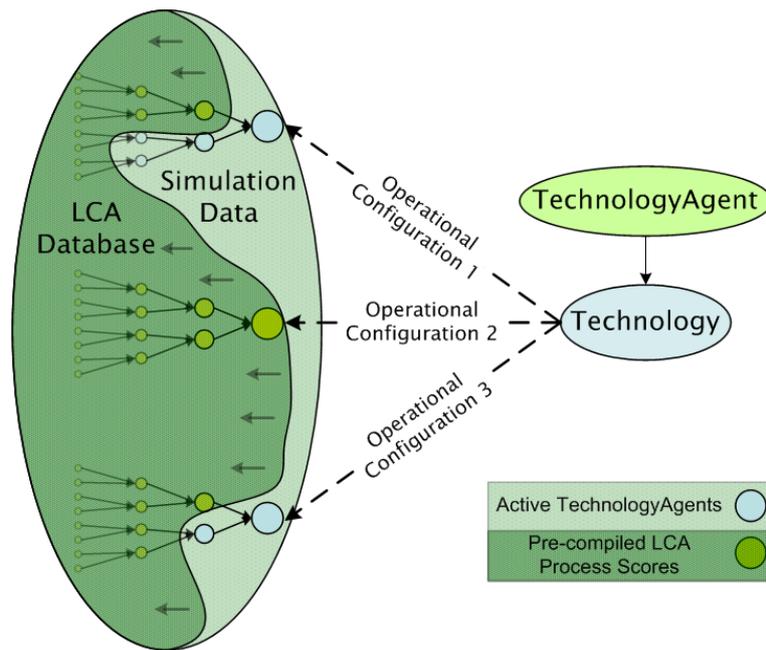


Figure 8.13: Integration of LCA into ABM

agents. The bottom left quadrant is not filled in, since the processes in the LCA database do not buy goods from the agents, and only use pre-defined supply chains. An emissions matrix is set up that is similar, except that the columns represent individual processes, while rows represent types of emissions.

**Calculation intensity** While a matrix makes these types of calculations computationally feasible, there are still issues with the amount of time needed, especially if one desires to perform a dynamic LCA as demonstrated in this paper. A major bottleneck lies in calculating the inverse of the technology matrix. In the worst case, most matrix inversion algorithms are said to run in  $O(n^3)$  time, meaning that as the size of the matrix increases by  $n$ , the maximum calculation time needed increases at a rate of  $n^3$ . In other words, a fraction of a second may be needed to solve a technology matrix of 100 processes, while two minutes may be needed for one involving 1,000 processes (Epstein, 1999).

**New algorithm** To overcome this problem, an iterative matrix inversion algorithm proposed by Peters (Peters, 2006) will be used. This particular algorithm runs in  $Cn$  time, where  $C$  is generally a very large constant. This means that algorithms running in  $O(n^3)$  time have an advantage when used for small matrices, but this iterative algorithm has a significant advantage in dealing with large matrices<sup>15</sup>. In practice, this algorithm has been invaluable in the creation of the model, considering that the number of LCA calculations increases with both the number of agents. The increase is nonlinear because the matrix increases as the number of agents increase and each agent needs to do a LCA at each time step.

<sup>15</sup><http://gux.tudelft.nl/svn/IndustryInfrastructureCoEvolutionModel/trunk/src/simulation/CalculateLCA.java>

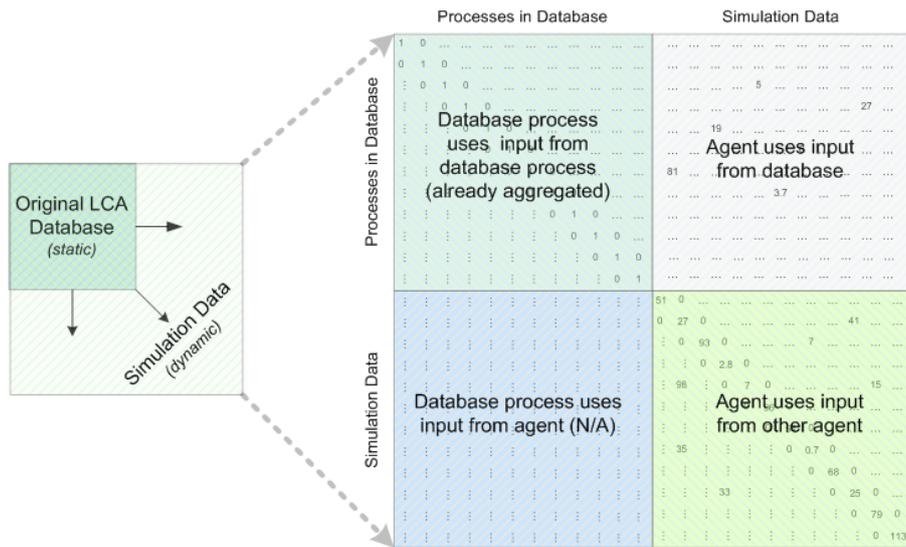


Figure 8.14: Merging technology matrices for LCA calculations

#### 8.4.4.1 Experimental Setup

The general experimental setup will now be detailed. The initial settings will be described, followed by the methods of agent additional and removal, and concluding with several of the parameters that will be varied. These parameters included varying a CO<sub>2</sub> tax and the decision-making strategy used by the agents.

**Initial Settings** Each simulation starts with three large fossil-based power plants. These technologies are coal co-fired with biomass, coal co-fired with syngas, and natural gas and heavy oil co-fired with bio-oil. Each of these large technologies is able to co-fire biomass at a fixed ratio, in addition to being able to use only fossil fuels. The agents used in the simulation all have different capacities for electricity production. In order to keep the number of agents in the simulation reasonable, some of the smaller technologies were scaled up to represent several instances of one type of technology.

**Electricity demand** The system is driven by the demand for electricity, which has a direct effect on the number of agents added. The actual initial demand for electricity is set slightly above the production capabilities for the three main fossil electricity plants, allowing for additional smaller producers to join in to fill the gap.

**Agent investment decision** Based on the decision algorithms developed in the past case studies, at the beginning of each simulation step, new Technology Agents have the opportunity to join the simulation. During this period, candidate types of Technology Agents are picked at random and asked whether they want to invest or not. Once ten in a row have declined, the simulation continues, since it is unlikely that any others may want to invest. A candidate Technology Agent's investment decision is based on several items. First, the demand to supply ratio for its reference product must be higher than a specified value to avoid oversaturating the market. If this condition is met, then the candidate will evaluate if it can be profitable based on

current market conditions by calculating its expected revenue from the sale of its products and then subtracting its fixed operating costs and variable costs of inputs. If this profit is positive, then the Technology Agent will decide to invest and join the simulation at that tick. Agents will be removed if they are consistently unprofitable for a specified length of time. They will also be removed if the lifetime of their technology has ended.

**CO<sub>2</sub> Tax** The CO<sub>2</sub> tax is directly related to the fossil CO<sub>2</sub> output for each of a technology's operational configurations. This CO<sub>2</sub> output refers to local emissions, not the ones indicated based on its LCA score. For the simulations run, the values of the CO<sub>2</sub> tax was swept in order to find transition points where system changes could be observed. The tax values range from 0 to 500 €/ton in increments of 25.

**Operational Decisions** Many of the agents have multiple operational configurations. Each of these configurations represents the use of a different feedstock. During each tick, an agent has to pick a single operational configuration to use. Each agent keeps a record of profit and LCA scores associated with each of its operational configurations. When the agent is initialized, it first collects this data by iterating through all its operational configurations on consecutive time steps. Essentially, the agent wakes up and assesses the economic and environmental state of the world from its own point of view. Once the agent has completed this stage, it will then select future operational configurations based on the decision behavior that has been specified for it. Agents are able to pick feedstocks based on those resulting in the most profit or the greatest reduction in CO<sub>2</sub> emissions. Additionally, they can apply weighting factors to these criteria and pick the feedstocks that best represent these preferences.

**Static and dynamic agents** Each of the Technology Agents had their material inputs and outputs based on data gathered during the creation of an LCA database. During the simulation, some technologies may be represented actively in the form of a Technology Agent, or passively as an instance in the LCA database. The Technology Agents are able to create their own dynamic supply chains, while the LCA database represents a collection of fixed 'pre-compiled' supply chains. A Technology Agent in a sense gives 'life' to this data, allowing interesting analysis.

**Learning via LCA** For the scenarios, LCA calculations are used as a basis for a decision which feedstock a technology to choose. LCA calculations are done for the each of the electricity producers in the simulation, with a chosen functional unit of 1 kWh of electricity production. These calculations are performed by using a mix of real-time simulation data and data from an aggregated pre-compiled LCA database ([Frischknecht and Rebitzer, 2005](#)). This database contains a list of goods and the upstream emissions resulting from their production. As the simulation starts, the agents have not yet begun to explore the mix of possible supply chains. In order to pick an operational configuration based on an LCA score, they consult the LCA database at this time. As the simulation continues, the agent makes its own connections and generates calculations of LCA scores based on the real-time data that it encounters, while using less information from the LCA database.

**MCA** An agent’s decision behavior is constant throughout the simulation and is based on an MCA-type algorithm. Both LCA scores and profitability are evaluated for all available Operational Configurations, with weighting factors applied to indicate the importance an agent places on each aspect. In the batch run of simulations, these weighting factors were swept in a linear fashion. In other words, in one run, agents will only try to maximize profit. In another run, they will place 80 % importance on profit, and 20% on LCA. This would continue until agents only try to maximize their LCA score.

### 8.4.5 Case study results

**CO<sub>2</sub> tax** Figure 8.15 represents the change in the fossilCO<sub>2</sub> / total CO<sub>2</sub> ratio after 20 steps of the simulation for a single CO<sub>2</sub> tax rate. Each value at the x-axis is a single experiment with a randomized lifetime of the technology. Data are sorted from the lowest to the highest value, and represent the maximum spread found. The red line represents the ration with pure economic decision making, blue the pure environmental decision making. Values below 0 represent an improvement. The figure clearly shows a consistent decrease in fossil CO<sub>2</sub> when environmentally based decision-making is dominant, but in order to achieve real systematic change, a CO<sub>2</sub> tax is needed. A transition is clearly seen between 225 and 250 €/ton. The high values come the situation where the three initial technologies (coal, natural gas, oil co-fired with biomass) start the simulation relatively young, meaning that the cleaner technologies don’t have a chance to take over.

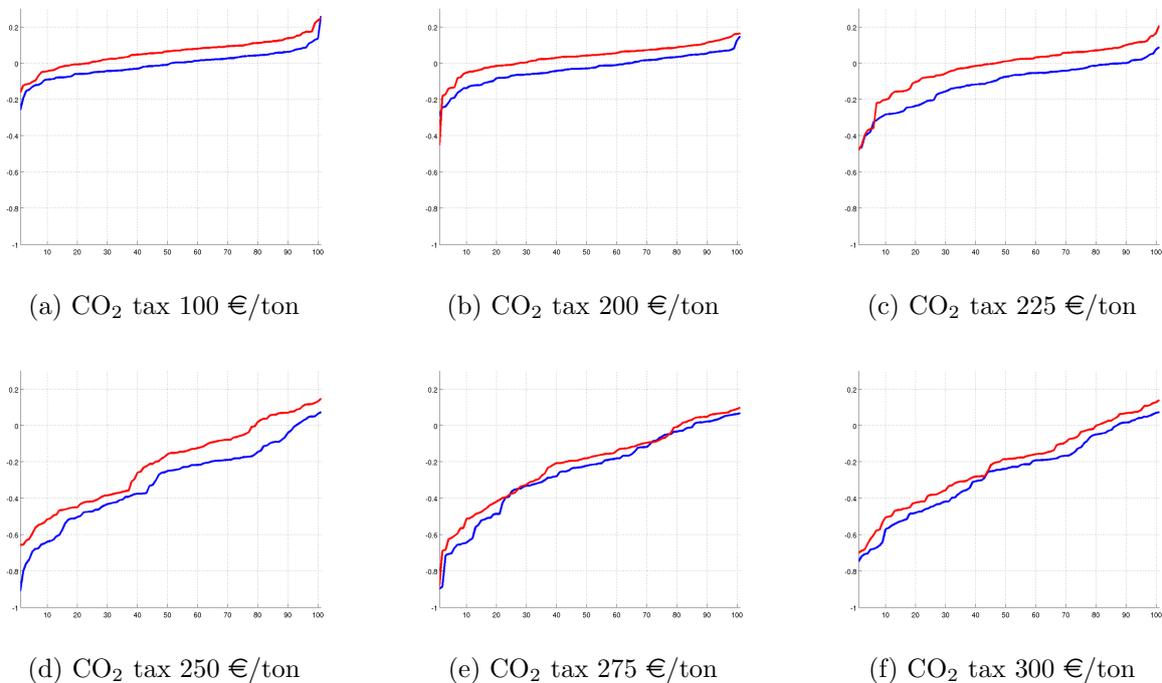


Figure 8.15: CO<sub>2</sub> emission levels at different tax rates

## 8.4.6 Domain-Specific Insights

**System change mechanisms** There are two primary means for the system to change. First, the overall portfolio of electricity producing technologies changes based on the CO<sub>2</sub> tax that was chosen. All agents would have to pay a tax based on their fossil CO<sub>2</sub> emissions. Secondly, agents could choose particular feedstocks in order to maximize profits or environmental benefit. In the simulation runs, all agents were specified to have the same behavior as a means to test the boundaries of these decision-making types.

**CO<sub>2</sub> reduction** From the simulation runs, it was found that the most effective way to reduce CO<sub>2</sub> emissions was to impose a high CO<sub>2</sub> tax rate, rather than only having the agents pick the feedstock that leads to the lowest emissions. This result was due to the fossil-based electricity agents defined as not being able to co-fire biomass above a certain ratio. This means that for drastic CO<sub>2</sub> reductions to occur, this limitation will be a barrier unless it can be overcome. Otherwise, given the assumptions and framework used for this case study, changing the portfolio of technologies is necessary to achieve these reductions.

**General value** In viewing these results, one should also remember that these outcomes were case study specific and thus tied to the definitions of technologies used. This hints that other case studies involving different industries with more complex networks of supply chains could lead to a much different interpretation of results. What has been shown is a proof of concept, and the future applications of the integration of LCA within ABM could lead to an analysis of systems in ways that have yet to be anticipated here.

**Tax is effective** From the simulations that were conducted, we were able to see that above a certain CO<sub>2</sub> tax rate, fossil-based electricity producers could no longer compete with bio-based electricity producers. When a high CO<sub>2</sub> tax rate exists, then the mix of electricity producing technologies shifts towards those that primarily use biomass. The effect of decision-making was rather minor compared with the reductions in CO<sub>2</sub> that could be achieved by changing the portfolio of technologies used by electricity producing technologies. It was also found that the best way to reduce overall CO<sub>2</sub> emissions of the system was through a change in the portfolio of electricity producing technologies. In other words, some electricity producers were physically limited in the amount of biomass they could co-fire.

**Future uncertainty is good** Comparing the probability of final outcomes under different tax rates and decision-making, Figure 8.4.6 was created by 400 simulation runs, showing 4 distinct sets of parameters (100 runs per set). The data used for the graph examines the % Fossil/Total CO<sub>2</sub> for the system. The x axis represents the %Fossil/Total CO<sub>2</sub> . 120 = 20% increase in Fossil CO<sub>2</sub> ratio, 100 = 0% increase, 80 means 20% decrease, 0 = no more Fossil CO<sub>2</sub> . The y axis represents time steps (beginning of simulation = back, end of simulation = front). For each step, a probability distribution function (assuming normal distribution) is constructed for each of the 4 sets of parameters. By combining these PDFs into a surface, we can see the probability that the system will have a particular value of %Fossil/Total CO<sub>2</sub> for each tick. The z axis represents the probability. Since the graph shows a PDF for each tick, the area under the curve (along the x axis) for each tick is 1. Each set of parameters is shown as a surface. From left to right, these are:

- CO<sub>2</sub> tax 0 €/ton, Agents maximize Profits
- CO<sub>2</sub> tax 0 €/ton, Agents maximize LCA score
- CO<sub>2</sub> tax 500 €/ton, Agents maximize Profits
- CO<sub>2</sub> tax 500 €/ton, Agents maximize LCA score

These show that the type of decision-making does have an effect on the overall performance. At higher tax rates, there is actually more uncertainty about where the system will end up. This is caused by the system undergoing transition and that the mix of technologies will be much more uncertain. Since this is a PDF, uncertainty isn't necessarily bad; what matters is the area under the curve in the regions we want to be in. Additionally, this graph shows the learning period that agents undertake when they first start (looking at the back of the graph). They switch through each of their Operational Configurations to gather data on profitability and LCA scores. Once this stage is complete, they begin applying their decision-making criteria.

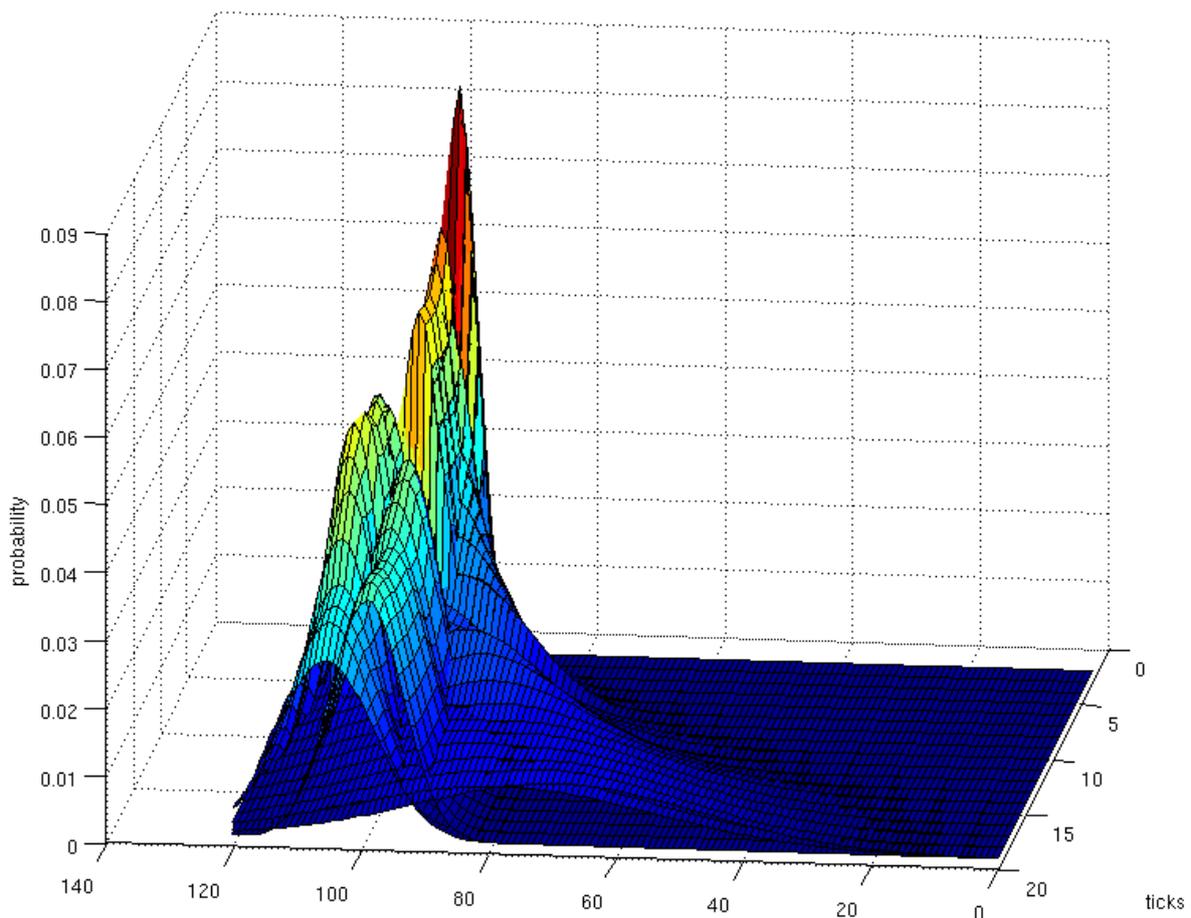


Figure 8.16: Probability density function. See text

## 8.4.7 Method Development Conclusions

**Combining LCA and ABM works** The most important methodological conclusion of this case study is that an LCA can be used within an ABM in a practical fashion. One of the major issues that had to be overcome involved keeping the computation time reasonable. As described more fully in the MSc thesis (Davis, 2007), this was achieved by using a much more efficient matrix inversion algorithm and using pre-calculated entries in the LCA database to avoid redundant computation.

**Learning from each other** Developers of ABM and LCA may find this merger of tools interesting from their own perspectives. From the LCA point of view, this means an ability to perform a dynamic LCA with the potential to overcome many of the obstacles they have identified. For those in the ABM community, this work shows a means to gain access to realistic abstraction of the outside world through the inclusion of an LCA database.

**Dynamic LCA** To understand why the work presented here is novel, one must understand the previous difficulties encountered in creating a dynamic LCA. While many people have worked on this problem, we argue that the limiting factor has been the set of tools they used to approach the problem. Heijungs and Suh (Heijungs, 2002) describe the difficulties in creating a true dynamic LCA, especially with regard to issues in the inventory collection, inventory analysis, and impact assessment stages. In the past, the task of creating a dynamic LCA appears to have been approached from a pure mathematical perspective based on frameworks that previously have been developed. While striving for mathematical elegance, we believe that these approaches miss a hybrid approach, where complex behavior is continuously generated by agents, and an LCA is performed at discrete time intervals. The key is the use of an ontology that allows for a flexible network structure to emerge. A matrix is extraordinarily useful when used as a tool for complex calculations like an LCA, but it is simply too constraining to use it as the primary data storage structure for the simulation.

**Limitations** A dynamic LCA only makes sense in certain circumstances. It also represents a level of complexity above a normal static LCA due to additional data requirements. The combination of ABM and LCA can be very valuable when there are many possible permutations of supply chains components that can lead to non-linear effects. The same is true if decision-making can lead to a wide variety of outcomes. If these conditions do not exist, then performing a dynamic LCA may not be very valuable. This is similar to LCA itself, where studies may involve different levels of detail based on the needs of the research.

**What does ABM gain from LCA?** Developers of agent-based models can benefit from concepts of LCA in several ways. Two such benefits will be explored below. First, LCA is a methodology that allows for a type of structural analysis to be performed. Secondly, the use of the LCA database in this paper demonstrates how this information can be used to abstract the world outside the simulation.

**Structural Analysis** The value of ABM lies in its ability to generate complex emergent behavior. While this is a benefit compared to other types of modelling techniques, it can also present a challenge, as one needs a means to analyze the systems that emerge. LCA is

one such method for achieving this. It is important to realize that LCA is really a particular implementation of a class of algorithms meant to analyze network structure. LCA builds on the achievements of the fields of ecology and economics, which have long been concerned with studying money, material, and energy flows within systems (Suh, 2005).

**Limited diversity** In this case study, many of the supply chains had the electricity production step as the most CO<sub>2</sub> intensive. Also, the larger technologies were limited in the amount of biomass that they could co-fire. So even though they had LCA knowledge available, their system was constrained in its ability to change unless the larger polluting technologies were killed off by a high CO<sub>2</sub> tax. This recurring problem of limited agent diversity is so far unsolved.

#### 8.4.7.1 Future work

The realization that LCA is a network metric opens the door for implementation of other types of analysis. Applying this method to other types of databases, such as those based on macro-economic data, could provide a more realistic context of empirical data for the agents to operate in. At a higher level, this merger is the result of trends in the development of information technologies, which are changing the ways in which we handle massive amounts of complex data. This is coupled with advances in understanding of Complexity Science and a shift towards what some would call the Generative Sciences (Epstein, 1999). The combination of these trends is worth following, as it may provide us a means to get closer to the originally stated goals of finding solutions that benefit people, the planet, and profit.

The technique implemented in this case study also allows the combination of Agent Based Models with CGE economic Computable General Equilibrium (CGE) models, such as the one presented by Duchin (Duchin, 2005).

#### 8.4.7.2 Requirements Checked

Table 8.7 presents an overview of the modelling requirements and the performance of the model. There is obviously a long way to go before all the requirements are fully met.

Table 8.7: Performance of the case study

Requirement	Score	Explanation
Method		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	Partial	The case study is performed as an extension the work for the government commitee. Only the scientiffic staff was involved in the case study.
Organically growing	Yes	The case study is based on the previous one and extends it organically.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful	Yes	Greatly increases the insight into environmental impacts of a simulated network and provides domain specific insights into bioelectricity production.
Testable	Yes	All outcomes are fully testable.



# Part III

## Insights



Facts are meaningless. You could use facts to prove anything that's even remotely true!

*Homer J. Simpson*

## 9.1 Results and Discussion

Having presented the work in the previous 8 chapters, we will now present and discuss the results of this work. First, the overall outcomes of the co-evolutionary modelling method will be presented and discussed, followed by the domain-specific insights. Next, the design method itself and the associated requirements will be examined, and the section will conclude with the outcomes and discussions on each of the four elements of the co-evolutionary modelling method.

### 9.1.1 Co-evolutionary Method Outcome

**Main result** The main result of this thesis is the design and implementation of a co-evolutionary method for creating continually improving models of  $\lambda$ -system evolution. It consists of a practical modelling method and a modular, expandable simulation engine for modelling  $\lambda$ -system evolution. The modelling method creates subsequently richer and more useful models. 'Useful' should be understood here as being able to answer a question that the involved stakeholder has, or to provide new insights into the modelling method itself. The evolutionary method was played out over 7 generations, each resulting in a case study depicted in Figure 9.1.

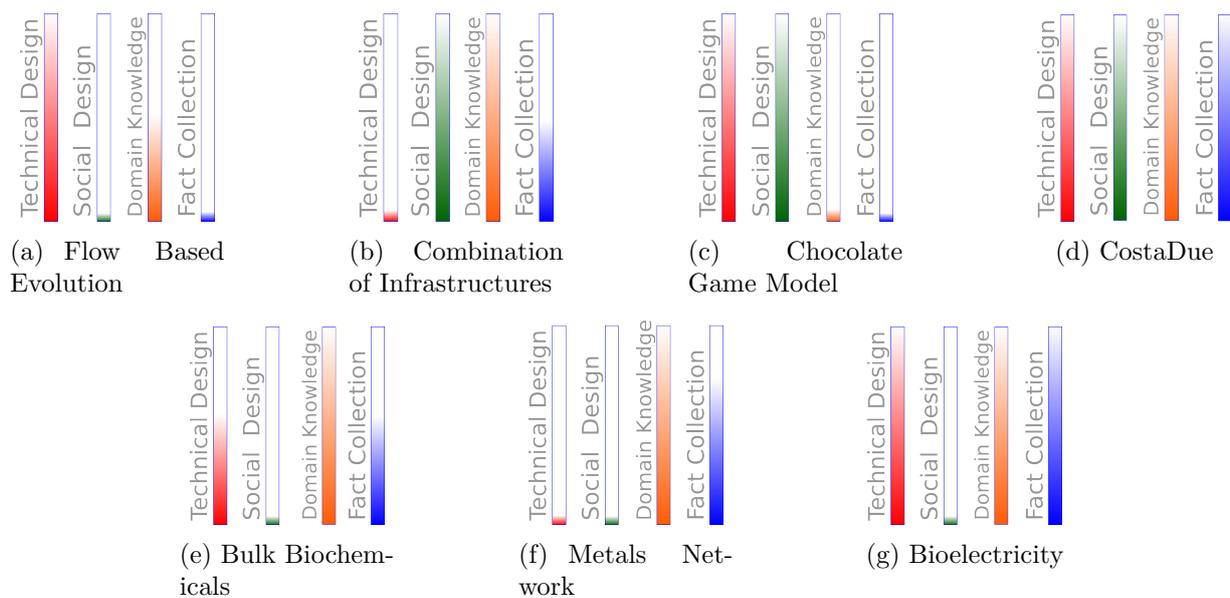


Figure 9.1: Evolution of models

**Co-evolution in a fitness landscape** The co-evolutionary modelling method, presented in chapter 5, is envisioned as a four dimensional coupled fitness landscape (see Section 5.2.2). In this landscape, four aspects of the modelling method interact: the technical design of the model, the social process for knowledge formalization and modelling, the knowledge formalized and facts collected. Whenever a aspect changes, the fitness of the other aspects is reduced. For example, when the technical design is extended to incorporate a new formalism, social process design aspect becomes less fit, as they are unable to encode this new type of knowledge until it is improved. The method progresses in generations, which consist of case studies, improving different aspects as it goes on. In each generation, changes in different dimensions create new possibilities and challenges for other dimensions, driving the overall process forward. The summary of the changes between generations is presented in Table 9.1.

Table 9.1: Cumulative overview of the evolutionary improvements over case studies

Case	Dimensions				Formalisms (new/total)
	Technical	Social	Knowledge	Facts	
Flow-Based Evolution	Agents with non-conserving mass flows	None	Rudimentary PSE	None	1/1
CoI	Combinability fitness landscape	Initial design	Legal, Spatial, Technical and Safety aspects of infrastructures	Combinability scores for 46 infrastructures	4/5
Chocolate Game	Agents with discrete mass flows and initial formal ontology	Initial SDM development	Basic economic	None	1/6
CostaDue	Simulation engine design, Agents with inference reasoning, Continuous mass and energy flows	Full application of SDM	Basic corporate finance, PSE, RDA strategy	27 technologies	0/6
Bulk biochemicals	Agents with MCA reasoning, LHS for parameter exploration	No additions	MCA concepts, RDA strategy	19 technologies	1/7
Metals network	IRR, NPV, temporary production shutdowns	No additions	Economic and RDA reasoning	29 technologies	0/7
Bioelectricity	Agents can perform LCA	No Additions	Environmental impacts	26 technologies, emissions data on 3000 processes	1/8

**First generation: Flow-Based Evolution** Using the notion of co-evolutionary across a coupled fitness landscape, we can examine the evolution of the case studies in this thesis. The first case study, the Flow-Based Evolution (see Figure 9.1(a)) started the evolutionary method by testing a technical design for an agent-based model of a simplified industrial network. The heights of the colored bars in Figure 9.1(a) denote the extent to which the case study deformed the fitness landscape. Flow-Based Evolution created the initial peak in the technical design and encoded a moderate amount of chemical process engineering domain knowledge. As it used a fact-free approach to model development, it did not encode any facts. It did, however, offer insights into which types of facts would need to be collected if one were to build a realistic model of industrial network evolution.

**Second generation: Combination of Infrastructures** The second generation (see Figure 9.1(b)) was the Combination of Infrastructures (CoI) case study, presented in section 6.3. This case study explored the social process of knowledge formalization and collected a moderate number of facts about the spatial combinability of different infrastructures. The case study laid bare the shortcomings of an ad-hoc approach to expert consultations and posed the challenge to create a structured and deliberate method of knowledge formalization. This generation did not improve the technical design, as it did not involve an ABM.

**Third generation: Chocolate Game** The third and final learning case study was the Chocolate Game (see Figure 9.1(c)). This study greatly improved the initial technical design of the model developed in the first case. It also developed the System Decomposition Method, based on insights from the second generation. The SDM is used to extract formalized knowledge from a group of stakeholders and domain experts and shape it so that it is usable in an ABM. The SDM was presented in section 6.4. It set a new and improved standard for knowledge formalization and solved the problem posed in the first case study of having a closed mass balance across agents. The case study did not encode any new domain knowledge, as it was based on conceptualizations developed in the first case study. Nor did it encode any new facts, as it was based on chocolate production, an analogy of the process industry.

**Fourth generation: CostaDue** The fourth evolutionary generation was the Costa Due case study (see Figure 9.1(d)), presented in chapter 7. This case study improved the technical model design by providing a mechanism for modular agents' economic reasoning and a modular description of technology. It performed a full-scale SDM with different domain experts, thus testing the method. It formalized concepts from the economic and technical domain necessary for formal description of industrial network evolution and encoded a large number of facts on chlorine, biobased chemicals and energy production processes. It maximized the fitness landscape across all dimensions, effectively resetting it. It provided a solid and practical base to continue the co-evolutionary method.

**Fifth generation: Bulk biochemicals** The fifth generation evolved further with the Bulk Biochemicals case, presented in section 8.2 (see Figure 9.1(e)). This case study improved the technical design by adding a Multi Criteria reasoning and introducing Latin Hypercube Sampling as a novel technique for model analysis. It made no changes to the social process

design. New knowledge encoded consisted of formalizing the Multi Criteria Analysis concepts, and the study collected a fair number of new facts on biorefineries.

**Sixth generation: Metals network** The sixth generation, the Metals Network case study presented in section 8.3 (see Figure 9.1(f)), modelled the evolution of a metals production network. The technical and knowledge improvements gained from the fifth generation allowed this case study to explore more sophisticated economic decision-making by introducing considerations of Net Present Value and Internal Rate of Return to agents and enabled agents to decide whether or not to join a cluster based on market conditions. It furthermore formalized concepts describing a dynamic world market. The case study encoded many new facts on metals extraction, processing and recycling.

**Seventh generation: Bioelectricity** Using these advances, the last generation consisted of the case study on Bioelectricity (see Figure 9.1(g)), presented in section 8.4. The case study created a model of bioelectricity production that incorporated the sophisticated economics of the previous generation and added environmental impact and reasoning to the agents. The main technical design contribution was the addition of Life Cycle Assessment to the model, allowing agents to consider environmental issues when making their decisions. The case study did not change the social process design and encoded a very large number of facts on environmental emissions of technologies. This case study is a base that will be used for future work as outlined in chapter 11.

### 9.1.2 Domain-specific Results

**Flow-Based Evolution** The first case study was a proof of principle model, and due to the simplifications made it was unable to produce interesting insights into the functioning of real world  $\lambda$ -systems. Yet the model did provide a wealth of model development insights.

**Combination of Infrastructures** The main domain-specific insight of the CoI case study was the robustness of social knowledge. Before the start of the case study, there was a lack of a systematic understanding of the issues within the Rotterdam Port Authority (RPA). Knowledge was fragmented and unshared, and departments rarely consulted each other for insights which, when shared, were nonetheless often mistrusted. The CoI project provided the RPA with a clear sense of what can and cannot be combined, based on expertise from many different parts of the organization. The interesting result was that while the study did not deliver surprises in terms of new combinations, it showed that the implicit, unsystematic 'gut feelings' of the involved experts functioned relatively well. This gut feeling might serve the RPA for some time to come before it eventually fails due to the organization's growing complexity.

**Chocolate Game model** The Chocolate Game case study used an analogy of an industrial network to create a serious game. While the game provided a wealth of modelling method insights, the analogy used was simplified to such a degree that no domain-specific insights were gleaned.

**CostaDue** The main finding of the CostaDue case was that the transition from a chlorine to a bio-based cluster, envisioned by the stakeholder, is unlikely under the current economic conditions. It does not appear that bio-based technological options will lead to a diverse bio-materials based cluster in Groningen, even under very low economic selection conditions. This outcome is dependent on the assumed survival of the energy-intensive incumbent industry in the region.

The importance of path dependency in cluster development was demonstrated, as was the very limited power that the RDAs have in controlling this evolutionary process due to their lack of control over which companies appears when in their regions.

**Bulk biochemicals** There are three main insights obtained from the Bulk biochemicals case study. First, the examined biorefinery cluster is likely to be economically successful over a wide range of economic conditions. Whether that positive cash flow is enough to recuperate investment costs and provide an acceptable rate of return is impossible to say, since the agent's economic model is too simple.

The second insight is that agent diversity matters. Experiments examining the difference in cluster behavior under different RDA styles show very little variation. The RDAs simply have very little or nothing to choose from, even when they strongly prefer certain types of technology over others.

The third insight is the importance of the long-term view. When using an MCA to choose which firm and technology to add to the cluster, the RDA only looks forward by a single time increment. This short-term view creates clusters that are 'stuck' in local maxima. In reality, an RDA has a somewhat longer time horizon than one year, which is shorter than the lifetimes of most technical installations.

**Metals network** The main insight gained from the Metals network case study is that the structure of large-scale industrial networks with limited technical diversity is robust across a wide range of world market scenarios. We do not observe radical changes in structure over time, only outcome spread caused by the randomness of the model. As there are very few technical options available for metals refining, the system cannot easily adapt to changing global economic conditions. This case study confirmed the previously obtained insights on the importance of diversity and long-term vision for both agents and the RDAs.

**Bioelectricity** The main insight from the Bioelectricity case study is that the most effective way to reduce CO<sub>2</sub> emissions of an energy production network is to impose high CO<sub>2</sub> taxes, rather than having the agents choose the feedstock that leads to the lowest emissions. This is caused by the fact that the technical design of fossil fuel burning plants limits the amount of biomass co-firing, and such firms cannot limit their emissions voluntarily. If a drastic CO<sub>2</sub> reduction is to occur, the technology mix of the electricity production system has to change, meaning that the incumbent co-firing plants will have to shut down, which will only happen under severe economic pressure.

**Overall insights** Based on the model runs, insights from Complex Adaptive Systems and evolution theory, seven general guidelines for managing an evolving regional industrial cluster can be given:

- Cluster development is strongly path dependent. The order of appearance of firms matters, and the RDA must be very careful to develop an understanding of future evolution patterns. RDAs must be aware of the fact that they are 'gardening' the region, meaning that newcomers feed off of the existing residents.
- Once established, a cluster's structure is robust. From a Complex Adaptive Systems perspective, once a stable attractor has evolved around a particular cluster structure, the chaotic and path dependent nature of the evolutionary process will tend to keep it stable by amplifying the initial success. Changing a winning team is not only a bad idea but is also very difficult.
- The environment, or context, of the cluster is very important. The RDA must be aware of the fact the cluster they manage, regardless of its size, is just a small component of a much larger global system. Changes in the external world can disrupt even the most successful and stable cluster and possibly even cause its complete collapse. Furthermore, the social, legal, institutional and regulatory environments can make or break a cluster, even if the right firm and technology mix is present.
- Even as clusters evolve under the watchful eye of the RDA, mistakes are inevitable. Incompatible firms will be attracted and good firms will turn bad. It is therefore very important the retain control of the land that is allocated. Selling land is a fast way to lose control of the cluster.
- Diversity of firm types and diversity in the technical options available to firms are essential for maintaining a cluster's ability to adapt. When conditions in the external world change, clusters with high diversity will be able to respond more quickly to them. Low diversity can potentially lead clusters into an evolutionary dead end.
- The importance of the long-term view must be emphasized. In order to be able to plan the evolution of clusters, RDAs must be able to look ahead by several generations of firms or technical installations. Given the average installation lifetime of 15 years or more, RDAs need to use a multi-decade perspective instead of the current multi-quarter one.
- Finally, RDAs must realize that cluster evolution is a matter of balance. On the one hand, too much top-down control will stifle change, and too many bottom-up initiatives will destroy the cluster's coherence. A mix, good enough for a given cluster, should be striven for.

### 9.1.3 Co-evolutionary Method Design

**Requirements** The previous section discussed the outcomes of the co-evolutionary method for creating models of  $\lambda$ -system evolution. We now need to turn to the method itself and examine it against the requirements defined in chapter 5. These requirements, together with the stakeholders and the modellers, form the environment that guides and shapes the co-evolutionary method. Throughout this thesis, after each case study, the state of the method was measured by applying the requirements and evaluating the performance of the four dimensions of the design at that moment in time. Appendix F presents the scoring of all case studies per requirement. We can see that from the CostaDue case study onwards all requirements but

the community diversity were satisfied. The individual requirements and the way the method satisfies them will be discussed below.

**Open source** The requirement of open source calls for the availability of all knowledge and data needed to perform the modelling exercise. The reason is that if one cannot open the hood and tinker with the engine, one cannot verify the engine and be certain that it is doing what it is supposed to be doing, and one is unable to change or improve anything. The model source code, formalized and unformalized knowledge and data are accessible to all involved parties. It is not yet open source in the sense of being available to the general public. The proposed opening of the code with a suitable public license will coincide with the defense of this thesis.

**Sufficient community diversity** The community diversity requirement ensures that a sufficient number of formalisms, in the form of different domain experts and modellers, is present in order to deal with the multiformal nature of the  $\lambda$ -systems we are attempting to understand. At this moment, the community of modellers and users is strong and expanding. As can be seen from the case performances in Appendix F, several cases are not considered to have a sufficiently diverse social network, as some disciplines and types of users are not represented. For example, risk perception and user interface design were not included in the models so far (for future work, please refer to chapter 11). We would also ideally like to see broader stakeholder participation, as this was lacking in some case studies for practical reasons. However, the modelling method is designed in such a way that it can accommodate a growing diversity of stakeholders and modellers, and the fact that some cases do not have this diversity is not a measure of the failure of the method, but a practical case setup issue.

**Organic growth** The requirement of organic growth is based on the concept of local optimization, that is, every change to the design should be directly useful and should improve a direct problem. Furthermore, organic growth is bottom-up, ensuring the involvement of all stakeholders, as the design is not laid upon them. Through the iterative application of the modelling method across 7 generations, the technical design, social process design, knowledge formalization and fact collection have grown and improved organically. The code base has undergone many expansions and several major restructuring steps. The social process has been created by improving the existing approaches. It continues to be developed from the inside out, with more and more formalisms being added and new facts being collected. We can conclude that the requirement of organic growth has been met.

**Recorded history** As the modelling method is co-evolutionary, it is path dependent and error-prone. The requirement of recorded history ensures that all changes to the software, formalized knowledge and collected facts are recorded and can be undone if necessary. A record of the history is fully available in the various Subversion repositories <sup>1</sup>, as demonstrated throughout this thesis. The repositories track all changes to source code, knowledge and data. Wiki<sup>2</sup> tracks all tacit knowledge and its development. We can conclude that this method requirement has been fully met.

---

<sup>1</sup><http://gux.tudelft.nl/svn/>

<sup>2</sup><http://wiki.tudelft.nl>

**Enforceable authorship** The co-evolutionary modelling method is performed by people. Based on the previous requirement of recorded history, we also need to know *who did what and when*, both for credit and blame. The social network that emerged during this project numbers several hundred individuals with different degrees of activity. Every contributor is authenticated to the system and their contributions recorded. This requirement has been met by the co-evolutionary method.

**Modular** The requirement of modularity ensures maximum flexibility in the design, as it jointly enforces a standard interface between components and allows for the piecemeal replacement of components that become inadequate. Given the modular technical design described in appendix D, the state-based SDM discussed in section 6.5 and the discrete entities that compose the ontology, we can conclude that this requirement has been met.

This modularity will become more important as the method continues in the future and as the models increase in complexity. It poses a challenge, as the interfaces across which the modules interact need to continually evolve in order to accommodate new and unforeseen types of knowledge. Path dependency is a threat, and a redesign of the model interaction may be necessary in the near future (see chapter 11).

**Useful** The usefulness requirement has been defined at two levels: that of the method and that of the model's outcomes. A co-evolutionary step in the method must either be useful in further improving the co-evolutionary method itself or by providing new insights into the modelling method. For example, the addition of IRR and NPV in the metals case (section 8.3) allows for increased sophistication of the models. In terms of useful insights, the CostaDue case demonstrated that the future bio-based cluster envisioned by the stakeholders is unlikely to emerge under the current conditions.

Usefulness, as a requirement, cannot be measured absolutely. It is a relatively soft criterium, and it is up to the involved stakeholders and modellers to determine when something is useful.

Despite its relative softness, this is possibly the most important requirement of the method. It has been the major driver of the locally optimal development, as it called for immediate usefulness of every and any change in the different dimensions. It continues to drive the evolution forward, but ensures that no overengineering occurs, specially when the goals are set too high, i.e., "it will be great one day when we implement it...". Based on the developed models and domain insights, we can conclude that this outcome requirement has been met.

**Testable** The requirement of testability ensures the scientific nature of the entire exercise. If a developed model or method cannot be tested, it cannot be verified, and thus it cannot be called science. All models produced by the method are fully repeatable (based on a fixed random seed) and can be falsified. The modelling method itself can be repeated. As it involves a social network, the current accumulated knowledge cannot be unknown, so the new models will always be created with the advantage of hindsight. In addition, the involved stakeholders might be different, so the precise outcome will be different. But the method can be tested to produce useful results again. The evolutionary nature of the method furthermore means that it is tested at each co-evolutionary step, as the mechanism is reapplied. This continuous testing ensures its long-term viability. This final requirement has also been met.

**Co-evolution with the environment** As mentioned above, the requirements together with the users form the environment of the co-evolutionary method. It is the stakeholders, the modellers and the computers used to run the simulation that form a socio-technical environment that applies the requirements and performs the deformation of the fitness landscape. It is interesting to note that this socio-technical environment is also affected by the method, learning from the method (as this is the main goal of the exercise) and adapting to it. The method of co-evolution happens not only within the dimensions of the method, but also between the method and its environment.

**Main scientific contribution** The main scientific contribution of this thesis is the creation of the co-evolutionary modelling method, along with the associated method requirements. This allows the decision makers to operationalize their thinking on evolution on  $\lambda$ -system and complex adaptive systems, and aid them in the decision making process

This thesis is about the method and practice of creating new *combinations* of existing elements, and making those elements interact in a explicit, testable, repeatable and practically useful manner. By accepting the complex and evolutionary nature of both the object of study *and* the scientific process needed to understand it, new ways of organizing and operationalizing the thinking on  $\lambda$ -system evolution become possible.

Previous to this work there was no established body of knowledge on the modelling design method for developing Agent-Based Models in general. The focus has traditionally been on the descriptions of models and their results (see Appendix G for an extensive literature review). The most cited methodological literature (Barreteau et al., 2001; Bonabeau, 2002; Kendall et al., 1996; Wooldridge et al., 2000) is either very general for the purpose of introducing ABM to non-modellers, or deals with the design of multi-agent systems, which has distinctly different goals (see section 4.3.1). There is no literature related to methodologies for modelling evolving  $\lambda$ -systems. Methodology issues in general have been discussed in design literature (Dijkema, 2004; Herder, 1999; Westerberg et al., 1997), but the focus has usually been on conceptual process design, not modelling design.

**Four dimensions** The main result of the thesis was presented above, and we will now discuss the four dimensions that make up the co-evolutionary method. We will start with the technical design, followed by the social process design and the knowledge formalization, ending with the fact collection dimension.

#### 9.1.4 Technical Design

**Main result** The main result in the technical dimension is the design and implementation of the simulation engine, presented in Figure 9.2. For an extensive discussion on the diverse elements, please refer to Appendix D.

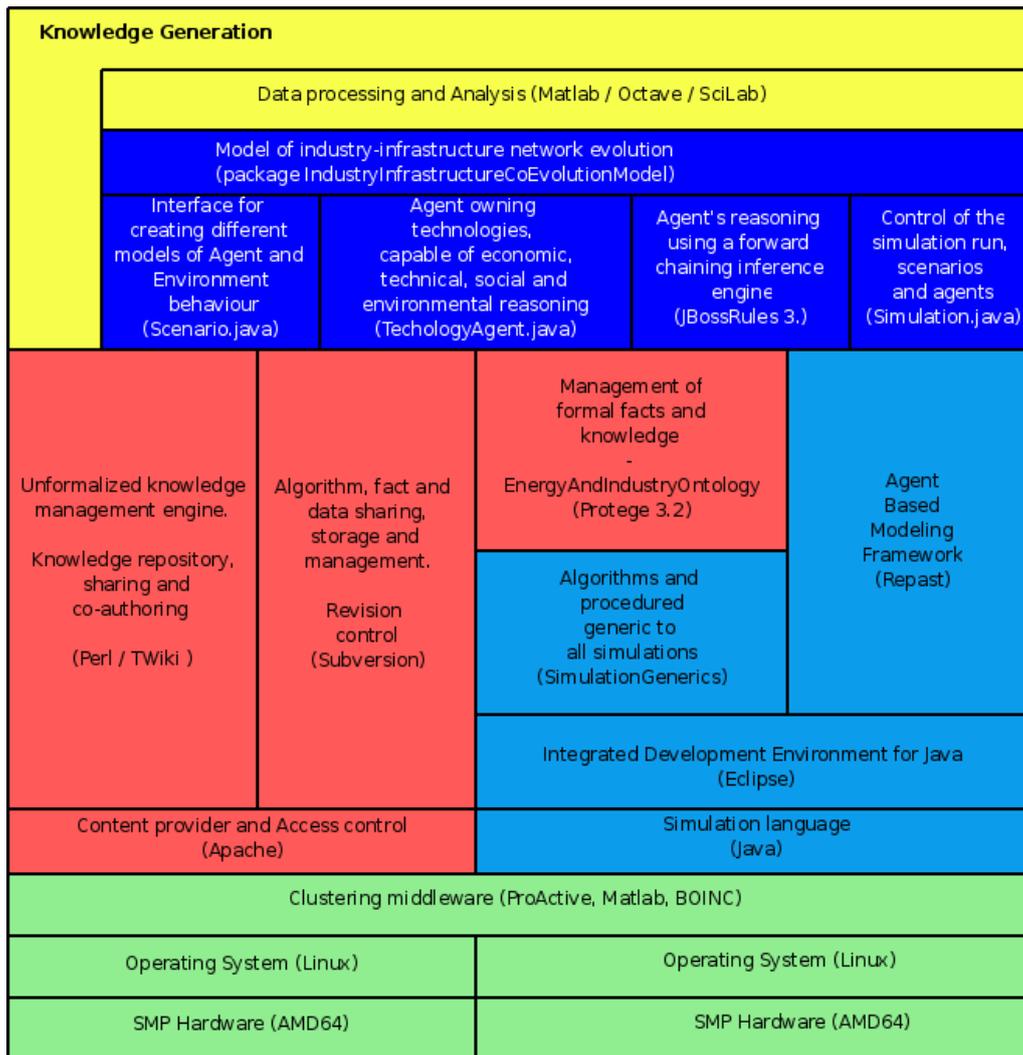


Figure 9.2: The structure of the simulation engine. Green areas are the Hardware and Operating System. Red areas are the knowledge management components. The simulation software components are in blue, and data and knowledge processing & analysis components are shown in yellow.

**Estimated effort** The iterative application of the requirements during the co-evolutionary method discussed earlier provides us with a sense of quality and usefulness of the simulation engine that evolved. A more quantitative estimate of the simulation engine code base can be achieved by analyzing the code using D. A. Wheeler's 'SLOCCount' program<sup>3</sup>. The SLOCCount program, often applied in open source projects (Deshpande and Riehle, 2008), estimates the time, effort and cost involved in creating a particular set of source codes. The results are presented in Table 9.2.

<sup>3</sup><http://www.dwheeler.com/sloccount/>

Table 9.2: Costs and effort estimates for the model code

Aspect	Model	Sim.Gen.	Total
Total Physical Source Lines of Code (SLOC)	10674	14159	<b>24833</b>
Development Effort Estimate, Person-Years (Person-Months)	2.40 (28.84)	3.23 (38.80)	<b>5.63 (67.65)</b>
Schedule Estimate, Years (Months)	0.75 (8.97)	0.84 (10.04)	<b>1.59 (19.01)</b>
Estimated Average Number of Developers (Effort/Schedule)	3.22	3.86	<b>7.08</b>
Total Estimated Cost to Develop	\$ 324,627	\$ 436,743	<b>\$ 761,370</b>

Considering that the code was produced over the duration of a Ph.D. thesis (almost 5 years including writing) and that on average at any given time one person was writing code, the development effort estimates are a fair representation of the actual effort, as is the estimated number of developers.

**No energy balance** The main aspect of the description of the physical reality missing from ABM simulation software is the absence of the closed energy balance. Implementing a closed energy balance in a network describing processing industries is a daunting task, requiring a massive data collection effort, orders of magnitude larger than closing the mass balance. The main reason for this is the fact that energy use and transformations are pervasive in the processing industry, as well as the fact that energy can have many different forms. For example, in order to have a closed energy balance, one would have to consider all the temperatures and pressures, the energy content of all in- and outflows and also the electric energy entering the production plant. Furthermore, efficiencies of all equipment would need to be known, in order to estimate the losses to the environment. Different equipment for processing identical materials flows can have energy efficiencies differing by orders of magnitude. Finally, parts of the mass inflows are often used for energy generation as well as for processing, as is the case in refineries, for example, further complicating the mass/energy link.

While the energy balance is not implemented, this does not pose a significant problem at the level of abstraction used in the models. Mass balances in the form of fuel and CO<sub>2</sub> flows, and the energy flows in the form of electricity, add sufficient realism to describe the networks without burdening the modelling method with unnecessary data requirements.

**Main scientific contribution** The scientific contribution of this dimension lies in the actual design of the entire software stack, from the silicon up. Traditionally, software tools offered for ABM are either very generic or very specific and focus only on the agent description. There is no literature on the design of scalable, high performance agent-based modelling systems that simultaneously encompass agent description, formal ontologies, supporting knowledge infrastructure and choices of hardware and operating system. The design presented in this thesis can be seen as a set of interconnected and interactive tools, effectively forming a scientific Lego that enables multiformal modelling of  $\lambda$ -system evolution.

### 9.1.5 Social Process

**Main results** There are two main main results of the evolution of the social process design. First is the System Decomposition Method (SDM), discussed in detail in sections 6.5 and 7.6.1. The second result is the actual social network established around the co-evolutionary modelling method and the developed models.

**SDM** The SDM is a practical and structured collaboration script that extracts unformalized knowledge residing in the heads of domain experts and stakeholders and converts it into an agent-based model. It is presented in Figure 9.3 below.

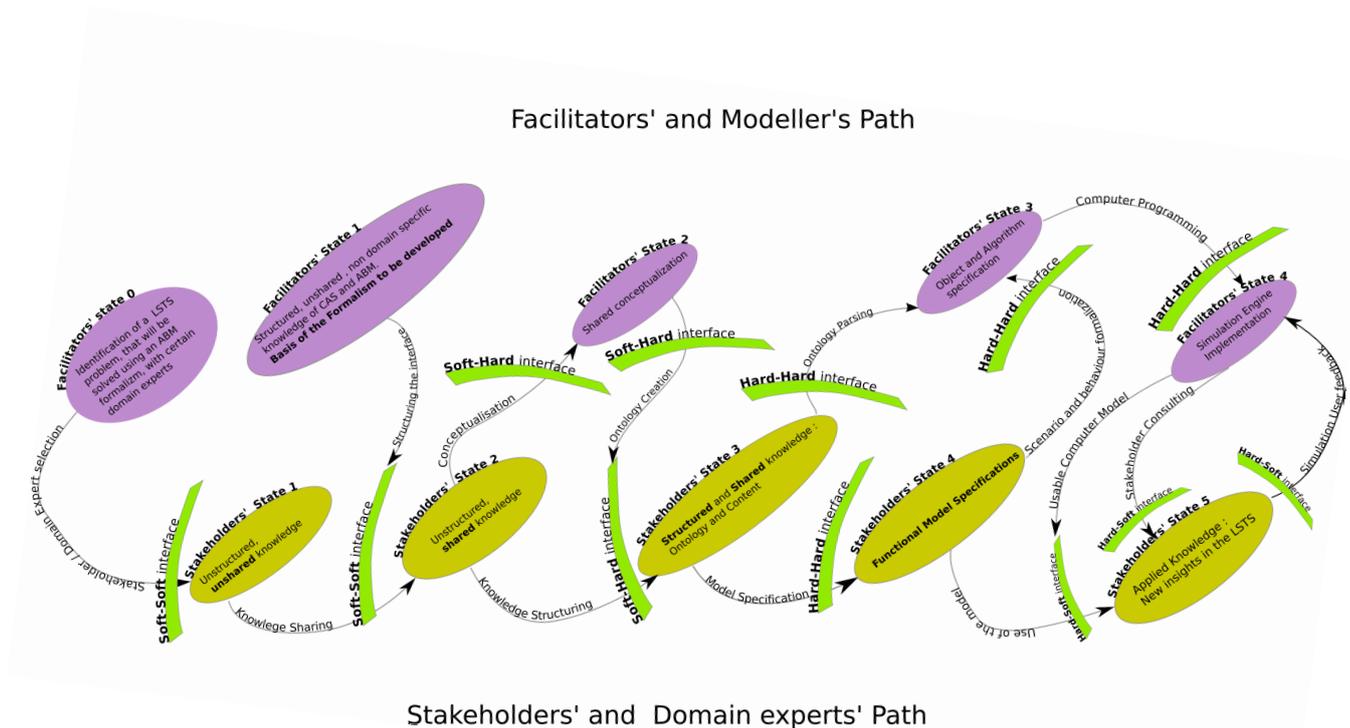


Figure 9.3: System Decomposition Method

The SDM method consists of a number of knowledge states and knowledge interfaces (see section 6.5). Starting with a stakeholder’s question, knowledge about a  $\lambda$ -system’s state and behavior moves from the unshared and unstructured knowledge held in different minds, across the soft-soft interface to a shared unstructured state. From there, the knowledge moves across the soft-hard interface, where it is structured into an ontology, becoming shared and structured. This knowledge then crosses the hard-hard interface, becoming a model specification. The final transition is across the hard-soft interface, where the model is used by the stakeholder to answer the initial question.

**Issues with the SDM** There are several aspects of the SDM that deserve further attention. These are the absence of a hard to soft interface, the absence of a proper social science analysis

of the social process and the pragmatic approach to network growth. These will be discussed below.

**Hard to soft** The SDM as presented in this thesis has one missing link, namely across the hard to soft interface. This is the step in which the formal outcomes of the models are transferred to the stakeholders' minds. This step is important for the optimal support of stakeholders in their decision-making. It has not been explored in this thesis in detail, as it requires a thorough social science and psychological treatment. In order to study it, one would need to examine the parameters governing the stakeholders' acceptance of models in general, their ability to deal with complex information, etc., which would be beyond the scope of this work. This thesis assumes that the modellers will engage the stakeholders in an ongoing discussion about the meaning and relevance of model outcomes without formally specifying the interaction.

**No model use analysis** This thesis did not perform a comprehensive social science study on the use of models by the stakeholders. The feedback from the stakeholders has been informal in nature. Again, a thorough analysis of the actual impact that the model outcomes would have on the behavior of decision makers is outside the scope of this work. One important observation is that being involved in the SDM method was seen by stakeholders as a valuable contribution to their understanding of  $\lambda$ -system evolution.

**Pragmatic approach** As a consequence of the previous issues, we should point out the pragmatic approach used to design the ABM and the pragmatic choice for evolution of the social network. These were not ideal, but were practical and useful. They do, however, suffer from traditional 'engineers meddling with social science' problems. One consequence of this pragmatic approach was that involvement was limited to that by the domain experts who were directly available. A more thorough and systematic social analysis could have possibly suggested a more diverse set of domain experts to be included, which would have further enriched the social network and allowed for formalizations of more diverse knowledge domains. Again, such an analysis was beyond the scope of the work.

**Social network** In addition to the SDM, the second result of the social process is the social network that has self-organized around the co-evolutionary modelling method and case-specific models. This was not a designed outcome, but emerged as a byproduct of the modelling method. This social network was organized around [wiki.tudelft.nl](http://wiki.tudelft.nl) and collaboratively created the model code and collected the facts. The wiki has been organized, maintained and administered by the author. The collaboration structure of this social network has been visualized as a force-directed graph in Figure 9.4<sup>4</sup>. The closer a user is to the center of the graph, the more intense her collaboration is with the others.

---

<sup>4</sup>Figure was developed together with Chris Davis, see <http://wiki.tudelft.nl/Research/WikiCollaborationNetworkVisualizationBot>, and the source code is available at <http://gux.tudelft.nl/svn/IgorNikolic/tools/WikiAnalysis>

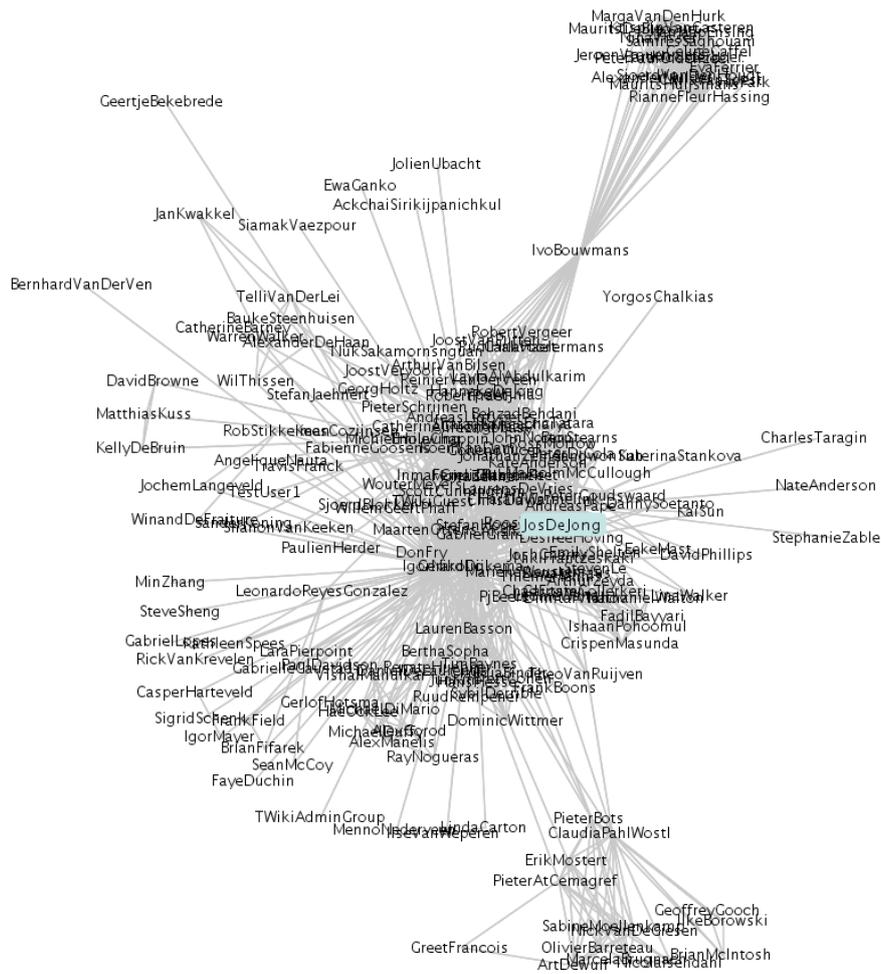


Figure 9.4: Collaboration structure of the social network on wiki.tudelft.nl

Figure 9.4 provides interesting insights into the social network. It is clearly a preferentially attached graph, with an active core of collaborators and a less active periphery. This corroborates the practical experience of wiki collaboration. Furthermore, if we examine the activity of individual users, presented in Figure 9.5, we can observe that the activity has a power law distribution which is common in Complex Adaptive Systems. Thus, not only does the social network study Complex Adaptive Systems, but it is one itself.

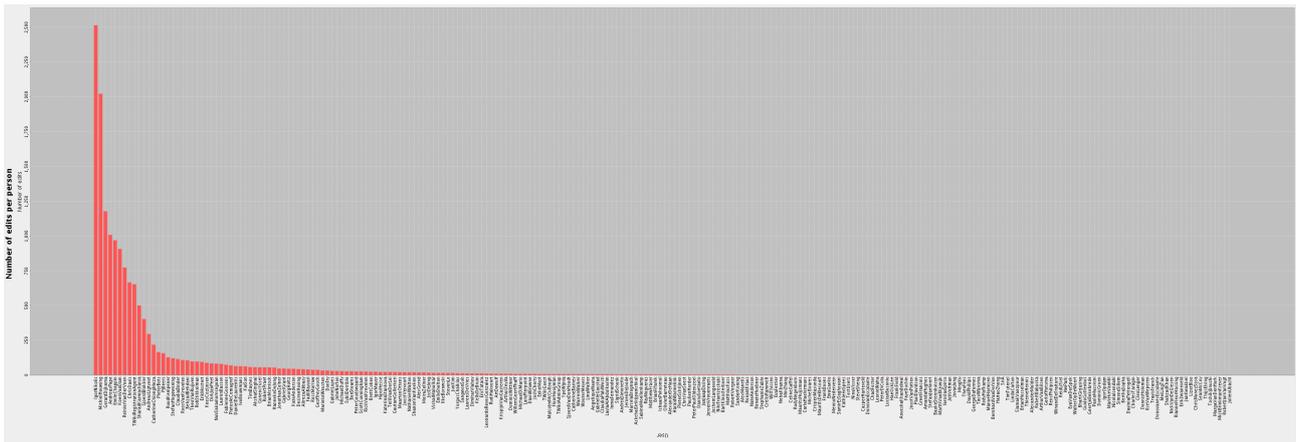


Figure 9.5: Wiki activity in number of edits per user

**Other models** In addition to the models presented in this thesis, the social network has produced a number of other models, all based on the Simulation Generics model core. A series of power sector models were constructed by Chappin (Chappin, 2006; Chappin and Dijkema, 2009). In these models, investment in electricity generation technology is central. Also modelled are power and fuel trade, and different carbon policies such as emission-trading and carbon taxation have been implemented to assess their potential impact on the system. Power-producing agents strategically invest or dis-invest in power plants with unique sets of criteria. In addition, electricity-producing agents exhibit operational behavior by negotiating and engaging in contracts. The modelled system also contains other agents, e.g. government, markets for CO<sub>2</sub> rights, power and fuels, power consumers and the underlying technology (power plants, consumer technology, physical networks and flows) (Chappin and Dijkema, 2009). Scenario analysis was used to operationalize the parameter space and outplay system developments under different exogenous conditions and trends.

An entirely different direction toward which the base model has evolved is a micro model of combined heat and power generation in households that has been constructed in order to study the dynamics of distributed power generation (van Dam et al., 2008). Similarly, a detailed model of an oil refinery was developed, modelling both the social and technical aspects of an oil firm’s operations. A more infrastructure-oriented model is the description of a container transport hub, examining the ideal location of a large-scale transport hub, given distance and cost as well as noise and pollution disturbance for the surrounding population (Sirikijpanichkul et al., 2007).

Finally, an Internet-based electricity market game <sup>5</sup> was designed in order to help players better understand the short- and long-term dynamics of electricity markets. In the game, several players work together as a power company and compete against other companies by generating and selling electricity. Players compete in a power exchange, where they try to sell their electricity. The game, while obviously requiring very different software than the agent-based models developed, is based on the Simulation Generics and on the ontology that describes the necessary concepts and game setup.

<sup>5</sup><http://emg.tudelft.nl>

**Main scientific contributions** The scientific contributions of the outcomes from the social process design evolution are twofold. The first contribution concerns the formalization of the method and the context in which it is performed. The explication of the interfaces and knowledge state of a system decomposition method, presented in section 6.5, is novel. A publication dealing with the interfaces and states is currently under review (Beers et al., 2009). The second contribution is the application of the SDM. Application of a systematic, socially inclusive system decomposition is new in the context of agent-based modelling of evolving  $\lambda$ -systems and ABM in general.

### 9.1.6 Knowledge collection

The third dimension of the co-evolutionary method is the knowledge collection and formalization. There are two types of knowledge collected: the formalized and the tacit. The first type is collected in the ontology and was the primary focus of this thesis. The unformalized, tacit knowledge was used to support and organize the social network around the knowledge formalization efforts and has mainly taken place on the Wiki. Obviously, the knowledge stored in the Wiki is not real tacit knowledge, since by definition tacit knowledge is not stored. Yet because of the free form collaborative nature of the Wiki, its concepts are as close as one can come to written tacit knowledge. Both types of knowledge stored are evolving, growing structures.

**Unformalized knowledge** The Wiki has evolved to be an on-line clearing house of information, a free-form workspace and a repository of near tacit knowledge. It is used daily to communicate project progress, make notes on idea development, collaborate on papers, etc. Its structure has emerged one page (topic) at a time. Each page roughly represents one coherent unit of content or idea. Figure 9.6 presents the growth of the Wiki structure over time<sup>6</sup>, which shows the development from its beginning up to the moment of this writing. The graph is force directed, edges represent pages referring to each other. This means that portal type pages, collecting information on certain topics, are in the middle. At the time of writing<sup>7</sup>, the Wiki contains 3151 pages.

---

<sup>6</sup>Graph is created by C.B. Davis, see the Wiki page at <http://wiki.tudelft.nl/Research/WikiGrowthOverTimeBot>. The source code of the generation algorithm is available at <http://gux.tudelft.nl/svn/IgorNikolic/tools/WikiAnalysis/src/VisualizeWikiEvolution.java>

<sup>7</sup>25<sup>th</sup> November 2008

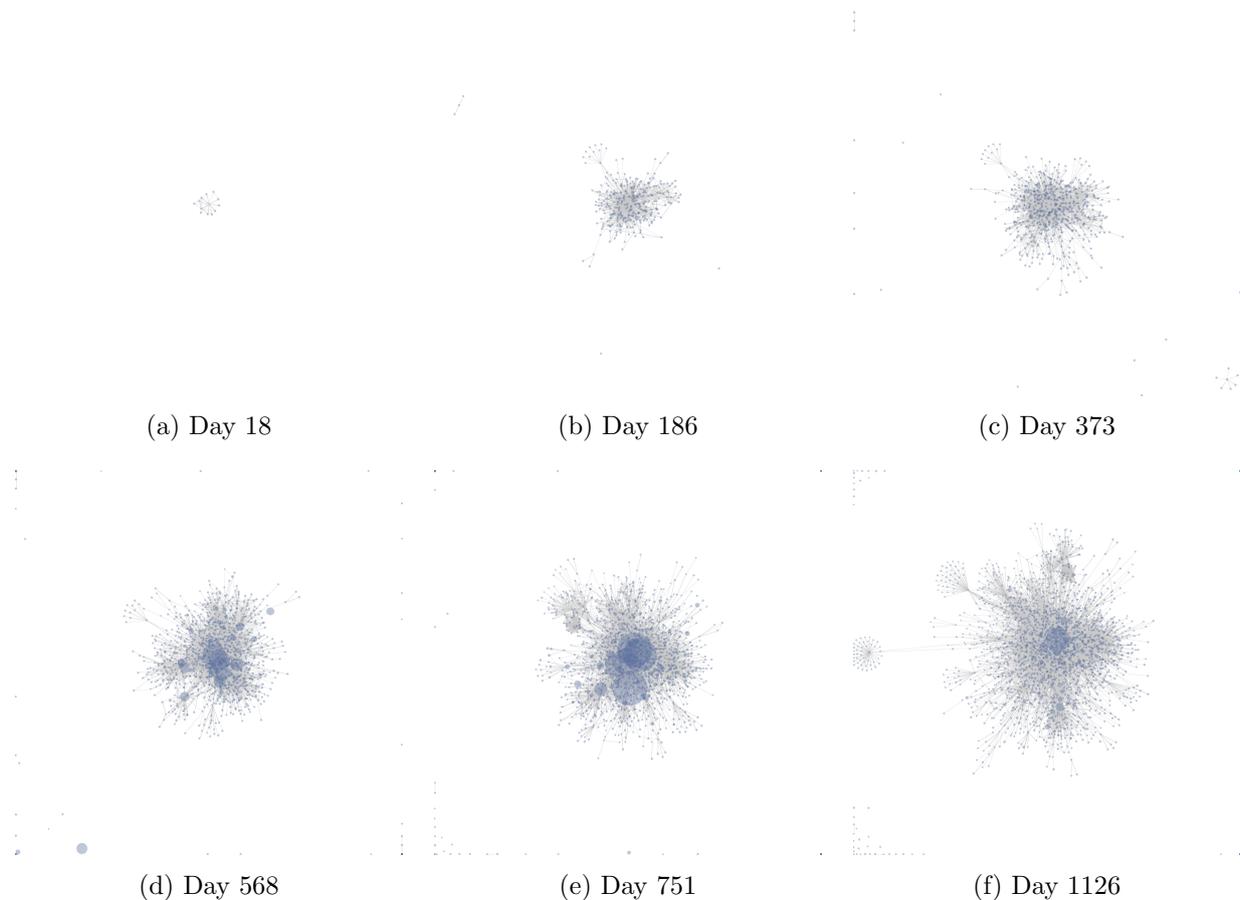


Figure 9.6: Structure of knowledge on the Wiki at any given day since its start. Size of the circle denotes relative activity on that page at that moment.

Figure 9.6 demonstrates the organic growth of the wiki and the preferential attachment growth pattern. Coupled with the social collaboration network presented in Figure 9.4 and the user activity histogram presented in Figure 9.5, the idea that the social network is a Complex Adaptive Systems is reinforced.

**Formalized knowledge** The difficulty in going from unformalized to formalized knowledge is reflected in the difference in number of formalized concepts. The Wiki contains around 3500 concepts, of which the formalized ontology defines 250 classes and some 160 properties (see Table 9.3). However, the Wiki is not computer readable, other than its structure, whereas a computer can use the ontology to reason about the knowledge encoded in it. The structure of the ontology is presented in Figure 9.7. A full-resolution image is available on-line <sup>8</sup>.

<sup>8</sup>[http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/EnergyAndIndustryKnowledgeBase\\_clean.png](http://gux.tudelft.nl/svn/IgorNikolic/phd/thesis/trunk/EnergyAndIndustryKnowledgeBase_clean.png)

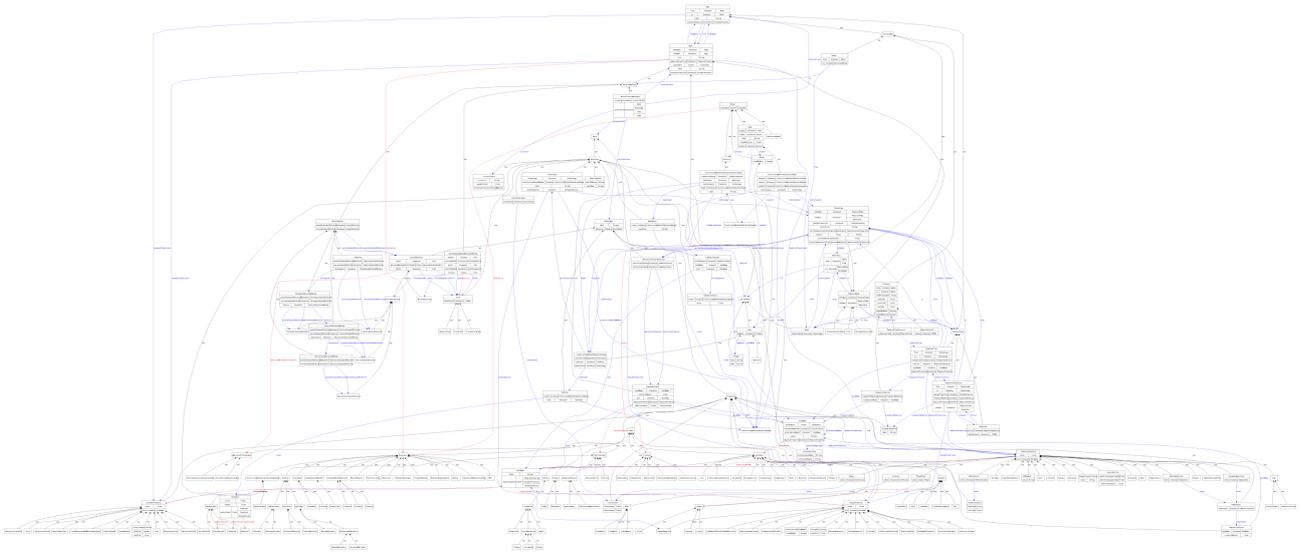


Figure 9.7: Structure of the ontology

**Ontology of actions** The ontology formalizes knowledge of *what* the system components are in a  $\lambda$ -system. What it is unable to encode is *how* these components act and interact. What is missing is an ontology of agent actions. There is a significant difference between creating an ontology of objects (the *what* or declarative level) and an ontology of the action (the *how* or procedural level). Objects can be defined as having a limited number of useful properties, and the list of properties can easily be expanded as the need arises. Actions, however, demand a more thorough treatment. The trouble with actions is that one must not only consider a very broad set of things acted upon (that is, the objects), but also what the action is, (that is, what happens), and what its effects are, worded in terms of objects, their properties, and possibly even other actions. In words probably more familiar to programmers, one must consider all possible operators and associated data structures. This amounts to writing a specification of a programming language, which was beyond the scope of the system decomposition method. One (partial) way around this problem is the specification of the Scenario class, as defined in chapter 7. The scenario defines a template of the actions of which the agent is capable. Although this does not fully specify how an action is to be performed, it does offer a unified interface between agents. A new approach using Aspect Oriented Programming is currently under development to alleviate this problem (see chapter 11).

**Overscripting** Finally, there one important danger in using formal ontologies. A traditional pitfall of using formalisms lies in the so-called over-scripting of a collaborative activity (Dillenbourg, 2002). This means that the structure imposed on the collaborators is so strict that instead of improving productivity, it stifles creativity and innovation. If we hold the SDM against this light, it would seem that its large degree of genericness and its small number of initial concepts would keep it from stifling creativity. Balancing flexibility with exactness is more of a black art than a science, and the right mix should evolve over time if the ontology is set up in a flexible fashion.

**Main scientific contribution** Knowledge collection was not intended as a field of study per se, but has evolved as a practical result supporting the co-evolutionary modelling method and the models created. However, it can be argued that the main novelty in this dimension is the practical implementation of ontologies and Wikis in a modelling process. While neither technology is new and they have seen wide use, their combined application in an ABM is in its infancy.

### 9.1.7 Fact Collection

**Main result** The final dimension of the co-evolutionary method is fact collection. Its main result is the number of encoded instances in the ontology that can directly be used by the simulation engine. Statistics of the ontology, collected at the time of writing, are presented in Table 9.3. In total, the social network has collected 68 instances of different agents, 203 instances of technologies with 278 different operational configurations. The rest of the instance count is accounted for by the data tuples that create the Operational Configurations.

**Issues** Fact collection, while conceptually straightforward, has proven to be a challenging practical issue. Three main issues make fact collection difficult, namely the general unavailability of data, the proprietary nature of existing data and the lack of 'glory' in collecting data.

**Unavailability** The main problem with fact collection is the general unavailability of data. Often, the necessary facts on  $\lambda$ -system agents are simply not collected. In cases where such data is collected, it tends to be highly aggregated and therefore of relatively little use for a generativist, bottom-up approach.

**Proprietary** When data is available on the agents, firms, institutions and technologies, it is regarded as proprietary and a trade secret by actors. It is therefore not accessible to open scientific research, and when it is, one must abide by strict non-disclosure agreements.

**No glory** Finally, an important issue with data collection is that researchers are less likely to work on this level than on the knowledge level. Collecting data is tedious, hard work that brings little recognition.

**Diversity** These problems with data collection lead to a more serious problem, namely the relatively limited diversity of facts, see chapter 8. When the diversity of agents, technologies, Operational Configurations, etc. is low, evolving systems get 'stuck'. The system cannot evolve

Table 9.3: Statistics of the ontology classes, slots and instances

Item	Number
Classes	247
Slots	166
Instances	4312

towards a new state when it lacks the internal diversity needed for the adaptation, until new diversity is created. Generating diversity in the model is not trivial. One can generate artificial diversity by using synthetic agents, but then the model's relevance and realism suffer greatly. Real diversity requires a lot of effort in finding and encoding facts.

**Main scientific contribution** As was the case with knowledge formalization, fact collection was not the focus of this thesis, but rather a result of the modelling method. It has not been scientifically examined. The ontology generated during the method can be seen as a contribution that can be used by others to make different models of the same systems.

**On to conclusions** After having presented and discussed the results of this thesis, the next chapter will draw conclusions about it.



My ideas have undergone a process of emergence by emergency. When they are needed badly enough, they are accepted.

*R. Buckminster Fuller*

**Recapitulation** This thesis opened<sup>1</sup> with a holistic portrait of the Large-Scale Socio - Technical systems ( $\lambda$ -systems) that humans have created within the biological and geological boundaries of planet Earth. The planet is an interconnected, complex adaptive system of geological, biological, social and technical systems. The combined perspectives of multiple formalisms are required if we are to be able to fully understand it. Over the past two hundred years, human activities, being part of the larger  $\lambda$ -systems that have evolved, have begun to influence system Earth to such a degree that a global sustainability crisis is imminent. In order to ensure sustainable human prosperity, we must have a longer time horizon when considering the consequences of our collective actions on the future of these globally interconnected systems, and we must improve our understanding of the evolutionary paths of such  $\lambda$ -systems in order to be better able to choose actions that may bring sustainable development within reach.

Predicting the global effects of our actions is very difficult because of the deeply interlinked nature of the system we are part of and because of the inherent intractability of future developments, i.e., the fundamental inability to exactly predict the effects of an action on an evolving system. One possible solution - the subject of this research - is to collectively create models of the evolution of the complex world around us and to use them in examining possible futures resulting from possible actions. This will provide us with a deeper understanding of system behavior, even if it does not provide us with exact prediction.

This thesis focused on a subsystem of the global socio-technical system, namely regional industrial networks and their evolution. The goal was to create a collaborative, co-evolutionary method for creating models of  $\lambda$ -system evolution. Models created by this method can support decision makers in steering the evolution of such complex systems towards a more sustainable state.

---

<sup>1</sup>A casual reader of this thesis who started reading at this chapter may wish to read chapter 9 as well for context of the conclusions presented here.

**Hypothesis** The main hypothesis of this thesis was that *the use of models of simulated  $\lambda$ -system evolution can improve decision-making about industrial cluster development. By examining patterns of evolution across different simulation scenarios, we can reduce decision makers' uncertainty by answering 'what if' questions about  $\lambda$ -system evolution.*

The hypothesis has not been falsified. The models developed in this thesis are adequate for the problem posed. The models can aid decision makers because they make it possible to examine evolutionary patterns across large parameter spaces and different scenarios.

**Objectives** The objective of the work was *to increase our knowledge of  $\lambda$ -system evolution patterns through simulation of the co-evolution of physical and social networks. The ultimate goal is to provide decision-making support for those involved in shaping the development of industrial clusters.* The objectives were detailed as:

**Gaining insight** *Gain insight in to the social, economic and technological aspects of the co-evolution of  $\lambda$ -systems, and more specifically, of regional industrial systems* This goal has been achieved. The reader interested in domain-specific insights from the case studies should refer to chapter 7.7 for the Costa Due case study results, sections 8.2.6 for the Bulk Biochemicals study, 8.3.6 for the Metals Network study and 8.4.6 for the Bioelectricity study. The overall domain-specific insights are presented in section 9.1.2.

**Creating a method** *To compile models that suitably represent the social and technical realities of industrial networks. These must comply with the laws of conservation of energy and mass and must enable the exploration of the design space of sustainable industrial network evolution.* This goal was achieved with the CostaDue model, presented in chapter 7, and subsequent models. All models are mass balanced and represent agents' decision-making processes in ever-increasing levels of detail and realism. Implementing a closed energy balance has been found to be theoretically possible yet impractical, as will be discussed in Section 9.1.4.

**Supporting decision-makers** *Support decision makers by creating a scientifically sound tool that could be used to support critical actors, notably the RDAs, in decision-making processes regarding regional industrial development.* The generated models, while not formally tested in RDA settings, have generated much interest from the RDAs and have been used to examine the effectiveness of RDAs' strategies (see section 8.2) and to examine alternative policy options (see section 8.4).

**Research questions** The central research question was formulated as: *How can we create a model for exploring the evolutionary patterns of  $\lambda$ -systems?* Three subquestions were derived from the central research question:

**RQ 1** *How can a generativist Complex Adaptive Systems perspective be operationalized in models that capture  $\lambda$ -system evolution?* This question was addressed in chapter 3. Generative science means answering the question "can you grow it?". In order to grow, or generate, emergent phenomena in complex, multiformal  $\lambda$ -systems from interactions between simple, low-level components, the notion of different formalisms was introduced and their use in ontologies presented. Theoretical concepts from the fields of Complex Adaptive Systems and Evolution

Theory were presented as building blocks for a generative approach for creating and understanding emergent properties of  $\lambda$ -systems. This resulted in a unified framework for simulating evolutionary development pathways of  $\lambda$ -systems.

**RQ 2** *What are the content specifications of such models in terms of the relevant formalisms (knowledge domains)?* This question was addressed in chapter 4. The relevant knowledge domains identified, in addition to Complex Adaptive Systems and evolution theory, are theories on industrial clusters and agent-based modelling. Industrial clusters were conceptualized as networks of firms that own technologies, their reasoning behavior described by basic business economic rules and their technical state described with basic process systems engineering input-output mass flow models. Agent-based models are seen as collections of agents, which are defined by their rules and states and create emergent system behavior through their interactions. An extensive literature review demonstrates that all building blocks needed to proceed to RQ 3 were present, and that there is no literature describing previous attempts to synthesize a simulation model of  $\lambda$ -systems evolution from this diverse collection of building blocks.

**RQ 3** *What are the specifications for a method that would create such models?* This question was addressed in chapter 5. This thesis has presented a method for creating models of  $\lambda$ -systems evolution that itself evolves. The method involves the co-evolution between four aspects, namely the technical model design, the social process used to encode knowledge, the knowledge formalization process itself and fact collection. Progress in each dimension deforms the coupled fitness landscape that describes how well the modelling method and the models produced by it perform. Several guiding principles and requirements for this method were defined. Part II of this thesis demonstrated the implementation and outcomes of this evolutionary modelling method.

**Reflection and future work** This ends the conclusions of the thesis. Readers interested in the more informal reflections and future work plans should please see chapter 11.



Now, for the first time in its billions of years of history, our planet is protected by far-seeing sentinels, able to anticipate danger from the distant future - a comet on a collision course, or global warming - and devise schemes for doing something about it. The planet has finally grown its own nervous system: us.

(Dennett, 2003)

## 11.1 Reflections

The words of Daniel C. Dennet are particularly fitting for this final chapter of the thesis. The human species has achieved a level of socio-technical organization unparalleled by any other species on this planet. Science and technology, coupled with the social systems that use them, can indeed be seen as the planet's nervous system. We can gaze far out into space and time and can observe the approach of (self-inflicted) danger. For the first time in human history, we indeed may have the power and means to act in order to avoid approaching doom. How we will use this power is up to us. Are we going to act as deranged and self-destructive mind, too absorbed in our own petty profit and war making, letting the global bio-geo-chemo-socio-technical systems collapse, or are we as a human race going to rise to the challenge and become a conscientious and responsible caretaker of our spaceship Earth?

As a tiny part of the global collective nervous system, this thesis can be seen as striving to increase our understanding and provide practical tools for achieving a more secure and sustainable future. Reflecting back on the thesis, the following line of argumentation was presented:

- Work started with a realization that human society is an evolving, complex  $\lambda$ -system embedded in a global biogeochemical system, and that its evolution needs to be steered if we are to avoid a global sustainability crisis.
- We examined the scientific and modelling foundations of Complex Adaptive Systems and realized that in using traditional modelling approaches we miss vital social and adaptive components. These limitations hamper our ability to understand and shape the evolution of  $\lambda$ -systems.

- We broadened the traditional scientific and modelling basis with a socially constructed knowledge perspective and proposed a new approach for creating models of  $\lambda$ -system evolution that share the dynamics and structure with real-world systems.
- By concentrating on the multiformal nature of models that to be created, we developed a social interface (the SDM). This social interface facilitates the integration of the social process of knowledge formalization with the scientific modelling process.
- We observed that if the outcomes of such integration are to be complex models of Complex Adaptive Systems, they must be created through a evolutionary process. This observation led to the insight that the modelling method, and science itself, is an evolutionary process.

In the remainder of this chapter we will first reflect on the scientific aspects of this work, then on the models created in this thesis, and finally on the co-evolutionary modelling method that made them. We will close this work with thoughts and plans for future work.

### 11.1.1 Scientific Contribution

Starting the reflection, we will consider the place of this work in the grander scheme of science. This work fits within the notions of exploratory modelling of multi-scale problems and may potentially lead to the founding of Complex Adaptive Systems Engineering. But first, we need to position this work within that which has already been done on post-normal science.

**Post-normal science** Traditionally, through modelling one attempts to answer a particular domain-based question. The process starts with the collection of relevant data, proceeds with model construction and ends with the interpretation of the model's outcomes and conclusions about the problem. The modelling method presented in this thesis changes this by starting with the identification of the stakeholders' knowledge problem and proceeds with a social knowledge collection process and the formalization of this knowledge. Then the actual question is identified by the stakeholder, given the formalized domain, and the model for answering it is constructed. Model analysis and feedback are again explicitly social processes. This modelling method facilitates the creation of a knowledge community, rather than simply answering the question posed.

Organizing the modelling process in this way means creating a coherent, focused and integrative research path with explicit, socially constructed multiformal knowledge. It also means acknowledging that there is no single correct way to understand the world. It is in fact post-normal science, as defined by [Funtowicz and Ravetz \(1993\)](#), where gaps exist in knowledge and understanding that cannot be resolved using the traditional scientific paradigm but which require the incorporation of multiple - and sometimes contradictory - viewpoints into the same problem-solving process ([Kay et al., 1999](#)).

**Exploratory modelling** The traditional Baconian approach to modelling poses the question of whether a sufficiently correct model can be created to describe the evolution of the system under scrutiny. In a post-normal world view, this question cannot be answered, as 'correct' is ill defined, especially when social aspects are involved. Being aware of this limitation, our approach examines whether we can gather a large enough diversity of perspectives on the system elements behavior and, when this is accomplished, sets out to explore possible futures.

Is this useful, then, if it cannot generate the 'correct' model? Indeed it is, as it reflects back to stakeholders the consequences of their beliefs, while exploring vast model outcome spaces. It is up to them to recognize patterns in the multitude of evolutionary paths and potential futures - and to learn from them.

**Multi-scale problems** This exploratory modelling approach is naturally extended to the notion of multi-scale problems. One of the major sources of complexity in systems is the intertwining of different levels of scale in the problem. The generativist perspective clearly points toward understanding systems from the bottom up, across many scales of aggregation and interaction. Theoretically, there are no limitations to the multi-scale models. Practical limitations concern the availability of knowledge, model mechanisms and data as well as the computational resources.

### 11.1.2 Models Created

This thesis presented seven different models, each model emerging from the previous one. They have provided a wealth of insights on  $\lambda$ -system evolution and were shown to be useful. But what are the strengths and weaknesses, the limitations and the possibilities of those models?

**What the models *can* do** The main thing that Agent Based Models of  $\lambda$ -system evolution can do is to help us explore and provide a sense of the range of possible system futures. By examining the models across a wide range of parameter values, a map or pattern of possible future development can be seen, depending on, for example, market changes, the introduction of new technologies, new policy interventions or changes in management styles.

As discussed in Chapter 1, humans are relatively weak at systematic reasoning across a myriad of interactions between system elements, but excel at pattern recognition. Computers, and computer models are, as Steve Jobs has observed, bicycles of the mind <sup>1</sup>, systematically exercising the relations between system elements, creating complex behavioral patterns. It is up to us to interpret those patterns.

The created models are built on objective physical and behavioral characteristics of physical networks and on the beliefs and assumptions of stakeholders about the properties and mechanisms of the social network. Using the created ABMs of  $\lambda$ -system evolution provides us with a formal mechanism for testing the understanding and the intuition of the stakeholders. The ABMs often surprise us by showing emergent properties - logical but nonetheless surprising - caused by unpredicted interactions between agents.

**What the models *cannot* do** Compared to what the models can do, there are of course infinitely more things they cannot do. There are two main limitations relevant to understanding evolving  $\lambda$ -system evolution.

Because of the thesis focus, all models produced in this work are unable to answer questions at the tactical or operational level. The models are not meant to aid the decision-making process on selecting suppliers, shutting down or starting a production of a single plant, etc. The models are meant to support strategic thinking only. This limitation, however, applies to

---

<sup>1</sup>[http://www.youtube.com/watch?v=PUagMQZ\\_WFQ](http://www.youtube.com/watch?v=PUagMQZ_WFQ)

the models developed. It is relatively straightforward to create the next generation of models that will be able to support answering operational and tactical questions.

On a more fundamental level, there is a theoretical limit to what any generative model of an evolving system can do. As extensively discussed in this thesis, evolution is intractable. That means that there can never be a model that exactly predicts the future state of an evolving system. We might find models that provide *good enough* predictions, for varying levels of that qualification, given certain resources.

**Model advantages** There are a number of advantages to using ABMs in the manner presented in this thesis. ABMs' main strengths lie in the fact that they form a very straightforward representation of the system they are describing. ABMs consists of discrete entities that interact in parallel and contain multiple formalisms embedded in them. This strong structural correspondence leads to an intuitive understanding of models by their users. Our experience in interacting with stakeholders, the users and modelers taught us that the behavior of agents and their interaction is very easily understood, as we humans tend to think about entities having goals and 'wanting' things. ABMs offer a new modelling paradigm. They are in general seen as something new and exciting. Modelers and users alike are drawn to "shiny new toys". ABMs allow for relatively easy collective modelling. Compared to more top-down models, the distributed nature of collective modelling makes it relatively easy to split up the creation of a bottom-up model among different people.

**Models' disadvantages** There are two main disadvantages that lie within the technique of ABM itself, and three social disadvantages, stemming from models (mis)use. Given the inherent structure of ABMs, they are have a rather high data requirement. As the degree of realism of models increases, the data requirement rises, quickly becoming impractical. This is mainly caused by the large increase in the number and detail of individual entities that need to be described. Furthermore, implementation of a ABM is still not straightforward. Even though there are many efforts in making ABM more user friendly, the fact that actual behavioral algorithms need to be encoded requires computer programming skills for all but the simplest of agent descriptions.

Social aspects of model use lead to three main disadvantages of ABM. First, modelling has in general been oversold to policy makers. System Dynamics, Operations Research, etc. have been actively promoted by modelers to policy makers with the promise of a rational prediction of future system states. Often, these predictions have turned out to be erroneous, raising doubts about the usefulness of modelling in general. Users have to be educated about the scope and relevance of Agent Based Model, and the modelers need to be more careful about their claims of prediction. Somewhat contradictory to the first point, in cases where policy makers have accepted a model and are comfortable with its use, they tend to use it in ways not meant by modelers, leading to erroneous conclusions. This overuse of and overdependency on models is harmful both to policy and model use. Finally, because of the relative novelty of Agent Based Model and the prevalence of the traditional top-down, command and control paradigm, users seem to not fully understand the bottom-up, distributed approach to modelling, and tend to be distrustful of notions of emergence and self-organization that drive ABMs.

### 11.1.3 Modelling Method

The main insight gained from the modelling method is that scientific research is a highly chaotic evolutionary process during which we must make errors. This process is never completed, achieving at best 'good enough' results. This insight will be elaborated below.

**Path dependency and making errors** The modelling method is explicitly co-evolutionary. Evolution is a process of making mistakes. When actively evolving a socio-technical system, either at the scale of models or at the scale of  $\lambda$ -systems, mistakes will inevitably be made. For example, the relatively elaborate technical model design created in the Chocolate Game case study (see 6.4) used an inappropriate conceptualization; it modelled the flows between processes as discrete entities, whereas in reality they are continuous. Such mistakes are useful, either as lessons in how not to do things or as unexpected improvements. In this example, the next model created during the CostaDue case rectified the error discovered in the Chocolate Game. The discrete flow model proved to be useful much later when constructing a model of a multi-modal freight hub (Sirikijpanichkul et al., 2007). Success or failure is arguably determined by a system's ability to gracefully deal with mistakes, whether through the ability to learn from them or through the ability to undo them. The modelling method requirements of modularity, versioning and authorship (see section 5.4 directly contribute to this aspect.

From a co-evolutionary perspective, recording all generated data/knowledge becomes imperative. We are bound to repeat previous mistakes, and without a good historic record we would not be able to recognize them and rectify them in time.

**Chaotic progress** Scientific progress is strongly influenced by serendipity and chance. Chance encounters at a conference, mixed up bottles and subtle errors in computer code can lead to unexpected but novel insights. The future direction of scientific progress cannot be predicted in detail, as one does not know how some chance effect in the future will influence other events. It is chaotic and intractable, just like any evolutionary process. As long as it remains testable, falsifiable and reproducible, such chaotic processes can be usefully harnessed, as demonstrated in this thesis. Thus, as already discussed, the presented co-evolutionary modelling method is not about reinventing the scientific wheel. It is about making visible the spokes, sprockets and chain that drive it, and allowing us to explicitly observe the intractable, evolutionary nature of science and engineering design.

**Never finished** Science, just as evolution, is never finished. Scientific results are never perfect, as the coupled fitness landscape of human understanding of the world we live in changes and deforms continually, raising the bar continuously. Instead of aiming for a grand goal by saying "it will be great once it is done", evolution teaches us that gradual progress and continuous improvement through local optimization is the way to go. Speaking in software terms, evolving processes are in a "perpetual beta" state, always incomplete but useful as is.

**Good enough** The notion of chaotic and never finished progress leads us to rethink the general approach to  $\lambda$ -system "design". Instead of aiming for a perfect solution, we must realize that the most we can achieve is an adequate, or good enough state, as both the problem formulation and the problem solution change all the time. The notion of good enough requires that we consider what would be an acceptable minimum threshold of a system's performance

and allow the system to settle into any state that is above that. This way we allow the system's internal dynamics to run its course, and the unwanted side effects that occur when a system is far removed from its attractor are reduced.

**What the method *can and cannot* do** Summarizing the above discussion, the co-evolutionary modelling method as presented in this thesis can integrate many different formalisms into a single coherent vision on  $\lambda$ -system evolution. It can involve many different participants in collaborating on a greater whole, greatly increasing the overall output of the method. The method, being a Complex Adaptive System itself, can display emergent, surprising outcomes. If the method requirements are followed carefully, these emergent outcomes will be useful and insightful.

On the other side, the method cannot be used as a top-down steering mechanism, as it is by design bottom-up and distributed. Also, the co-evolutionary method can not guarantee any particular outcome, due to its chaotic nature, and finally, the co-evolutionary modelling method is not something one can perform alone.

**When should the method *be or not be* used?** When should the method be used? The method is suitable in supporting long-term multidisciplinary research with a coherent theme. It is useful when the modelers wish to build strong relationships with stakeholders, and the problem addressed is complex, multi-actor, multi-perspective and multi-scale.

The presented model should not be used to create 'quick and dirty', one off models, nor should it be used when it is important to urgently answer a new, burning question. The method is a long-term, multi-person effort that does not downscale well. It is also not suitable for rapid model prototyping and testing of ideas, unless they are evolutionary off-shoots of a larger modelling method.

**Advantages of the method** As was done with the model outcomes, we will discuss the main advantages of using the co-evolutionary modeling method to create models of  $\lambda$ -systems evolution. The main advantages are that through soliciting a lot of stakeholder interaction in the model creation process, the method and its outcomes are likely to be accepted by the stakeholders. The method is adaptive. Since it is co-evolutionary, the method can adapt to changes in stakeholder preferences, new insights, etc. Furthermore, by focusing on a continuous improvement of the existing models, the method is suitable to capacity building and deepening of existing insights by research groups. Added to that, once the method has been established as a standard working protocol of a research group, it becomes a proverbial model machine, producing new outcomes very quickly and effectively. Finally, the modelling method is explicitly meant to integrate different types of knowledge from different people into one multiformal, n-dimensional whole.

**Disadvantages of the method** The main practical disadvantage is that the method is relatively expensive. Performing the method requires many different people and lots of their time. Furthermore, the method has a relatively slow takeoff in its first step. It requires a lot of up-front time and effort investment before yielding results. Creating the social network, understanding the real problem the stakeholder has, creating ontologies, setting up the simulation engine, etc. are time consuming. The social network needed to operate the method is also a

possible disadvantage, as the method is highly dependent on the quality of the social network and individual contributions by its members. This makes it relatively fragile to changes in the composition and performance of the social network. If key individuals leave, or if the cooperative atmosphere is disturbed, the method can fail. Furthermore, the method is relatively new and is often seen by stakeholders as being far removed from their daily work practice. It can be seen as experimental and too complicated. It takes time for them to become accustomed to the required way of working. Finally, inherent to the co-evolutionary nature of the method, it suffers from path dependency or lock-in. The method development is heavily dependent on its history. If the requirements of the stakeholders change, a new investment must be made to restart the method.

## 11.2 Outlook and Future Work

Just as any other evolutionary process, the modelling method presented in this thesis does not just stop. In this last section of the thesis, we will first reflect on the short-term practical future work ahead of us and then reflect on the more general outlook for the work that has been started.

**Short-term future work** The simulation engine presented in this thesis has served its purpose well. Yet ongoing demand for more complex, multiformal and larger models has identified several areas in which the current design is lacking. There are four issues we need to deal with in the near future, namely scalability, conceptual complexity, ease of model use and community diversity.

**Scalability** Due to the design of Repast, the agent framework, a single simulation cannot scale across multiple processor machines. The fact that the length of a simulation run increases exponentially with the number of agents simulated will limit the maximum size of clusters that can be examined. A redesign of the software stack is therefore needed in order to create a distributed simulation engine.

**Conceptual Complexity** In addition to the limitations in computation capacity, we also face challenges in organizing the conceptual complexity of agents. As more and more formalisms are added to an agent's description, cross cutting concerns begin to limit our ability to create algorithms that are understandable and maintainable. An example of a cross cutting concern is risk assessment, since there is the need to be able to modify the behavior of agents at multiple places: during purchasing, during price determination, during decisions on temporary shutdown, etc. A very promising development in software programming called Aspect Oriented Programming will allow us much greater diversity in agent formalisms.

**Ease of use** As we move towards solving the distributed computation and conceptual complexity problem, the ease of using the simulation engine is likely to deteriorate. We need to make an effort to increase the ease of creating agent behaviors, as this will allow for greater stakeholder participation, reducing the need for skilled programmers.

**Community diversity** The learning cases, while providing useful modelling insights, highlighted the problem of too little community diversity. The main question is how to engage

a much larger and diverse stakeholder pool, so that both acceptability and diversity of formalisms can be increased. This is not necessarily a scientific problem, but a social one.

**Long-term future work** In addition to the short-term activities, we have identified several longer-term directions for future work. These are a thorough analysis of the modelling method in all four dimensions, a link with serious gaming and the addition of several new formalisms.

**Modelling method analysis** There is a wealth of information buried in the wiki/svn collaboration data, the structure of the ontology, etc. We believe a social scientist can extract valuable information on the functioning of the social network and provide further insights into refining the aspects of the co-evolutionary method that were not explored in this thesis. Some work on this has already begun <sup>2</sup>.

**Serious gaming** Linking Agent Based Model with serious gaming is a promising direction for increasing the ease of model use and increasing the diversity of agents' behavioral models. The idea is to use real life players that play a serious game as 'programmers' of ABM. Human players are very sophisticated and rich in behavior. However, they can only play a very limited number of games, making parameter sweeps across model/game parameters impossible. ABMs can replay the game/simulation essentially infinitely (millions of times if necessary), and a parameter sweep is then very simple. However, the behavior of agents in ABMs is much more limited than that of humans. Ideally, in combining the power of both we will be able to develop a model of a human player, and then examine its response over a wide parameter space.

**Other formalisms** There are several formalisms not encoded so far in which stakeholders have expressed an interest and which pose an interesting scientific challenge.

**Risk perception** The formalism most in demand by stakeholders is risk perception by agents. Risk is a basic element of all decision-making processes in firms. The formalization of risk has begun with the writing of this thesis <sup>3</sup>.

**Strategic agent behavior** The second most requested formalism is that of strategic agent behavior, in the form of management strategies, cheating and deception. In the current situation, for example, an agent always delivers a contract, while in some cases it might be beneficial for it not do so. This is a complex formalism that will require nontrivial conceptualizations and software programming.

**Hybrid modelling** As the case studies become more realistic, the need arises for a high quality and flexible description of agents' environments. Most scenarios are defined in terms of oil price, global interest rate, etc. Well-tested methods for modelling top-down, large-scale system behavior are tools like system dynamics and general equilibrium models. The challenge is to combine these standard tools with ABM and not just to use them in parallel, but to actually integrate the static structure of the top-down aggregate model with individual agent behavior. This poses a wealth of scientific challenges. Initial work as already been started <sup>4</sup> on integrating CGE models of industrial regions and ABMs.

---

<sup>2</sup><http://wiki.tudelft.nl/Project/SimulationOntologyEvolution>

<sup>3</sup><http://wiki.tudelft.nl/Research/TheosNotes>

<sup>4</sup><http://wiki.tudelft.nl/Research/IntegratingTheWorldTradeModelWithAgentBasedModel>

**Design Processes** Finally, the product or process design process itself can be examined with the use of an ABM. Work has already started <sup>5</sup> on describing the design process and its interaction with the products failure modes.

**Complex Adaptive Systems Engineering** A socially constructed, exploratory and multi-scale understanding of the process of the coming into being of system structures naturally leads to the notion of Complex Adaptive Systems Engineering (CASE). This is the deliberate act of creating an evolving complex adaptive system in order to reach a certain goal. The idea has already generated some interest in the literature (Clymer, 1999; Kroes, 2009; Sage, 2001).

Since the basic nature of CASE is that it is complex, many traditional design and engineering approaches are not valid. We face the following challenges:

- Complex Adaptive Systems cannot be designed at once, but have to evolve. Traditional planning becomes impossible.
- We cannot fully specify the exact outcome requirements of a system under CASE; we can only limit its outcome space, hoping to reach a 'good enough' state. In essence, it amounts to engineering emergence.
- Complex Adaptive Systems are inefficient but robust; CASE requires a totally new mindset among designers.

Engineering systems in this manner can only be done by a generative, bottom-up, socially inclusive and complex process, and this thesis is a first step towards exactly that.

---

<sup>5</sup><http://wiki.tudelft.nl/Project/FailureModeAvoidance>



- W. Achterberg. *Samenleving, natuur en duurzaamheid: een inleiding in de milieufilosofie*. Gorcum b.v., Konink. Van, 1994.
- D. Adams. *The Hitchhiker's Guide to the Galaxy: A Trilogy in Four Parts*. London: Picador, 2002.
- R.K Ahuja, T.L. Magnati, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.
- V. Albino, N. Carbonara, and I. Giannoccaro. Innovation in industrial districts: An agent-based simulation model. *International Journal of Production Economics*, 104(1):30–45, 2006.
- A. Aldea, R. Banares-Alcantara, L. Jimenez, A. Moreno, J. Martinez, and D. Riano. The scope of application of multi-agent systems in the process industry: three case studies. *Expert Syst. Appl.*, 26(1):39–47, 2004.
- F. Alkemade, H. La Poutre, and H. M. Amman. On social learning and robust evolutionary algorithm design in the Cournot oligopoly game. *Comput. Intell.*, 23(2):162–175, 2007.
- B. Allenby. The ontologies of industrial ecology? *Progress in Industrial Ecology, an International Journal*, 3(1):28–40, 2006.
- B. R. Allenby. *Industrial Ecology: Policy framework and implementation*. Prentice-Hall Inc., 1st edition, 1999.
- L. Alpay, A. Giboin, and R. Dieng. Accidentology: An example of problem solving by multiple agents with multiple representations. In Maarten W Van Someren, Peter Reimann, Henny P A Boshuizen, and Ton De Jong, editors, *Learning with multiple representations*, Advances in learning and instruction series, pages 152–174. Elsevier, Oxford, UK, 1998.
- D.F. Andersen and G.P. Richardson. Scripts for group model building. *System Dynamics Review*, 13(2):107–129, 1997.
- D.F. Andersen, G.P. Richardson, and J.A.M. Vennix. Group model building: adding more science to the craft. *System Dynamics Review*, 13(2):187–201, 1997.

- M. H. Anderson, J. R. Ensher, M. R. Matthews, C.E. Wieman, and E. A. Cornell. Observation of bose-einstein condensation in a dilute atomic vapor. *Science*, 269(5221):198–201, 1995.
- L. Andrasik and J. Krempasky. Emergence of deterministic chaos: Limited passage in the evolution of economy. *Ekonom. Cas.*, 50(6):1076–1099, 2002.
- C.A. Andrews, A. Baptista, and S. Patton. A multi-agent model of a small firm. Technical report, Rutgers University, <http://radburn.rutgers.edu/andrews/projects/abm/EPAppr4-6-04.pdf>, 2005.
- K. Antelman. Do open-access articles have a greater research impact. *College & Research Libraries*, 65(5):372–382, 2004.
- A.M.D. Approach. Web-Based collaboration and Decision Making Support. *Int. J. of Web-Based Learning and Teaching Technologies*, 2(4):12–23, 2007.
- C. Argyris and D. A. Schon. *Organisational learning II; theory, method and practice*. Amsterdam, Addison-Wesley, 1996.
- H.r. Arkes and C. Blumer. The psychology of sunk cost. *Organizational behavior and human performance*, 35(1):124–140, 1985.
- W.R. Ashby. Variety, Constraint, and the Law of Requisite Variety. *Modern Systems Research for the Behavioral Scientist*. Hawthorne, NY: Adline de Gruyter, 1968.
- I. Asimov. *Prelude to Foundation*. Bantam Books, 1989.
- R.M. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.
- F.J. Ayala and J.A. Kiger. *Modern Genetics*. Benjamin-Cummings Publishing Company, 1980.
- E. Baark. Technological entrepreneurship and commercialization of research results in the west and in china - comparative perspectives. *Technol. Anal. Strateg. Manage.*, 6(2):203–214, 1994.
- S. Bandini, S. Manzoni, and G. Vizzari. Crowd modeling and simulation - The role of multi-agent simulation in design support systems. *Innovations In Design & Decision Support Systems In Architecture Urban Planning*, pages 105–120, 2006.
- S. Banks, R. Lempert, and S. Popper. Making computational social science effective - Epistemology, methodology, and technology. *Soc. Sci. Comput. Rev.*, 20(4):377–388, 2002.
- Y. Bar-Yam. *Dynamics of Complex Systems*. Westview Press, 2003.
- Y. Bar-Yam. Concepts in complex systems: Systems perspective. [http://necsi.edu/guide/concepts/system\\_perspective.html](http://necsi.edu/guide/concepts/system_perspective.html), 2008.
- A.L Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999. URL <http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9910332>.
- A.L. Barabási, E. Ravasz, and T. Vicsek. Deterministic scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 299(3-4):559–564, 2001.

- R. Barbault and S. Sastrapradja. Generation, maintenance and loss of biodiversity. *Global Biodiversity Assessment. United Nations Environment Programme, Cambridge, USA*, 1995.
- O. Barreteau. The joint use of role-playing games and models regarding negotiation processes: characterization of associations. *Jasss-the J. Artificial Societies Social Simulation*, 6(2), 2003.
- O. Barreteau, F. Bousquet, and J.M. Attonaty. Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to Senegal River Valley irrigated systems. *Jasss-the J. Artificial Societies Social Simulation*, 4(2):U75–U93, 2001.
- O. Barreteau, F. Bousquet, C. Millier, and J. Weber. Suitability of Multi-Agent Simulations to study irrigated system viability: application to case studies in the Senegal River valley. *Agric. Syst.*, 80(3):255–275, 2004.
- P. Bartelmus. Dematerialization and capital maintenance: two sides of the sustainability coin. *Ecological Economics*, 46(1):61–81, 2003.
- T. Basar and GJ Olsder. Team-optimal closed-loop Stackelberg strategies in hierarchical control problems. *Automatica*, 16(4):409–414, 1980.
- N. Baxter, D. Collings, and I. Adjali. Agent-based modelling - intelligent customer relationship management. *BT Technol. J.*, 21(2):126–132, 2003.
- N. Becu, F. Bousquet, O. Barreteau, P. Perez, and A. Walker. A methodology for eliciting and modelling stakeholders' representations with agent based modelling. *Lect. Note. Artif. Intell.*, 2927:131–148, 2003.
- P.J. Beers and P.W.G. Bots. Eliciting conceptual models to support interdisciplinary research. In Ralph H Sprague, editor, *40th Hawaii International Conference on System Sciences*, Waikoloa, HI, 2007.
- P.J. Beers, H.P.A. Boshuizen, P.A. Kirschner, and W. Gijsselaers. Common ground, complex problems and decision making. *Group Decision and Negotiation*, 15:529–556, 2006.
- P.J. Beers, H.P.A. Boshuizen, P.A. Kirschner, and W.H. Gijsselaers. The analysis of negotiation of common ground in CSCL. *Learning and Instruction*, 17(4):427–435, 2007.
- P.J. Beers, I. Nikolic, P. Bots, and G.P.J. Dijkema. Facilitating the modelling of complex adaptive systems. *JASSS*, 2009. Forthcoming.
- J. Bentahar, B. Moulin, and J.J.C. Meyer. A tableau method for verifying dialogue game protocols for agent communication. *Lect. Note. Artif. Intell.*, 3904:223–244, 2006.
- T. Berger. Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis. *Agric. Econ.*, 25(2-3):245–260, 2001.
- T. Berger and P. Schreinemachers. Creating agents and landscapes for multiagent systems from random samples. *Ecol. Soc.*, 11(2), 2006.
- T. Berger, R. Birner, J. Diaz, N. McCarthy, and H. Wittmer. Capturing the complexity of water uses and water users within a multi-agent framework. *Water Resour. Manag.*, 21(1): 129–148, 2007.

- M. Bergholtz, P. Jayaweera, P. Johannesson, and P. Wohed. Reconciling physical, communicative, and social/institutional domains in agent oriented information systems - A unified framework. *Lect. Note. Comput. Sci.*, 2814:180–194, 2003a.
- M. Bergholtz, P. Jayaweera, P. Johannesson, and P. Wohed. Modelling institutional, communicative and physical domains in agent oriented information systems. *Lect. Note. Comput. Sci.*, 3030:189–205, 2003b.
- L. v. Bertalanffy. *General System Theory; Foundations, Development, Applications*. New York, Braziller., 1968.
- C. S. Bertuglia and F. Vaio. *Nonlinearity, chaos, and complexity the dynamics of natural and social systems*. Oxford, Oxford University Press., 2005.
- S. Bharwani. Understanding complex behavior and decision making using ethnographic Knowledge Elicitation Tools (knets). *Social Science Computer Rev.*, 24(1):78–105, 2006.
- S. Bharwani, M. Bithell, T. E. Downing, M. New, R. Washington, and G. Ziervogel. Multi-agent modelling of climate outlooks and food security on a community garden scheme in Limpopo, South africa. *Philosophical Transactions Royal Soc. B-biological Sciences*, 360(1463):2183–2194, 2005.
- W.E. Bijker, T.P. Hughes, and T.J. Pinch. *The Social construction of technological systems : new directions in the sociology and history of technology*. MIT Press. Cambridge, Mass, 1987.
- M. Binbasioglu and E. Winston. Knowledge-based management support: an application to the IS change agent role problem. *Expert Syst. Appl.*, 25(4):517–532, 2003.
- C.R. Binder. From material flow analysis to material flow management Part II: the role of structural agent analysis. *J. Clean Prod.*, 15(17):1605–1617, 2007.
- B.S. Blanchard and W.J. Fabrycky. *Systems engineering and analysis*. Prentice Hall, 1998.
- S. Blokker. Coevolution in the process industry: An agent based model for the Eemshaven chemical cluster. Master’s thesis, TU Delft, Faculty of Chemical Engineering, 2006.
- IEEE Standards Board. *IEEE Standard for Software Unit Testing: An American National Standard, ANSI/IEEE Std 1008-1987*. The Institute of Electrical and Electronics Engineers, Inc. Software Engineering Technical Committee of the IEEE Computer Society., 1999.
- O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(90001):267–270, 2004.
- C.A. Boer, A. Verbraeck, and H.P.M. Veeke. Distributed Simulation of Complex Systems: Application in Container Handling. In *Proceedings of SISO European Simulation Interoperability Workshop*, pages 24–27, 2002.
- R. Boero and F. Squazzoni. Does empirical embeddedness matter? Methodological issues on agent-based models for analytical social science. *Jasss-the J. Artificial Societies Social Simulation*, 8(4), 2005.

- R. Boero, M. Castellani, and F. Squazzoni. Micro behavioural attitudes and macro technological adaptation in industrial districts: an agent-based prototype. *JASSS*, 7(2), 2004.
- M. Boisot, I.C. MacMillan, and K.S. Han. Property rights and information flows: a simulation approach. *J. Evol. Econ.*, 17(1):63–93, 2007.
- E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(90003):7280–7287, 2002.
- E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA, 1999.
- Z. Bonen. Evolutionary behavior of complex sociotechnical systems. *Research Policy*, 10(1): 26–44, 1981.
- G. Booch, J. Rumbaugh, and I. Jacobson. *The unified modeling language user guide*. Addison-Wesley Reading Mass, 1999.
- A. Borshchev and A. Filippov. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, pages 25–29, 2004.
- H.P.A. Boshuizen and Hermina J.M. Tabachneck-Schijf. Problem solving with multiple representations by multiple and single agents: An analysis of the issues involved. In M W Van Someren, Peter Reimann, Henny P A Boshuizen, and Ton De Jong, editors, *Learning with multiple representations*, pages 137–151. Elsevier, Oxford, UK, 1998.
- N. Bostrom. Are You Living in a Computer Simulation? *Philosophical Quarterly*, 53(211): 243–255, 2003.
- F. Bousquet, J.C. Castella, G. Trebuil, C. Barnaud, S. Boissau, and S.P. Kam. Using multi-agent systems in a companion modelling approach for agro-ecosystem management in South-east asia. *Outlook On Agriculture*, 36(1):57–62, 2007.
- G.E.P. Box. Some problems of statistics and everyday life. *Journal of the American Statistical Association*, 74(365):1–4, 1979.
- S. Boyson, T. Corsi, and A. Verbraeck. The e-supply chain portal: a core business model. *Transportation Research Part E*, 39(2):175–192, 2003.
- L. J. Bracken and E. A. Oughton. 'what do you mean?' The importance of language in developing interdisciplinary research. *Transactions Institute British Geographers*, 31(3):371–382, 2006.
- R.A. Brealey, S.C. Myers, and A.J. Marcus. *Fundamentals of Corporate Finance*. McGraw-Hill Irwin, 2004.
- P. Bresciani and P. Donzelli. A practical agent-based approach to requirements engineering for socio-technical systems. In *Lecture Notes In Computer Science*, volume 3030, pages 158–173, 2004.

- R. Bromme. Beyond one's own perspective: The psychology of cognitive interdisciplinarity. In P Weingart and N Stehr, editors, *Practicing interdisciplinarity*, pages 115–133. University of Toronto Press, Toronto, Canada, 2000.
- R. Bromme, R. Rambow, and M. Nückles. Expertise and estimating what other people know: The influence of professional experience and type of knowledge. *Journal of Experimental Psychology: Applied*, 7(4):317–330, 2001.
- G. Brundtland. World commission on environment and development, our common future. Technical report, WCED, 1987.
- J.J. Bryson, Y. Ando, and H. Lehmann. Agent-based modelling as scientific method: a case study analysing primate social behaviour. *Philos. Trans. R. Soc. B-Biol. Sci.*, 362(1485): 1685–1698, 2007.
- L.L. Bucciarelli. Reflective practice in engineering design. *Design Studies*, 5(3):185–190, 1984.
- M. Buchanan. *Ubiquity: The Science of History or Why the World is Simpler than we Think*. London, Weidenfeld, 2000.
- S Buckingham Shum. Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In *AAAI Sprins Symposium on Artificial Intelligence in Knowledge Management*, Palo Alto, USA, 1997. AAAI Press.
- D.S. Callaway, M.E.J. Newman, S.H. Strogatz, and D.J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85:5468, 2000. URL <http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0007300>.
- J. Calmet, A. Daemi, R. Endsuleit, and T. Mie. A liberal approach to openness in societies of agents. *Lect. Note. Comput. Sci.*, 3071:81–92, 2003.
- A.D. Calvin, F.K. Hoffman, and E.L. Harden. The effect of intelligence and social atmosphere on group problem solving behavior. *Journal of Social Psychology*, 45:61–74, 1957.
- N.A. Campbell. *Biology*. Benjamin Cummings, 2002.
- A.M.C. Campos and D.R.C. Hill. An agent-based framework for visual-interactive ecosystem simulations. *Trans. Soc. Comput. Simul.*, 15(4):139–152, 1998.
- K.M. Carley. Artificial intelligence within sociology. *Sociol. Methods. Res.*, 25(1):3–30, 1996.
- K.M. Carley, D.B. Fridsma, E. Casman, A. Yahja, N. Altman, L.C. Chen, B. Kaminsky, and D. Nave. Biowar: Scalable agent-based model of bioattacks. *IEEE Trans. Syst. Man Cybern. Paart A-Syst. Hum.*, 36(2):252–265, 2006.
- S.R. Carpenter and W.A. Brock. Spatial complexity, resilience, and policy diversity: Fishing on lake-rich landscapes. *Ecol. Soc.*, 9(1), 2004.
- J. Carrillo-Hermosilla. A policy approach to the environmental impacts of technological lock-in. *Ecol. Econ.*, 58(4):717–742, 2006.

- R. Carson. *Silent Spring*. Houghton Mifflin, 1962.
- J. C. Castella and P. H. Verburg. Combination of process-oriented and pattern-oriented models of land-use change in a mountain area of vietnam. *Ecological Modelling*, 202(3-4):410–420, 2007.
- J. C. Castella, T. N. Trung, and S. Boissau. Participatory simulation of land-use changes in the northern mountains of Vietnam: the combined use of an agent-based model, a role-playing game, and a geographic information system. *Ecol. Soc.*, 10(1), 2005.
- J.C. Castella, S.P. Kam, D.D. Quang, P.H. Verburg, and C.T. Hoanh. Combining top-down and, bottom-up modelling approaches of land use/cover change to support public policies: Application to sustainable management of natural resources in northern vietnam. *Land Use Policy*, 24:531–545, 2007.
- M. Castells. *The Rise of The Network Society*. Blackwell Publishing, 2000.
- B. Chaib-draa. Causal maps: Theory, implementation, and practical applications in multiagent environments. *IEEE Trans. Knowl. Data Eng.*, 14(6):1201–1217, 2002.
- B. Chaib-draa. Causal reasoning in multi-agent systems. *Lect. Note. Artif. Intell.*, 1237:79–97, 1997.
- D. Challet, M. Marsili, and Y.C. Zhang. Stylized facts of financial markets and market crashes in Minority Games. *Physica A: Statistical Mechanics and its Applications*, 294(3-4):514–524, 2001.
- E.J.L. Chappin. Carbon dioxide emission trade impact on power generation portfolio, agent-based modelling to elucidate influences of emission trading on investments in dutch electricity generation. Master’s thesis, Delft University of Technology, 2006.
- E.J.L. Chappin and G.P.J. Dijkema. On the impact of CO<sub>2</sub> emission-trading on power generation emissions. *Technological Forecasting & Social Change*, 76(3):358–370, March 2009. URL <http://dx.doi.org/10.1016/j.techfore.2008.08.004>.
- K. Chatterjee and S.H. Xu. Technology diffusion by learning from neighbours. *Adv. Appl. Probab.*, 36(2):355–376, 2004.
- V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, and J.P. Rice. Okbc: a programmatic foundation for knowledge base interoperability. In *AAAI ’98/IAAI ’98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 600–607, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence. ISBN 0-262-51098-7.
- L. Cheng, E. Subrahmanian, and A. W. Westerberg. Design and planning under uncertainty: issues on problem formulation and solution. *Computers & Chem. Engineering*, 27(6):781–801, 2003.
- C. Choe and I. Fraser. A note on imperfect monitoring of agri-environmental policy. *J. Agric. Econ.*, 49(2):250–258, 1998.

- H.H. Clark and E.F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.
- A. Clauset, C.R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data, 2007. URL <http://www.citebase.org/abstract?id=oai:arXiv.org:0706.1062>.
- W. F. Clocksin. Artificial intelligence and the future. *Philos. Trans. R. Soc. Lond. Ser. A-Math. Phys. Eng. Sci.*, 361(1809):1721–1748, 2003.
- J.R. Clymer. Simulation-Based Engineering Of Complex Adaptive Systems. *Simulation*, 72(4):250, 1999.
- M.D. Cohen, J.G. March, and J.P. Olsen. Garbage can model of organizational choice. *Administrative Science Quarterly*, 17(1):1–25, 1972.
- J. Conklin and M.L Begeman. gibis: A hypertext tool for team design deliberation. In *Hypertext'87*, pages 247–251, Chapel Hill, US, 1987. ACM Press.
- SUN Corporation. *The source of JAVA technology*. <http://java.sun.com>, 2000.
- T.M. Corsi, S. Boyson, A. Verbraeck, S. Van Houten, C. Han, and J.R. Macdonald. The Real-Time Global Supply Chain Game: New Educational Tool for Developing Supply Chain Management Professionals. *Transportation Journal*, 45(3):61, 2006.
- R Costanza, L Wainger, C Folke, and Kg Maler. Modeling Complex Ecological Economic-Systems - Toward An Evolutionary, Dynamic Understanding Of People And Nature. *Bio-science*, 43(8):545–555, SEP 1993. ISSN 0006-3568.
- J.M. Coulson and J.F. Richardson. *Coulson & Richardson's Chemical Engineering*. Butterworth-Heinemann, 1999.
- J. Cramer. Criteria voor duurzame biomassa productie. *Final report of the project group Duurzame productie van biomassa*, page 33, July 2006.
- B.F.W. Croke, J.L. Ticehurst, R.A. Letcher, J.P. Norton, L.T. H. Newham, and A.J. Jakeman. Integrated assessment of water resources: Australian experiences. *Water Resour. Manag.*, 21(1):351–373, 2007.
- J.P. Crutchfield. The calculi of emergence: Computation, dynamics and induction. *Physica D*, 75(1-3):11–54, Aug 1994.
- L. M. Cysneiros and E. Yu. Requirements engineering for large-scale multi-agent systems. *Lect. Note. Comput. Sci.*, 2603:39–56, 2003.
- K. Damen and A. Faaij. A Greenhouse Gas Balance of two Existing International Biomass Import Chains. *Mitigation and Adaptation Strategies for Global Change*, 11(5):1023–1050, 2006.
- A. Damodaran. *Corporate Finance: Theory and Practice*. Wiley, 1997.

- P. D'Aquino, C. Le Page, F. Bousquet, and A. Bah. Using self-designed role-playing games and a multi-agent system to empower a local decision-making process for land use management: The SelfCormas experiment in senegal. *Jasss-the J. Artificial Societies Social Simulation*, 6 (3), 2003.
- C. Darwin. *The Origin of the Species*. Penguin books, 1985.
- P.A. David. Path dependence and varieties of learning in the evolution of technological practice. *Technological Innovation as an Evolutionary Process*, 119, 2000.
- P. Davidsson. Multi agent based simulation: Beyond social simulation. *Lect. Note. Artif. Intell.*, 1979:97–107, 2001.
- C.B. Davis. Integration of life cycle analysis within agent based modeling using a case study on bio-electricity. Master's thesis, Delft University Of Technology, November 2007.
- C.B. Davis, I. Nikolic, and G.P.J. Dijkema. Integrating life cycle analysis with agent based modeling: Deciding on bio-electricity. In *Proceedings of the NGInfra-IEEE International Scientific Conference "Building Networks for a Brighter Future"*, November 2008.
- R. Dawkins. *The Selfish Gene*. Oxfrd University press, 1990.
- H. de Bruijn and E. ten Heuvelhof. *Networks and Decision Making*. Utrecht, Lemma publishers, 2000.
- M. de Bruijne and M. van Eeten. Systems that should have failed: Critical infrastructure protection in an institutionally fragmented environment. *Journal of Contingencies and Crisis Management*, 15(1):18–29, 2007. doi: 10.1111/j.1468-5973.2007.00501.x. URL <http://www.blackwell-synergy.com/doi/abs/10.1111/j.1468-5973.2007.00501.x>.
- DA DeLaurentis and WA Crossley. A Taxonomy-based Perspective for Systems of Systems Design Methods. *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 1, 2005.
- J.B. DeLong. *Macroeconomics*. McGraw-Hill Boston, Mass, 2002.
- D.C. Dennet. *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. Simon & Schuster, New York, 1996.
- D.C. Dennett. *Freedom Evolves*. Viking, New York, 2003.
- A. Deshpande and D. Riehle. The Total Growth of Open Source. In *Proc. Fourth IFIP International Conference on Open Source Systems (OSS2008)*. Springer, 2008.
- J.M. Diamond. *Collapse: how societies choose to fail or succeed*. Blackwell Synergy, 2005.
- G.P.J. Dijkema. *Process system Innovation By Design – Towards a Sustainable Petrochemical Industry*. PhD thesis, TU Delft, 2004.

- G.P.J. Dijkema and R.M. Stikkelman. Positionpaper Eemsdelta. “perspectieven en kansen voor duurzame economische ontwikkeling van de eemsdelta”. een systematische verkenning. Technical Report 1-97, Delft University of Technology, Faculty of Technology, Policy and Management, 2006.
- G.P.J. Dijkema, J. Grievink, and M.P.C. Weijnen. Functional Modelling for a Sustainable Petrochemical Industry. *Process Safety and Environmental Protection/Official Journal of the European Federation of Chemical Engineering: Part B*, 81(B5):331–340, 2003.
- GPJ Dijkema, DJ Van Zanten, and J. Grievink. Public roles and private interests in petrochemical clusters: model-based decision support of the regional development board. *Chemical Engineering Research and Design*, 83:739–751, 2005.
- G.P.J. Dijkema, S. Biesheuvel, S. IJsselstijn, I. Nikolic, J. Baggen, J. Stoop, and O. vd. Riet. Combinatie van infrastructuren, systematische verkenning van (on)mogelijkheden van ruimtesparende infrastructuur voor het rotterdams haven- en industriegebied. Technical Report ISBN: 978-90-5638-170-7, Port Research Centre Rotterdam-Delft, Delft, 2007a.
- G.P.J. Dijkema, I. Nikolic, S. Biesheuvel, S. IJsselstijn, J. Baggen, J. Stoop, O. van Riet, H.A. Weustenenk, and J. Smits. More infrastructure capacity per acre? methodology to establish the (im)possibilities of combination of infrastructures. In Toh Ah Cheong and Chan Eng Soon, editors, *Proceedings of the 2nd International Maritime-Port Technology and Development Conference Sept. 26-28 2007*, number ISBN: 978-981-05-9102-1, pages 127–132. Singapore: Research Publishing Services, 2007b.
- E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1: 269–271, 1959. URL <http://jmvidal.cse.sc.edu/library/dijkstra59a.pdf>.
- P. Dillenbourg. Over-scripting cscl: The risks of blending collaborative learning with instructional design. In Paul A Kirschner, editor, *Three worlds of CSCL: Can we support CSCL?*, pages 61–91. Open Universiteit Nederland, Heerlen, The Netherlands, 2002. Gelezen.
- F. Divina and P. Vogt. A hybrid model for learning word-meaning mappings. *Lect. Note. Artif. Intell.*, 4211:1–15, 2006.
- N.F. Doherty, C.R. Coombs, and J. Loan-Clarke. A re-conceptualization of the interpretive flexibility of information technologies: redressing the balance between the social and the technical. *Eur. J. Inform. Syst.*, 15(6):569–582, 2006.
- P. Donzelli and P. Bresciani. Improving requirements engineering by quality modelling - a quality-based requirements engineering framework. *Journal of Research and Practice in Information Technology*, 36(4):277–294, 2004.
- T.E. Downing, S. Moss, and C. Pahl-Wostl. Understanding climate policy using participatory agent-based social simulation. *Lect. Note. Artif. Intell.*, 1979:198–213, 2001.
- F. Duchin. A world trade model based on comparative advantage with m regions, n goods, and k factors. *Economic Systems Research*, 17(2):141–162, 2005.

- J. Duffy. Learning to speculate: Experiments with artificial and real agents. *Journal of Economic Dynamics and Control*, 25(3-4):295–319, 2001.
- R.D. Duke and J.L.A. Geurts. *Policy Games for Strategic Management: Pathways Into the Unknown*. Dutch University Press; Global, 2004.
- B. Dumont and D.R.C. Hill. Spatially explicit models of group foraging by herbivores: what can Agent-Based Models offer? *Anim. Res.*, 53(5):419–428, 2004.
- H. Ebel, L.I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Phys. Rev. E*, 66(3):035103, Sep 2002. doi: 10.1103/PhysRevE.66.035103.
- D. W. Eccles and P.T. Groth. Wolves, bees, and football: Enhancing coordination in sociotechnological problem solving systems through the study of human and animal groups. *Comput. Hum. Behav.*, 23(6):2778–2790, 2007.
- N. Economides. The economics of networks. *Int. J. Industrial Organization*, 14(6):673–699, 1996.
- A.C. Edmondson. Learning from Mistakes is Easier Said than Done: Group and Organizational Influences on the Detection and Correction of Human Error. *The Journal of Applied Behavioral Science*, 40(1):66, 2004.
- T.M. Egyedi and Z. Verwater-Lukszo. Which standards’ characteristics increase system flexibility? Comparing ICT and batch processing infrastructures. *Technology in Society*, 27(3):347–362, 2005.
- J Ehrenfeld. The importance of leas – warts and all. *Journal of Industrial Ecology*, 1(2):41–49, 1997a.
- J. R. Ehrenfeld. Industrial ecology - Paradigm shift or normal science? *Am. Behavioral Scientist*, 44(2):229–244, 2000.
- J.R. Ehrenfeld. Industrial ecology: A framework for product and process design. *Journal of Cleaner Production*, 5(1-2):87–95, 1997b.
- B. Elvers, W. Gerhartz, S. Hawkins, W.E. Russey, H. Hofmann, G. Schulz, F. Ullmann, M. Pikart-Müller, H. Günzler, M. Polke, et al. *Ullmann’s encyclopedia of industrial chemistry*. VCH, 1988.
- J.M. Epstein. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60, 1999.
- T. Erber and S. Putterman. Randomness in quantum mechanics—nature’s ultimate cryptogram? *Nature*, 318(6041):41–43, 1985.
- M. Etienne. Sylvopast: a multiple target role-playing game to assess negotiation processes in sylvopastoral management planning. *Jasss-the J. Artificial Societies Social Simulation*, 6(2), 2003.

- P. Faratin, C. Sierra, and N.R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artif. Intell.*, 142(2):205–237, 2002.
- J. Fargione, J. Hill, D. Tilman, S. Polasky, and P. Hawthorne. Land Clearing and the Biofuel Carbon Debt. *Science*, 319(5867):1235, 2008.
- A. E. Farrell, H. Zerriffi, and H. Dowlatabadi. Energy infrastructure and security. *Ann. Rev. Environment Resources*, 29:421–469, 2004.
- M. Fathian, B. Amiri, and A. Maroosi. Application of honey-bee mating optimization algorithm on clustering. *Appl. Math. Comput.*, 190(2):1502–1513, 2007.
- M. Fehler, F. Klugl, and F. Puppe. Techniques for analysis and calibration of multi-agent simulations. *Lect. Note. Artif. Intell.*, 3451:305–321, 2005.
- E.A. Feigenbaum and P. McCorduck. *The 5th Generation*. Reading: Addison-Wesley Pub. Co., 1983.
- J. Filipe. An organizational semiotics model for multi-agent systems design. *Lect. Note. Artif. Intell.*, 1937:449–456, 2000.
- M. Fischer-Kowalski and C. Amann. Beyond IPAT and Kuznets Curves: Globalization as a Vital Factor in Analysing the Environmental Impact of Socio-Economic Metabolism. *Population & Environment*, 23(1):7–47, 2001.
- G. Fontana. Why we live in the computational universe, 2005. URL <http://www.citebase.org/abstract?id=oai:arXiv.org:physics/0511157>.
- C.L. Forgy. Rete a fast algorithm for the many pattern many object pattern match problem. *Artificial Intelligence*, 19(1):17 – 37, 1982. ISSN 0004-3702.
- J. W. Forrester. Industrial dynamics - a major breakthrough for decision makers. *Harvard Business Rev.*, 36(4):37–66, 1958.
- J.W. Forrester and J. Wright. *Industrial dynamics*. MIT Press, 1961.
- HN Fowler. Plato in Twelve Volumes. IV: Cratylus, Parmenides, Greater Hippias, Lesser Hippias (Loeb Classical Library, 167), 1977.
- X. Franch, G. Grau, E. Mayol, C. Quer, C. Ayala, C. Cares, F. Navarrete, M. Haya, and P. Botella. Systematic construction of i\* strategic dependency models for socio-technical systems. *International Journal of Software Engineering and Knowledge Engineering*, 17(1): 79–106, 2007.
- H.G. Frederickson and K.B. Smith. *The Public Administration Theory Primer*. Westview Press, 2003.
- T. Friedman. *The World is Flat: A Brief History of the Globalized World in the 21st Century*. Allen Lane–Penguin Group, London, ISBN 0-713-99878-4, 2005.
- R. Frischknecht and G. Rebitzer. The ecoinvent database system: a comprehensive web-based LCA database. *Journal of Cleaner Production*, 13(13-14):1337–1343, 2005.

- R.A. Frosch and N.E. Gallopoulos. *The Treatment and Handling of Wastes*, chapter Towards an industrial ecology, pages 269–292. Chapman and Hall, London, 1992.
- T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software—practice And Experience*, 21(11):1129–1164, November 1991.
- R. Fuentes, J.J. Gomez-Sanz, and J. Pavo. Activity theory for the analysis and design of multi-agent systems. *Lect. Note. Comput. Sci.*, 2935:110–122, 2004.
- R. Fuentes, J.J. Gomez-Sanz, and J. Pavon. Requirements elicitation for agent-based applications. *Lect. Note. Comput. Sci.*, 3950:40–53, 2006.
- K.K. Fullam, J. Sabater-Mir, and K.S. Barber. A design foundation for a trust-modeling experimental testbed. *Lect. Note. Artif. Intell.*, 3577:95–109, 2005.
- R.B. Fuller. *Operating Manual for Spaceship Earth*. Simon and Schuster, New York, 1969.
- S.O. Funtowicz and J.R. Ravetz. Science for the post-normal age. *Futures*, 25(7):739–755, 1993.
- D.J. Futuyma. Evolutionary interactions among herbivorous insects and plants. *Coevolution. Sinauer, Sunderland, Mass*, pages 207–231, 1983.
- B.R. Gaines and D.H. Norrie. Coordinating societies of research agents - IMS experience. *Integr. Comput.-Aided Eng.*, 4(3):179–190, 1997.
- J. Gall. *The Systems Bible: The Beginner's Guide to Systems Large and Small: Being the Third Edition of Systemantics*. General Systemantics Press, 2002.
- A. Gandolfi. *Formicai, imperi, cervelli. Introduzione alla scienza della complessità*. Bollati-Boringhieri, Torino., 1999.
- F. Gandon, A. Poggi, G. Rimassa, and P. Turci. Multi-agent corporate memory management system. *Appl. Artif. Intell.*, 16(9-10):699–720, 2002.
- R. Garcia-Flores. Uses of agent-based modeling in innovation/new product development research. *J. Prod. Innov. Manage.*, 22(5):380–398, 2005.
- R. Garcia-Flores and X.Z. Wang. A multi-agent system for chemical supply chain simulation and management support. *Advanced Planning and Scheduling Solutions in Process Industry*, 2003.
- J. Garreau. *Radical Evolution: The Promise and Peril of Enhancing Our Minds, Our Bodies—And What It Means to Be Human*. Broadway Books, 2006.
- L. Gasser. Perspectives on organizations in multi-agent systems. *Lect. Note. Artif. Intell.*, 2086: 1–16, 2001.
- F. W. Geels. From sectoral systems of innovation to socio-technical systems - Insights about dynamics and change from sociology and institutional theory. *Research Policy*, 33(6-7):897–920, 2004.

- M.R. Genesereth and N.J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1987.
- J. Gennari, M.A. Musen<sup>b</sup>, R.W. Ferguson<sup>b</sup>, W.E. Grossod, M. Crubézy, H.Eriksson, N.F. Noy, and S.W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89, 2003. [http://dx.doi.org/10.1016/S1071-5819\(02\)00127-1](http://dx.doi.org/10.1016/S1071-5819(02)00127-1).
- I. Giardina and J.P. Bouchaud. Bubbles, crashes and intermittency in agent based market models. *The European Physical Journal B-Condensed Matter*, 31(3):421–437, 2003.
- L. Giordano, A. Martelli, and C. Schwind. Specifying and verifying systems of communicating agents in a temporal action logic. *Lect. Note. Artif. Intell.*, 2829:262–274, 2003.
- J. Gleick. *Chaos; Making a New Science*. New York, Viking., 1988.
- G. Gordon. The development of the General Purpose Simulation System (GPSS). *History of programming languages I table of contents*, pages 403–426, 1978.
- V.I. Gorodetski, O. Karsayev, A. Khabalov, I. Kotenko, L.J. Popyack, and V.A. Skormin. Agent-Based Model of Computer Network Security System: A Case Study. *Lecture Notes In Computer Science*, pages 39–50, 2001.
- T.E. Graedel and B.R. Allenby. *Industrial Ecology*. Prentice Hall, 1995.
- J. M. Grandmont. Expectations formation and stability of large socioeconomic systems. *Econometrica*, 66(4):741–781, 1998.
- D.W. Green and J.O. Maloney. *Perry's chemical engineers' handbook*. McGraw-Hill, 1984.
- J.L. Gross and J. Yellen, editors. *Handbook of Graph Theory*. Discrete Mathematics and its Applications. CRC Press, 2004. ISBN 1-58488-090-2.
- I.E. Grossmann and A.W. Westerberg. Research challenges in process systems engineering. *AIChE Journal*, 46(9):1700–1703, 2000.
- T.R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–220, 1993.
- J. Guinee, editor. *Handbook on Life Cycle Assessment*, volume 7 of *Eco-Efficiency in Industry and Science*. Springer, 2002.
- N. Gurewitch. Reset, perry bible fellowship cartoon. <http://pbfcomics.com/?cid=PBF111-Reset.jpg>, 2005. Used with permission.
- P. Guyot and A. Drogoul. Multi-agent based participatory simulations on various scales. *Massively Multi-agent Systems I*, 3446:149–160, 2005.
- P. Guyot and S. Honiden. Agent-based participatory simulations: Merging multi-agent systems and role-playing games. *Journal of Artificial Societies and Social Simulation*, 9(4):8, 2006. ISSN 1460-7425. URL <http://jasss.soc.surrey.ac.uk/9/4/8.html>.

- P. Guyot, A. Drogoul, and S. Honiden. Multi-agent participatory simulations between experimental economics and role-playing games. *Agent-based Approaches In Econ. Social Complex Systems Iv*, 3:11–19, 2007.
- W. Hadeli, P. Valckenaers, M. Kollingbaum, and H. Van Brussel. Multi-agent coordination and control using stigmergy. *Comput. Ind.*, 53(1):75–96, 2004.
- R.J. Hansman, C. Magee, R. De Neufville, R. Robins, and D. Roos. Research agenda for an integrated approach to infrastructure planning, design and management. *International Journal of Critical Infrastructures*, 2(2-3):146 – 159, 2006.
- J. Hartmanis, V. Sewelson, and N. Immerman. Sparse sets in np-p: Exptime versus nexptime. In *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 382–391, New York, NY, USA, 1983. ACM. ISBN 0-89791-099-0. doi: <http://doi.acm.org/10.1145/800061.808769>.
- B. Hayes. Computing science: Computational creationism. *American Scientist*, 97:392–396, 1999.
- M. Heemskerk, K. Wilson, and M. Pavao-Zuckerman. Conceptual models as tools for communication across disciplines. *Conservation Ecology*, 7(3), 2003.
- R. Heijungs. *A Theory of the Environment and Economic Systems*. Advances in Ecological Economics. Edward Elgar Publishing, 2001.
- S. Heijungs, R. and Suh. *The Computational Structure of Life Cycle Assessment.*, volume 11 of *Eco-Efficiency in Industry and Science*. Kluwer Academic Publishers, 2002.
- L. Henrickson. Old wine in a new wineskin - College choice, college access using agent-based modeling. *Soc. Sci. Comput. Rev.*, 20(4):400–419, 2002.
- R. Henrion, P. Li, A. Moller, M.C. Steinbach, M. Wendt, and G. Wozny. Stochastic optimization for operating chemical processes under uncertainty. *Online Optimization of Large Scale Systems*, pages 457–478, 2001.
- L.M. Hercog and T.C. Fogarty. Social simulation using a multi-agent model based on classifier systems: The emergence of vacillating behaviour in the "El farol" Bar problem. *Lect. Note. Artif. Intell.*, 2321:88–111, 2002.
- PM Herder. *Process Design in a Changing Environment: Identification of Quality Demands Governing the Design Process*. Delft University Press, Delft, The Netherlands, 1999.
- P.M. Herder, M.P.C. Weijnen, and I. Bouwmans. Designing infrastructures from a complex systems perspective. *Journal of Design Research*, 7(1), 2008.
- G.E. Herrera. Stochastic bycatch, informational asymmetry, and discarding. *J. Environ. Econ. Manage.*, 49(3):463–483, 2005.
- R. Hill, S. Polovina, and M.D. Beer. Improving ROSE with an enriched modelling framework. *Lect. Note. Comput. Sci.*, 3950:94–108, 2006.

- C. E. Hmelo-Silver and R. Azevedo. Understanding complex systems: Some core challenges. *J. Learning Sciences*, 15(1):53–61, 2006.
- G.D. Hobson and W. Pohl. *Modern Petroleum Technology*. John Wiley & Sons, 1984.
- J. Holland. *Emergence: From Chaos to Order*. Addison-Wesley, 1997.
- J.H. Holland. *Hidden Order; How Adaptation Builds Complexity*. Reading, Addison-Wesley., 1996.
- W. Hoschek. Colt project. <http://dsd.lbl.gov/~hoschek/colt/index.html>, 2004.
- G. Hotsma. Manager product development, Groningen Seaports. Personal Communication, 2006.
- M. Houwing and I. Bouwmans. Agent-Based Modeling of Residential Energy Generation with Micro-CHP. In *Proceedings of the 2nd International Conference on Integration of Renewable and Distributed Energy Resources, Napa, CA, USA*, 2006.
- C.J. Huang, A. J.C. Trappey, and Y.H. Yao. Developing an agent-based workflow management system for collaborative product design. *Ind. Manage. Data Syst.*, 106(5-6):680–699, 2006.
- H. Iizuka, K. Suzuki, M. Yamamoto, and A. Ohuchi. Learning of virtual words utilized in negotiation process between agents. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E83A(6):1075–1082, 2000.
- R. L. Iman, J. C. Helton, and J. E. Campbell. An approach to sensitivity analysis of computer-models .1. introduction, input variable selection and preliminary variable assessment. *J. Quality Technology*, 13(3):174–183, 1981.
- M.E. Inchiosa and M.T. Parker. Overcoming design and development challenges in agent-based modeling using ascape. *Proc. Natl. Acad. Sci. U. S. A.*, 99:7304–7308, 2002.
- GD Ingram, IT Cameron, and KM Hangos. Classification and analysis of integrating frameworks in multiscale modelling. *Chemical Engineering Science*, 59(11):2171–2187, 2004.
- A. Jamaković. *Characterization of complex networks: Application to robustness analysis*. PhD thesis, TU Delft, 2008.
- I.L. Janis. *Groupthink: Psychological studies of policy decisions and fiascoes*. Houghton Mifflin Boston, 1982.
- M.A. Janssen and E. Ostrom. Empirically based, agent-based models. *Ecol. Soc.*, 11(2), 2006.
- DH Jantzen. When is it coevolution. *Evolution*, 34:611–612, 1980.
- S. Jarupathirun and F.M. Zahedi. Dialectic decision support systems: System design and empirical evaluation. *Decis. Support Syst.*, 43(4):1553–1570, 2007.
- P. Jeffrey and B.S. McIntosh. Description, diagnosis, prescription: a critique of the application of co-evolutionary models to natural resource management. *Environmental Conservation*, 33(4):281–293, DEC 2006. ISSN 0376-8929. doi: {10.1017/S03768906003444}.

- K.A. Jehn and E.A. Mannix. The Dynamic Nature of Conflict: A Longitudinal Study of Intragroup Conflict and Group Performance. *Academy Of Management Journal*, 44(2):238–251, 2001.
- N.R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277 – 296, 2000a. ISSN 0004-3702.
- N.R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000b.
- P. Jiang, Q.T. Mair, and Z.R. Feng. Agent alliance formation using ART-networks as agent belief models. *J. Intell. Manuf.*, 18(3):433–448, 2007.
- A. Jones. An Environmental Assessment of Food Supply Chains: A Case Study on Dessert Apples. *Environmental Management*, 30(4):560–576, 2002.
- R.W. Jones. The Structure of Simple General Equilibrium Models. *The Journal of Political Economy*, 73(6):557, 1965.
- C.M. Jonker and J. Treur. Agent-based simulation of animal behaviour. *Appl. Intell.*, 15(2): 83–115, 2001.
- G. Joseph and K.J.A. Kundig. Copper: Its trade, manufacture, use, and environmental status. Technical report, ASM International - International Copper Association, 1999.
- M. Junginger and A. Faaij. International energy agency bioenergy task 40 country report for the netherlands. Technical Report NWS-E-2005-48, International Energy Agency, 2005.
- G. Kallis. Socio-environmental co-evolution: some ideas for an analytical approach. *International Journal Of Sustainable Development And World Ecology*, 14(1):4–13, FEB 2007. ISSN 1350-4509.
- G.A. Kaminka, D. V. Pynadath, and M. Tambe. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *J. Artif. Intell. Res.*, 17:83–135, 2002.
- T. Karp. Transforming organisations for organic growth: The DNA of change leadership. *Journal of Change Management*, 6(1):3–20, 2006.
- J. S. Katz. Indicators for complex innovation systems. *Research Policy*, 35(7):893–909, 2006.
- S.A. Kauffman. *Investigations*. Oxford University Press, New York., 2000.
- S.A. Kauffman. *Reinventing the Sacred: A New View of Science, Reason and Religion*, 2008.
- S.A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- S.A. Kauffman and S. Johnsen. Coevolution to the edge of chaos - coupled fitness landscapes, poised states, and coevolutionary avalanches. *J. Theoretical Biol.*, 149(4):467–505, 1991.

- S.A Kauffman and E.D. Weinberger. The NK Model of Rugged Fitness Landscapes and Its Application to Maturation of the Immune Response. *Molecular Evolution on Rugged Landscapes: Proteins, RNA, and the Immune System: the Proceedings of the Workshop on Applied Molecular Evolution and the Maturation of the Immune Response, Held March, 1989 in Santa Fe, New Mexico*, 1991.
- S. Kaufman. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. New York: Oxford University Press, 1995.
- J.J. Kay. On complexity theory, exergy and industrial ecology: Some implications for construction ecology. In C. Kibert, J. Sendzimir, and B. Guy, editors, *Construction Ecology: Nature as the Basis for Green Buildings*, pages 72–107. Spon Press, 2002.
- J.J. Kay, H.A. Regier, M. Boyle, and G. Francis. An ecosystem approach for sustainability: addressing the challenge of complexity. *Futures*, 31(7):721–742, 1999.
- J. Keirstead. Evaluating the applicability of integrated domestic energy consumption frameworks in the uk. *Energy Policy*, 34(17):3065–3077, 2006.
- R. Kemp and J. Rotmans. *Towards Environmental Innovation Systems*, chapter The Management of the Co-Evolution of Technical, Environmental and Social Systems. Springer Berlin Heidelberg, 2005.
- R. Kemp, D. Loorbach, and J. Rotmans. Transition management as a model for managing processes of co-evolution towards sustainable development. *Perspectives on Radical Changes to Sustain-able Consumption and Production (SCP)*, 20:21, 2006.
- R. Kemp, D. Loorbach, and J. Rotmans. Transition management as a model for managing processes of co-evolution towards sustainable development. *International Journal Of Sustainable Development And World Ecology*, 14(1):78–91, FEB 2007. ISSN 1350-4509.
- E. A. Kendall. Role models - Patterns of agent system analysis and design. *BT Technol. J.*, 17(4):46–57, 1999.
- E.A. Kendall, M.T. Malkoun, and C.H. Jiang. A Methodology for Developing Agent Based Systems. *Lecture Notes In Computer Science*, pages 85–99, 1996.
- T.B. Kepler, M.A. Marti-Renom, S.M. Maurer, A.K. Rai, G. Taylor, and M.H. Todd. Open Source Research—the Power of Us. *Australian Journal of Chemistry*, 59(5), 2006.
- Z. Lukszo K.H. van Dam, I. Nikolic and G.P.J. Dijkema. Towards a generic approach for analyzing the efficiency of complex networks. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 745–750. IEEE, April 2006.
- W.J.M. Kickert, E.H. Klijn, and J.F.M. Koppenjan. *Managing Complex Networks: Strategies for the Public Sector*. Sage, 1997.
- P. A. Kirschner, P. J. Beers, H. P. A. Boshuizen, and W. H. Gijsselaers. Coercing shared knowledge in collaborative learning environments. *Computers in Human Behavior*, 24(2): 403–420, 2008.

- F. Klein and H. Giese. Grounding social interactions in the environment. *Lect. Note. Artif. Intell.*, 3830:139–162, 2006a.
- F. Klein and H. Giese. Analysis and design of physical and social contexts in multi-agent systems. *Lect. Note. Comput. Sci.*, 3914:91–108, 2006b.
- K.M.Carley. Computational organization science: A new frontier. *Proc. Natl. Acad. Sci. U. S. A.*, 99:7257–7262, 2002.
- KNMI. De toestand van het klimaat in nederland 2008. Technical report, KNMI, 2008.
- R.E. Knox and J.A. Inkster. Postdecision dissonance at post time. *J Pers Soc Psychol*, 8(4): 319–23, 1968.
- D. Koesrindartoto, J. Sun, and L. Tesfatsion. An agent-based computational laboratory for testing the economic reliability of wholesale power market designs. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 931–936, 2005.
- A. Kofod-Petersen and J. Cassens. Using activity theory to model context awareness. *Lect. Note. Artif. Intell.*, 3946:1–17, 2005.
- G.L. Kolfschoten, R.O. Briggs, G.J. de Vreede, P.H.M. Jacobs, and J.H. Appelman. Conceptual foundation of the thinklet concept for collaboration engineering. *International Journal of Human Computer Science*, 7(64):611–621, 2006.
- J. Koppenjan and E.H. Klijn. *Managing Uncertainties in Networks, A network approach to problem solving and decision making*. London, Routledge, 2004.
- L. Koren. *Wabi-Sabi*. Stone Bridge Press, 1994.
- J.H. Kraay. Dutch approaches to surviving with traffic and transport. *Transport Reviews*, 16 (4):323–343, 1996.
- P. Kroes. *Functions in Biological and Artificial Worlds: Comparative Philosophical Perspectives*, chapter Technical Artifacts Engineering Practice And Emergence. Vienna Series in Theoretical Biology. MIT Press, 2009. ISBN 978-0-262-11321-2.
- W. Krohn and W. van den Daele. Science as an agent of change: finalization and experimental implementation. *Soc. Sci. Inf. Sci. Soc.*, 37(1):191–222, 1998.
- F. Kun, G. Kocsis, and J. Farkas. Cellular automata for the spreading of technologies in socio-economic systems. *Physica A*, 383(2):660–670, 2007.
- R. Lachman. Ai, decision science, and psychological theory in decisions about people - A case study in jury selection. *AI Mag.*, 19(1):111–129, 1998.
- G. Lanzola, L. Gatti, S. Falasconi, and M. Stefanelli. A framework for building cooperative software agents in medical applications. *Artif. Intell. Med.*, 16(3):223–249, 1999.
- B. Lee, A. Barua, and A.B. Whinston. Discovery and representation of causal relationships in MIS research: A methodological framework. *MIS Q.*, 21(1):109–136, 1997.

- R.S.T. Lee. ijade surveillant - an intelligent multi-resolution composite neuro-oscillatory agent-based surveillance system. *Pattern Recognit.*, 36(6):1425–1444, 2003.
- W. P. Lee. Applying domain knowledge and social information to product analysis and recommendations: an agent-based decision support system. *Expert Syst.*, 21(3):138–148, 2004.
- W. Leontief. Environmental Repercussions and the Economic Structure: An Input-Output Approach. *International Library Of Critical Writings In Economics*, 92:24–33, 1998.
- R. Levin. *Complexity : Life at the Edge of Chaos*. University of Chicago Press, 2000.
- DA Lindberg, BL Humphreys, and AT McCray. The Unified Medical Language System. *Methods Inf Med*, 32(4):281–91, 1993.
- CE Lindblom, DK Cohen, and J.N. Warfield. Usable Knowledge, Social Science and Social Problem Solving. *IEEE Transactions on Systems, Man and Cybernetics*, 10(5):281–281, 1980.
- A. Lopez-Parades, D. Sauri, and J. M. Galan. Urban water management with artificial societies of agents: The FIRMABAR simulator. *Simulation-Transactions Soc. For Modeling Simulation Int.*, 81(3):189–199, 2005.
- E.N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.
- J. Lovelock. *Gaia : A New Look at Life on Earth*. Oxford University Press, 3rd edition, 2000.
- Q. Lu and B. Botha. Process development: a theoretical framework. *Int. J. Production Research*, 44(15):2977–2996, 2006.
- N. Luhmann. *Social Systems*. Stanford, CA: Stanford University Press., 1995.
- I. Lustick. Ps-i: A user-friendly agent-based modeling platform for testing theories of political identity and political stability. *JASSS*, 5(3), 2002.
- T. Lynam, F. Bousquet, C. Le Page, P. d’Aquino, O. Barreteau, F. Chinembiri, and B. Mombeshora. Adapting science to adaptive managers: Spidergrams, belief models, and multi-agent systems modeling. *Conservation Ecology*, 5(2):337–354, 2002.
- K. E. Maani and R.Y. Cavana. *Systems Thinking and Modelling: Understanding Change and Complexity*. Auckland, Prentice-Hall., 2000.
- M.L. Maher, M. Rosenman, and K. Merrick. Agents for multidisciplinary design in virtual worlds. *AI EDAM-Artif. Intell. Eng. Des. Anal. Manuf.*, 21(3):267–277, 2007.
- S. Manson. Land use in the southern Yucatan peninsular region of Mexico: Scenarios of population and institutional change. *Comput. Environ. Urban Syst.*, 30(3):230–253, 2006.
- S.M. Manson. Agent-based modeling and genetic programming for modeling land change in the Southern Yucatan Peninsular Region of Mexico. *Agric. Ecosyst. Environ.*, 111(1-4):47–62, 2005.

- G. Mantovani. Social context in HCI: A new framework for mental models, cooperation, and communication. *Cogn. Sci.*, 20(2):237–269, 1996.
- A. De Martino, M. Marsili, and I.P. Castillo. Statistical mechanics analysis of the equilibria of linear economies. *J. Stat. Mech.-Theory Exp.*, 2004.
- I. Mayer and W. Veeneman. *Games in a World of Infrastructures: Simulation-games for Research, Learning and Intervention*. Eburon, 2002.
- J.G. McGuire, D.R. Kuokka, J.C. Weber, J.M. Tenenbaum, T.R. Gruber, and G.R. Olsen. SHADE: Technology for Knowledge-based Collaborative Engineering. *Concurrent Engineering*, 1(3):137, 1993.
- BS McIntosh and P Jeffrey. Transferring theories of biological (co)evolution to socio-natural science: A reply to Rammel and Staudinger. *International Journal Of Sustainable Development And World Ecology*, 11(1):1–8, MAR 2004. ISSN 1350-4509.
- M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
- D.H. Meadows, D.L. Meadows, J. Randers, and W.W. Behrens. The limits to growth. *New York*, 1972.
- A. Melchior, H. Wiig, and K. Telle. *Globalisation and Inequality: World Income Distribution and Living Standards 1960-1998*. Royal Norwegian Ministry of Foreign Affairs, 2000.
- D.C. Mikulecky. The emergence of complexity: science coming of age or science growing old? *Computers and Chemistry*, 25(4):341–348, 2001.
- H. J. Miser and E. S. Quade. *Handbook of Systems Analysis; Overview of uses, procedures, applications, and practice*. Chichester, Wiley., 1985.
- R. Mizoguchi, M. Ikeda, and K. Sinitsa. Roles of Shared Ontology in AI-ED Research. In *Proc. of AI-ED*, volume 97, pages 537–544, 1997.
- E. Morin. Organization and complexity. *Tempos In Science Nature: Structures, Relations, Complexity*, 879:115–121, 1999.
- Y. Morishima, H. Nakajima, S. Brave, R. Yamada, H. Maldonado, C. Nass, and S. Kawaji. The role of affect and sociality in the agent-based collaborative learning system. *Lect. Note. Comput. Sci.*, 3068:265–275, 2004.
- S. Moss and B. Edmonds. Sociology and simulation: Statistical and qualitative cross-validation. *Am. J. Sociol.*, 110(4):1095–1131, 2005.
- S. Moss and E. Norling. Multi-agent-based simulation: Why bother? *Lect. Note. Artif. Intell.*, 3891:1–13, 2006.
- J. Mylopoulos, M. Kolp, and P. Giorgini. Agent-oriented software development. *Lect. Note. Artif. Intell.*, 2308:3–17, 2002.

- T. Howe N. Collier and M. North. Onward and upward: the transition to repast 2.0. In *Proceedings of the First Annual North American Association for Computational Social and Organizational Science Conference*, page 5, Pittsburgh, PA USA, 2003. North American Association for Computational Social and Organizational Science.
- NAS, National Academy of Engineering, and Institute of Medicine. *Facilitating Interdisciplinary Research*. The National Academic Press, Washington, D.C., 2005.
- R.R. Negenborn, B. De Schutter, and H. Hellendoorn. Multi-agent model predictive control of transportation networks. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, pages 296–301, 2006.
- C. L. Nehaniv and K. Dautenhahn. Like me ? - Measures of correspondence and imitation. *Cybern. Syst.*, 32(1-2):11–51, 2001.
- M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
- L.D. Nielsen, P.H. Jespersen, T. Petersen, and L.G. Hansen. Freight transport growth - a theoretical and methodological framework. *European Journal of Operational Research*, 144(2):295–305, 2003.
- F.W. Nietzsche, F. Nietzsche, and RJ Hollingdale. *Thus Spoke Zarathustra: A Book for Everyone and No One*. Penguin Classics, 1961.
- I. Nikolic and G.P.J. Dijkema. Intelligent infrastructures in a changing environment: innovation and evolution of industry-infrastructure networks. *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, pages 581–586, 2005.
- I. Nikolic and G.P.J. Dijkema. Framework for Understanding and Shaping Systems of Systems The case of industry and infrastructure development in seaport regions. *System of Systems Engineering, 2007. SoSE'07. IEEE International Conference on*, pages 1–6, 2007.
- I. Nikolic, G.P.J. Dijkema, K.H. van Dam, and Z. Lukszo. General methodology for action-oriented industrial ecology, complex systems approach applied to the Rotterdam industrial cluster. In *Proceedings of the IEEE SNC 2006*, IEEE International Conference On Networking, Sensing and Control, Ft. Lauderdale, Florida, U.S.A., April 23-25 2006.
- I. Nikolic, P.J. Beers, and G.P.J. Dijkema. Facilitating interdisciplinary modelling of complex problems. In *Proceedings of HICSS-40*, Hawaii Internatinal Conference On Systems Science. University of Hawaii at Manoa, January 3-6 2007.
- I. Nikolic, G.P.J. Dijkema, and K.H. van Dam. *The Dynamics of Regions and Networks in Industrial Ecosystems*, chapter Understanding and Shaping the Evolution of Sustainable Large-Scale Socio-Technical Systems. Number ISBN: 978-1-84720-742-5. Edward Elgar, 2009.
- N.F. Noy and D.L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University, 2001. Accessed February 2007.
- P. Noy and M. Schroeder. Advancing profile use in agent societies. *Lect. Note. Comput. Sci.*, 3071:360–375, 2003.

- A. Ocampo and J. V. Munch. The REMIS approach for rationale-driven process model evolution. *Software Process Dynamics Agility, Proc.*, 4470:12–24, 2007.
- A. Omicini, A. Ricci, and M. Viroli. Coordination artifacts as first-class abstractions for MAS engineering: State of the research. *Lect. Note. Comput. Sci.*, 3914:71–90, 2006.
- J.W. Oosterwijk. Naar een open economie. *Economisch Statistische Berichten*, 91(4477), 2006.
- S. Ossowski and A. Omicini. Coordination knowledge engineering. *Knowl. Eng. Rev.*, 17(4): 309–316, 2002.
- M. Ottens, P. Kroes, M. Franssen, and I. van de Poel. Modeling engineering systems as socio-technical systems. In *IEEE International Conference On Systems, Man, And Cybernetics, Conference Proceedings*, volume 6, pages 5685–5690, 2004.
- M. Ottens, M. Franssen, P. Kroes, and I. van de Poel. Modelling infrastructures as socio-technical systems. *International Journal of Critical Infrastructures*, 2(2-3):133 – 145, 2006.
- Ozone Secretariat of the United Nations Environment Programme. *Handbook for the Montreal Protocol on Substances that Deplete the Ozone Layer*. UNEP, 7th edition, 2006.
- J.F. Padgett, D. Lee, and N. Collier. Economic production as chemistry. *Ind. Corp. Change*, 12(4):843–877, 2003.
- S.E. Page. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton University Press Princeton, NJ, USA, 2007.
- C. Pahl-Wostl. Towards sustainability in the water sector - The importance of human actors and processes of social learning. *Aquat. Sci.*, 64(4):394–411, 2002.
- C. Pahl-Wostl. Information, public empowerment, and the management of urban watersheds. *Environ. Modell. Softw.*, 20(4):457–467, 2005.
- C. Pahl-Wostl. The implications of complexity for integrated resources management. *Environmental Modelling and Software*, 22(5):561–569, 2007.
- C. Pahl-Wostl and M. Hare. Processes of social learning in integrated resources management. *J. Community Appl. Soc. Psychol.*, 14(3):193–206, 2004.
- J.Y.C. Pan and J.M. Tenenbaum. An intelligent agent framework for enterprise integration. *IEEE Trans. Syst. Man Cybern.*, 21(6):1391–1408, 1991.
- S. Panebianco and C. Pahl-Wostl. Modelling socio-technical transformations in wastewater treatment - A methodological proposal. *Technovation*, 26(9):1090–1100, 2006.
- ML Parry, OF Canziani, JP Palutikof, PJ van der Linden, and CE Hanson. Climate Change 2007: Impacts, Adaptation and Vulnerability. *Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, 2007.
- M. Pechoucek, J. Barta, and V. Marik. Cplant: Coalition planning multi-agent system for humanitarian relief operations. *Lect. Note. Artif. Intell.*, 2322:363–375, 2002.

- M. Penker and H.K. Wytrzens. Scenarios for the Austrian food chain in 2020 and its landscape impacts. *Landsc. Urban Plan.*, 71(2-4):175–189, 2005.
- E. Perea-López, B.E. Ydstie, and I.E. Grossmann. A model predictive control strategy for supply chain optimization. *Computers and Chemical Engineering*, 27(8-9):1201–1218, 2003.
- P. Persson, J. Laaksolahti, and P. Lonnqvist. Understanding socially intelligent agents - A multilayered phenomenon. *IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum.*, 31(5):349–360, 2001.
- G.P. Peters. Efficient algorithms for life cycle assessment, input-output analysis, and monte-carlo analysis. *International Journal of Life Cycle Assessment*, 8, 2006.
- JR Petit, J. Jouzel, D. Raynaud, NI Barkov, JM Barnola, I. Basile, M. Bender, J. Chappellaz, M. Davis, G. Delaygue, et al. Climate and atmospheric history of the past 420,000 years from the Vostok ice core. *Antarctica: Nature*, 399(6735):429–436, 1999.
- P. Petta and R. Trappl. Emotions and agents. *Lect. Note. Artif. Intell.*, 2086:301–316, 2001.
- G. Pezzulo and G. Calvi. Designing and implementing MABS in akira. *Lect. Note. Comput. Sci.*, 3415:49–64, 2005.
- S.L. Pimm and P. Raven. Biodiversity: extinction by numbers. *Nature*, 403(6772):843–845, 2000.
- K.R. Popper. *The Logic of Scientific Discovery*. Routledge, 2002.
- M.E. Porter. Location, Competition, and Economic Development: Local Clusters in a Global Economy. *Economic Development Quarterly*, 14(1):15, 2000.
- I. Prigogine. *Introduction to thermodynamics of irreversible processes*. New York: Interscience, 1967, 3rd ed., 1967.
- I. Prigogine and I. Stengers. *Order out of chaos : man's new dialogue with nature*. Boulder, CO : New Science Library : Distributed by Random House, 1984. ISBN: 0877733023 : 0394542045.
- M. Proctor, M. Neale, P. Lin, and M. Frandsen. Drools. Technical report, <http://labs.jboss.com/portal/jbossrules/docs>, 2007.
- H. Purnomo, G.A. Mendoza, R. Prabhu, and Y. Yasmi. Developing multi-stakeholder forest management scenarios: a multi-agent system simulation approach applied in indonesia. *Forest Policy Econ.*, 7(4):475–491, 2005.
- A.M. Ramanath and N. Gilbert. The design of participatory agent-based social simulations. *JASSS*, 7(4), 2004.
- C. Rammel, B.S. McIntosh, and P. Jeffrey. (Co)evolutionary approaches to sustainable development. *International Journal Of Sustainable Development And World Ecology*, 14(1):1–3, FEB 2007a. ISSN 1350-4509.

- C. Rammel, S. Stagl, and H. Wilfing. Managing complex adaptive systems - A co-evolutionary perspective on natural resource management. *Ecological Economics*, 63(1):9–21, JUN 15 2007b. ISSN 0921-8009. doi: {10.1016/j.ecolecon.2006.12.014}.
- R. Razavi, J. F. Perrot, and N. Guelfi. Adaptive modeling: An approach and a method for implementing adaptive agents. *Lect. Note. Artif. Intell.*, 3446:136–148, 2005.
- C. Reed, B. Boswell, and R. Neville. Multi-agent patient representation in primary care. *Lect. Note. Artif. Intell.*, 3581:375–384, 2005.
- W. Reich. Reasoning about other agents: A plea for logic-based methods. *JASSS*, 7(4), 2004.
- M. Remondino. *Analysis of Agent Based Paradigms for Complex Social Systems Simulation*. PhD thesis, Universita di Torino, 2004.
- A.F. Ribas, B.L. Panisello, and A. V. Suso. The higher level of performance in the chemical industry cluster located in tarragona. Workshop on spatial networks and clusters, Universitat Rovira i Virgili, 2003.
- A. Ricci, M. Viroli, and A. Omicini. Environment-based coordination through coordination artifacts. *Lect. Note. Comput. Sci.*, 3374:190–214, 2005.
- G.P. Richardson and D.F. Andersen. Teamwork in group model building. *System Dynamics Review*, 11:113–113, 1995.
- H. Rittel and M.M. Webber. Planning problems are wicked problems. *Developments in Design Methodology*, pages 135–144, 1984.
- H.W.J. Rittel and M.M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.
- M. M. Robbins and A. M. Robbins. Simulation of the population dynamics and social structure of the Virunga mountain gorillas. *Am. J. Primatol.*, 63(4):201–223, 2004.
- R.M. Roberts. Serendipity: Accidental Discoveries in Science. *Serendipity: Accidental Discoveries in Science*, by Royston M. Roberts, page 288, 1989.
- M. Rodriguez-Achach, L. Casillas, and F. J. Espinosa. Diffusion of innovations in a social network under mixed Pareto-Nash strategies. *Physica A*, 335(3-4):671–676, 2004.
- W. G. Rong and K. C. Liu. A multi-agent architecture for CSCW systems: From organizational semiotics perspective. *Lect. Note. Artif. Intell.*, 4088:766–772, 2006.
- R.R. Rosenberg and D.C. Karnopp. *Introduction to Physical System Dynamics*. McGraw-Hill, Inc. New York, NY, USA, 1983.
- R. J. Ross, B. McEleney, R. Kelly, T. Abu-Amer, M. Walsh, J. Carson-Berndsen, and G. M. P. O’Hare. Speaking autonomous intelligent devices. *Lect. Note. Artif. Intell.*, 2464:240–245, 2002.

- R.J.F. Rossetti, R.H. Bordini, A.L.C. Bazzan, S. Bampi, R. Liu, and D. Van Vliet. Using BDI agents to improve driver modelling in a commuter scenario. *Transp. Res. Pt. C-Emerg. Technol.*, 10(5-6):373–398, 2002.
- J. Rotmans. Methods for ia: The challenges and opportunities ahead. *Environmental Modeling and Assessment, Special Issue: Challenges and opportunities for Integrated Assessment*, 3(3): 155–179, August 1998. doi: 10.1023/A:1019019024003.
- G. Rotundo and M. Ausloos. Microeconomic co-evolution model for financial technical analysis signals. *Physica A*, 373:569–585, 2007.
- E. Rouwette, J.A.M. Vennix, and T. van Mullekom. Group model building effectiveness: a review of assessment studies. *System Dynamics Review*, 18(1):5–45, 2002.
- A. F. Rutkowski and B. Walle. Cultural dimensions and prototypical criteria for multi-criteria decision support in electronic markets: A comparative analysis of two job markets. *Group Decis. Negot.*, 14(4):285–306, 2005.
- K. Ryokai, C. Vaucelle, and J. Cassell. Virtual peers as partners in storytelling and literacy learning. *J. Comput. Assist. Learn.*, 19(2):195–208, 2003.
- J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.
- A.P. Sage. On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information, Knowledge, Systems Management*, 2(4):325–345, 2001.
- K. Sakamornsnguan. Supporting condition for sustainable metal network development: agent-based simulation on infrastructural investment of cu & al. Master’s thesis, Delft University of Technology, 2007.
- D. L. Sallach. Social theory and agent architectures - Prospective issues in rapid-discovery social science. *Soc. Sci. Comput. Rev.*, 21(2):179–195, 2003.
- J.R. Sand, S.K. Fischer, and V.D. Baxter. Energy and Global Warming Impacts of HFC Refrigerants and Emerging Technologies. *US Department of Energy and AFEAS, Arlington, Va, USA*, 215, 1997.
- C. Sansores and J. Pavon. Agent based simulation for social systems: From modeling to implementation. *Lect. Note. Artif. Intell.*, 4177:79–88, 2006.
- T.J. Santner, W.I. Notz, and B.J. Williams. *The Design and Analysis of Computer Experiments*. Springer, 2003.
- E.R. Santos, E. Boff, and R.M. Vicari. Semantic web technologies applied to interoperability on an educational portal. *Lect. Note. Comput. Sci.*, 4053:308–317, 2006.
- F. W. Scharpf. *Games Real Actors Play, Actor-Centered Institutionalism in Policy Research*. Westview Press., 1997.

- T. A. Schenk, G. Loffler, and J. Rauh. Agent-based simulation of consumer behavior in grocery shopping on a regional level. *J. Bus. Res.*, 60(8):894–903, 2007.
- S.H. Schneider, W.E. Easterling, and L.O. Mearns. Adaptation: Sensitivity to natural variability, agent assumptions and dynamic climate changes. *Clim. Change*, 45(1):203–221, 2000.
- J.A. Schumpeter. *The Theory of Economic Development: An Inquiry Into Profits, Capital, Credit, Interest, and the Business Cycle*. Transaction Publishers, 1983.
- A Sen. On The Darwinian View Of Progress. *Population And Development Review*, 19(1): 123–137, MAR 1993. ISSN 0098-7921.
- P.M. Senge, C. Roberts, and R.B. Ross. *The Fifth Discipline Fieldbook; Strategies and Tools for Building a Learning Organization*. Garden City, Doubleday, 1994.
- A. Serenko. The development of an instrument to measure the degree of animation predisposition of agent users. *Comput. Hum. Behav.*, 23(1):478–495, 2007.
- A. P. Shah and A. R. Pritchett. Work-environment analysis: environment centric multi-agent simulation for design of socio-technical systems. *Lect. Note. Comput. Sci.*, 3415:65–77, 2005.
- C.R. Shalizi. Methods and techniques of complex systems science: An overview. *arXiv.org*, arXiv.org:nlin/0307015, 2006. URL <http://www.citebase.org/abstract?id=oai:arXiv.org:nlin/0307015>.
- F.H. Shang, X. F. Li, and J. Xu. An agent-based soft computing society with application in the management of establishment of Hydraulic Fracture in Oil field. *Lect. Note. Comput. Sci.*, 3612:505–514, 2005.
- K.A. Shepsle and M.S. Bonchek. *Analyzing politics*. WW Norton & Company, 1997.
- H.A. Simon. *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization*. Free Press, New York, 4 edition, 1997.
- H.A. Simon. *Models of Bounded Rationality*. MIT Press, 1982.
- J. Singh, L. Cuttler, and J. B. Silvers. Toward understanding consumers’ role in medical decisions for emerging treatments: Issues, framework and hypotheses. *J. Bus. Res.*, 57(9): 1054–1065, 2004.
- A. Sirikijpanichkul, K.H. van Dam, L. Ferreira, and Z. Lukszo. Optimizing the Location of Intermodal Freight Hubs: An Overview of Agent Based Modelling Approach. *J Transp Sys Eng & IT*, 4, 2007.
- D. Skole and C. Tucker. Tropical Deforestation and Habitat Fragmentation in the Amazon: Satellite Data from 1978 to 1988. *Science*, 260(5116):1905, 1993.
- E.R. Smith and F.R. Conrey. Agent-based modeling: A new approach for theory building in social psychology. *Pers. Soc. Psychol. Rev.*, 11(1):87–104, 2007.
- K. Stathis, O. de Bruijn, and S. Macedo. Living memory: agent-based information management for connected local communities. *Interact. Comput.*, 14(6):663–688, 2002.

- M. Stefik. *Introduction to knowledge systems*. Morgan Kaufmann Publishers, San Francisco, Calif, 1995a.
- M. Stefik. *Introduction to Knowledge Systems*. Morgan Kaufmann, 1995b.
- J D Sterman. Learning in and about complex systems. *System Dynamics Review*, 10(2-3): 291–330, 1994.
- J. D. Sterman. *Business dynamics: systems thinking and modeling for a complex world*. London, Irwin Professional., 2000.
- S.H. Strogatz and S. Henry. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview Press, 2000.
- S. Suh. Theory of materials and energy flow analysis in ecology and economics. *Ecological Modelling*, 189:251–269, 2005.
- C. Sung and C.T. Maravelias. An attainable region approach for production planning of multiproduct processes. *AIChE Journal*, 53(5):1298–1315, 2007.
- A.G. Sutcliffe. Requirements analysis for socio-technical system design. *Inf. Syst.*, 25(3):213–233, 2000.
- D.D. Suthers. Towards a systematic study of representational guidance for collaborative learning discourse, 23 January 2004 2001.
- J. Tanimoto and H. Fujii. A study on research performance in Japanese Universities: Which is more efficient - A professor who is leading his research group or one who is working alone? The multi-agent simulation knows. *Adv. Complex Syst.*, 6(3):375–391, 2003.
- G. Teisman. *Publiek Management op de Grens van Chaos en Orde: Over Leidinggeven en Organiseren in Complexiteit*. Den Haag, SDU Uitgevers bv., 2005.
- L. Tesfatsion. Agents come to bits: Towards a constructive comprehensive taxonomy of economic entities. *J. Econ. Behav. Organ.*, 63(2):333–346, 2007.
- J.N. Thompson. *The Coevolutionary Process*. University Of Chicago Press, 1994.
- D. Tillman, T.A. Larsen, C. Pahl-Wostl, and W. Gujer. Modeling the actors in water supply systems. *Water Sci. Technol.*, 39(4):203–211, 1999.
- P.M. Torrens and I. Benenson. Geographic automata systems. *Int. J. Geogr. Inf. Sci.*, 19(4): 385–412, 2005.
- A. Troisi, V. Wong, and M. A. Ratner. An agent-based approach for modeling molecular self-organization. *Proc. Natl. Acad. Sci. U. S. A.*, 102(2):255–260, 2005.
- Wilensky U. and Resnick M. Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology*, 8:3–19(17), March 1999. URL <http://www.ingentaconnect.com/content/klu/jost/1999/00000008/00000001/00220140>.
- UN. *Demographic Yearbook of the United Nations*. United Nations Publications, 2006.

- M.A. Upal. Simulating the emergence of new religious movements. *JASSS*, 8(1), 2005.
- J.M. Van Bruggen, P.A. Kirschner, and W. Jochems. External representation of argumentation in cscl and the management of cognitive load. *Learning and Instruction*, 12:121–138, 2002.
- K.H van Dam, M. Houwing, Z. Lukszo, and I. Bouwmans. Agent-based control of distributed electricity generation with micro combined heat and power - Cross-sectoral learning for process and infrastructure engineers. *Computers and Chemical Engineering*, 32(1-2):205–217, 2008.
- J.C.J.M. van den Bergh. Evolutionary thinking in environmental economics. *Journal Of Evolutionary Economics*, 17(5):521–549, OCT 2007. ISSN 0936-9937. doi: {10.1007/s00191-006-0054-0}.
- J.C.M. van den Bergh and M.A. Janssen, editors. *Economics of Industrial Ecology, Materials, Structural Change, and Spatial Scales*. MIT Press, 2005.
- W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, 2007.
- E. van der Voet. *A Handbook of Industrial Ecology*, chapter Substance flow analysis methodology. Edward Elgar Publishing, 2001.
- R.E.J. van Krevelen. Two analysis methods for the evaluation of cluster formation; assessment using a case study on biofuel and bulk chemicals clusters in the netherlands. Master’s thesis, Delft University of Technology, 2007.
- R. van Ree and B. Annevelink. Status report biorefinery 2007. Technical Report Report 847, Agrotechnology and Food Sciences Group, 2007.
- D.J. van Veldhuisen and P.A. Poole-Wilson. The underreporting of results and possible mechanisms of ‘negative’ drug trials in patients with chronic heart failure. *International Journal of Cardiology*, 80(1):19–27, 2001.
- J.A.M. Vennix. Group Model-Building: Tackling Messy Problems. *System Dynamics Review*, 15(4):379–401, 1999.
- J.A.M. Vennix, W. Scheper, and R. Willems. Group Model Building: What Does the Client Think of It. In *Proceedings of the 1993 International System Dynamics Conference*, pages 534–543, 1993.
- E.V. Verhoef, G.P.J. Dijkema, and M.A. Reuter. Process knowledge, system dynamics, and metal ecology. *Journal of Industrial Ecology*, 8(1-2):23–43, 2004. doi: 10.1162/1088198041269382. URL <http://www.mitpressjournals.org/doi/abs/10.1162/1088198041269382>.
- M. Vila, C. Y. Ferro, and M. Rodriguez. Agrupamientos sectoriales territoriales (ast): Reflexiones acerca de los recursos compartidos. *Revista de Economía y Empresa*, XIV(40):87–101, 2000.

- A. Vishwanath and G. M. Goldhaber. An examination of the factors contributing to adoption decisions among late-diffused technology products. *New Media Soc.*, 5(4):547–572, 2003.
- H. von Stackelberg. *The Theory of the Market Economy*. William Hodge, 1952.
- L.J. De Vries. *Securing the public interest in electricity generation markets, The myths of the invisible hand and the copper plate*. PhD thesis, TU Delft, 2004. ISBN 90-5638-123-7.
- M.M. Waldorp. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, New York, 1992.
- W.E. Walker. Policy analysis: a systematic approach to supporting policymaking in the public sector. *Journal of Multi-Criteria Decision Analysis*, 9(1-3), 2000.
- T. Wanyama and B. H. Far. A protocol for multi-agent negotiation in a group-choice decision making process. *J. Netw. Comput. Appl.*, 30(3):1173–1195, 2007.
- S. Wasserman and K. Faust. *Social Network Analysis, Methods and Applications*. New York, Cambridge University Press, 1994, reprint edition 2005.
- D.J Watts and S.H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature(London)*, 393(6684):440–442, 1998.
- C. Webster and F.L. Wu. Coase, spatial pricing and self-organising cities. *Urban Stud.*, 38(11): 2037–2054, 2001.
- H. Weigand, V. Dignum, J.J. Meyer, and F. Dignum. Specification by refinement and agreement: Designing agent interaction using landmarks and contracts. *Lect. Note. Artif. Intell.*, 2577:257–269, 2003.
- A.W. Westerberg, E. Subrahmanian, Y. Reich, S. Konda, D. P. Cunningham, A. H. Dutoit, H. L. Granger, S. Konda, K. C. Marshall, R. C. Milliken, I. A. Monarch, J. P. Neergaard, R. H. Patrick, and M. E. Thomas. Designing the process design process. *Computers & Chem. Engineering*, 21:S1–S9, 1997.
- Wikipedia. Unit testing - wikipedia, the free encyclopedia, 2007. URL [http://en.wikipedia.org/wiki/Unit\\_testing](http://en.wikipedia.org/wiki/Unit_testing). Online; accessed 10-September-2007.
- R. Wilds, S.A. Kauffman, and L. Glass. Evolution of complex dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18:033109, 2008.
- O.E. Williamson. The Economics of Organization: The Transaction Cost Approach. *American Journal of Sociology*, 87(3):548, 1981.
- O.E. Williamson. *The economic institutions of capitalism*. Free Press New York, 1987.
- N Winder, BS McIntosh, and P Jeffrey. The origin, diagnostic attributes and practical application of co-evolutionary theory. *Ecological Economics*, 54(4):347–361, SEP 15 2005. ISSN 0921-8009. doi: {10.1016/j.ecolecon.2005.03.017}.
- S. Wolfram and M. Gad-el Hak. A New Kind of Science. *Applied Mechanics Reviews*, 56:B18, 2003.

- M. Wooldridge. Agent-based software engineering. *IEE Proceedings - Software Engineering*, 144(1):26–37, 1997. doi: 10.1049/ip-sen:19971026. URL <http://link.aip.org/link/?IPS/144/26/1>.
- M. Wooldridge and N.R. Jennings. Intelligent agents - theory and practice. *Knowledge Engineering Rev.*, 10(2):115–152, 1995.
- M. Wooldridge, N.R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
- D. Yamaguchi, Y.F.M. Hassan, and E. Tazaki. New model based on cellular automata and multiagent techniques. *Cybern. Syst.*, 38(1):47–82, 2007.
- S. Yaskawa and A. Sakata. The application of intelligent agent technology to simulation. *Math. Comput. Model.*, 37(9-10):1083–1092, 2003.
- E. Yu. Agent orientation as a modelling paradigm. *Wirtschaftsinformatik*, 43(2):123–132, 2001.
- J. F. Zhang. Residential segregation in an all-integrationist world. *J. Econ. Behav. Organ.*, 54(4):533–550, 2004.
- J.F. Zhang. Growing Silicon Valley on a landscape: an agent-based approach to high-tech industrial clusters. *J. Evol. Econ.*, 13(5):529–548, 2003.
- P. Zhang, S. Peeta, and T. Friesz. Dynamic game theoretic model of multi-layer infrastructure networks. *Netw Spat. Econ.*, 5(2):147–178, 2005.
- N. Zhong, C. N. Liu, and S. Ohsuga. Dynamically organizing KDD processes. *Int. J. Pattern Recognit. Artif. Intell.*, 15(3):451–473, 2001.



**Part IV**  
**Appendices**



## A.1 Agent

In this section the agent level is introduced. Agents are the smallest system elements that generate all emergent behavior. Artificial Intelligence is introduced as a dominant field. Agents are defined as software entities with a state and rules, having inputs and outputs. Agent interaction is structured through interfaces, protocols and message concepts. Adaptiveness, diversity, and interface and protocol similarity are presented as key agent-level properties.

**Artificial Intelligence** Artificial Intelligence (AI) is the dominant field for the discussion of the agent level, as this community has spent a lot of effort to formally describe agents. Indeed, their definitions appear to be the most usable. An agent in the AI field is a self-contained problem solving system capable of autonomous, reactive, pro-active social behavior (Jennings, 2000b; Wooldridge, 1997; Wooldridge and Jennings, 1995). Agents are computer algorithms; they have behavioral rules, and because they are computer entities they are strictly deterministic and follow logical rules. They can have the possibility to learn according to the given rules.

**Actors and agents** In the social and management sciences, “actors” are used to describe agents (Koppenjan and Klijn, 2004; Wasserman and Faust, 1994, reprint edition 2005). Actors are mostly individuals, coalitions or organizations that have to follow social rules but that have the freedom to deviate from these rules in a not necessarily rational or logical way. From these definitions and descriptions, it is clear there are slight differences between the concepts/terms agent and actor. In this thesis, when using the word *actor* we refer to the real-world entity, and when using the word *agent* we refer to its computer-simulated abstraction. Examples of actors in systems are cars on the road network, government agencies in the political system, species or individual animals in the ecosystem, firms in the industry system. Figure A.1 presents the visualization of a single agent. The agent is considered to consist of: input, state and rules, and output. The properties associated with agents are adaptiveness, internal diversity and interface

and protocol similarity of this level. While the whole agent level consists out of numerous agents please note that only a single agent is visualized on the right side of Figure A.1.

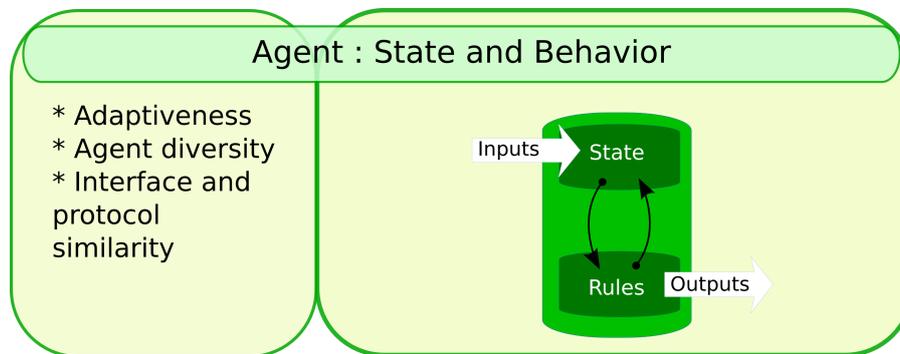


Figure A.1: Agent level

### A.1.1 Agent Interaction

The inputs of an agent are the outputs of other agents and inputs from the environment. The agent’s actions are its outputs; these outputs can effect the environment and other agents. Jennings refers to these outputs as effectors (Jennings, 2000b). An input for a human agent is, for example, a message transmitted through speech from another agent that conveys an approaching vehicle on a road. The output of the human agent is a hasty retreat back to the sidewalk and maybe a message of thanks. This interaction can be further formalized. In this thesis, all interaction is explicitly defined to consist of three components: interface, protocol and message. The horizontal arrows in Figure A.1 represents the directionality of the interaction.

**Interface** In order for agents to interact, they must have a common type of interface. Agents quite literally need to be able to speak the same language. Language in this case must be understood in an abstract sense; that is, information or effects must be exchanged in a way that all interacting parties can make sense of the messages (Wooldridge and Jennings, 1995). For example, in an English speaking debating group a person speaking only Chinese will not be able to interact. In a more physical sense, flowers interact with butterflies through scents and colors butterflies can perceive. The flower’s color and scent is compatible with the butterfly’s vision and scent organs. In chemistry, a boundary surface between phases - solid, liquid, and gas - across which interaction is possible can be seen as an interface. Network cables in a computer, the computer screen and the graphical user interface (GUI) are all examples of interfaces. Ontologies (see Section 6.5, formal conceptual schemes for describing domain-specific knowledge, are also examples of interfaces (Gruber, 1993).

**Protocol** A protocol is the set of rules that describes the format of the message. It can be seen as the grammar of the message. In order for interaction to take place, agents must also have some degree of protocol similarity. An example of a protocol is the Hypertext Transfer Protocol (HTTP). “Het groene boekje”, the book describing the rules of the Dutch language, is another example of a protocol. The Foundation for Intelligent Physical Agents (FIPA) develops

standards for agents <sup>1</sup>. The FIPA standard is an agent communication language, which consists of an interface and a protocol.

**Message** The message is the content of the arrows depicted as input and output in Figure A.1. A message is everything that flows through the interface - such as the warning of an approaching car or the scent of a flower. Depending on the protocol, messages can consist of words, light, energy, etc. An integrated example of the components that make up interaction is the connection of an electrical appliance. The socket in which a plug is placed forms an interface, the 220 V running through the interface is the protocol, and the electrical energy is the message.

### A.1.2 Agent Structure

**State** The state of an agent is a specification of the particular collection of parameters that defines an agent (Wooldridge and Jennings, 1995). The state *is* all of the relevant information that gives it its identity. Based on the current state, the inputs and the available behavior, the agent will perform some action, causing an output to happen. It is the set of information about what this agent is at this moment. A light switch, for example, has only two states: on and off. There are observable and unobservable states. The states of a shy flirting couple, for example, may be that of excitement and wishful thinking. These states are private and not observable by the other. A sudden redness in the face is a publicly visible part of that state.

**Rules** Rules describe how the different inputs and internal states are translated to outputs and new states. Holland (Holland, 1996) calls rules the “internal models” of agents. For example, when observing our courting couple, their internal behavioral rules tell them to observe the publicly visible state and outputs of the other and react accordingly. When one of the couple starts displaying behavior that can be interpreted as positive, the other partner can positively react. Rules of computer agents in models of complex systems are usually rational, based on the maximization of utility of some sort. However, there is no requirement of rationality for agents. Agents can act irrationally, if given such rules. However, they cannot act *illogically*, given that they are computer entities. Actors (humans) can act rationally, irrationally or even illogically. Further, actors can have different behavioral rules in different societal contexts (Scharpf, 1997). Since these rules are usually private and unobservable, abstraction and modelling of their behavior is difficult.

### A.1.3 Important Properties

Three main properties (among others) that are relevant at this level are: adaptiveness, internal diversity and interface similarity. The first is a property of the agents themselves, and the latter two are properties of the agents relative to each other. Below, these properties are elaborated upon and it is shown how they are described in different fields.

**Adaptiveness** As an influence of the surroundings or internal states, an agent can change its rules; this is called adaptiveness (Holland, 1996; Kauffman, 1993; Levin, 2000). It is the ability

---

<sup>1</sup><http://fipa.org/>

of an agent to change towards a more “optimal” or “fit” state. For example, the bodies immune system becomes stronger after it has been exposed to pathogens frequently. If one considers a biological species as an agent, the evolution of species over time is an example of adaptiveness. Learning ([Argyris and Schon, 1996](#)) is a specific form of adaptiveness. A manager in a business learns through the feedback it receives on the work done. Learning requires an agent to have a memory. If the agents possess a memory mechanism, their actions will be partly determined by their past actions. The state and dynamics of an agent, therefore, also depend on its previous states and dynamics. The manager in our example has a memory and DNA of a species can store inactive genes to be reactivated later.

**Agent diversity** Complex systems exhibit large internal diversity of agents. Internal diversity means that agents have their own state and rules that may differ from agent to agent. The agent diversity is responsible for complex emergent behavior on higher levels ([Kauffman, 2000](#); [Kaufman, 1995](#); [Page, 2007](#); [Waldorp, 1992](#)). The molecules in a room, for example, all have a different velocity. Most people have differing opinions about the US president. All parties in the electricity market determine the prices of their electricity on their own and in different ways. The field of Public Administration and Management Science refers to “pluriformity”. Diversity in this sense are the different perspectives of actors (human agents) ([de Bruijn and ten Heuvelhof, 2000](#)). However, there is much diversity between the agents, so that they would fail to interact at all, one could not speak about a complex system.

**Interface and protocol similarity** All agents of a system share a common interface and protocol. Agents must be similar enough in interface and protocol to be able to interact. All organisms in ecosystems metabolize and grow. People respond to financial incentives like environmental subsidies and taxes. Ottens ([Ottens et al., 2006](#)), for example, refers to regulation and rules as being the common interface humans have agreed upon. In the field of management, people need to have a common language in order to understand each other; as a result they use the same codes or jargon. Mobile phones share the same data protocol, and all over the world ships use the same containers in which to transport different goods. Ecosystems, too, consist of many different species but ecosystems maintain a common interface. In nature almost anything can eat something else. The interface is the body structure of organisms, composed of fats, sugars, and proteins that most living organisms can process.

## A.2 Network

Networks are introduced as medium-level systems. They describe the structure of interactions between agents. The concepts of nodes and edges are introduced. Average shortest path length and degree distribution are introduced as two traditional metrics, as well as the limitation of graph theory to measure multidimensional graphs. Network growth, function and evolution, together with network topology, are presented as important properties of this level.

All the definitions of complex systems presented above contain the notion of structured interactions in networks. The field that studies networks is graph theory. It defines a network as ([Newman, 2003](#)):

a set of items, which we will call vertices or sometimes nodes, with connections between them called edges.

Paraphrasing the definition, a network is an abstraction of reality, where all system components are either nodes or edges, things or connections. In our case, agents are vertices or nodes, edges are interactions between the agents and are created at the moment of interaction. The concepts and system properties used at the network level are presented in Figure A.2.

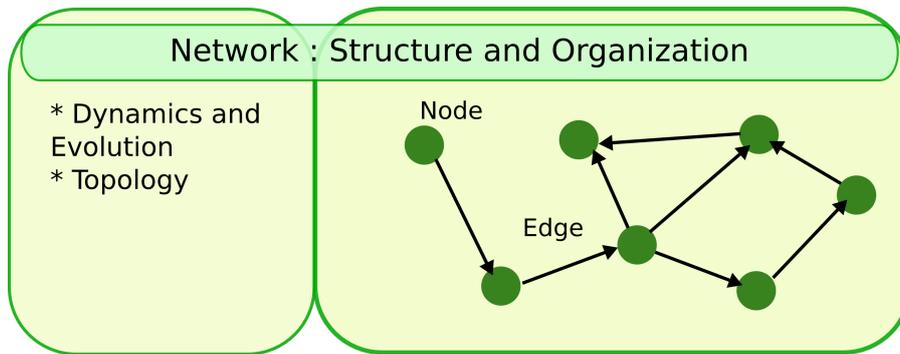


Figure A.2: Network level

### A.2.1 Nodes and Edges

**Node** In Complex Adaptive Systems, examples of nodes are Internet routers that are connected with fiber-optic connections, people in a social network, organisms in a food web, factories in a production network, etc. (Ahuja et al., 1993; Newman, 2003). From these examples, it is clear that nodes can have more than one edge. When considering networks in a strictly graph-theoretical perspective, the only difference between nodes is the number and weight of their edges and the identity of the nodes connected to them.

**Edge** Edges are created the moment agents (nodes) interact. Edges can be abstract and non-material, such as information exchange or political influence. They can also be very concrete and physical, such as electricity or water flows. In a physical network the edges are the cables, roads or pipelines for the transportation of goods and information. On the other hand, in a social network these edges are based on the relationships and actual contacts individuals have with other individuals. The edges in the social networks can be social rules, informal power and norms. In Management Science, for example, organizational networks develop and exist because of the interdependencies and repetitiveness of interactions that lead to stable patterns in social networks (Koppenjan and Klijn, 2004). In economics, relations are represented as the exchange of goods, services and money.

Edges can be undirected, directed or bidirectional. Edges can have weights associated with them. Water in pipes, for example, always flows from the area of high pressure to an area of low pressure.

**Multidimensional networks** Complex adaptive systems, as discussed earlier, are multiformal. This means that the interaction networks that give rise to system structure are multidimensional. With traditional graph theory we cannot handle these types of networks, since one

can only describe networks with edges of a single type, i.e., where the edges represent exactly one type of interaction. One can, for example, only model the intensity of car traffic on road networks. The intensity of bicycles on bicycle paths cannot be added to the road traffic model. The different weights that can be attributed to the different edges are insufficient, as a weight does not imply a different type of edge. The same problem concerns the nodes. This limitation severely restricts the use of graph theory in understanding  $\lambda$ -systems. Reducing the components and their interaction into just two types of items greatly oversimplifies the description of the system. However, knowing this limitation, it is still interesting to use graph theory as a perspective when examining system structures.

## A.2.2 Network Metrics

The structure of interactions can be quantified by using network metrics. The metrics presented here mainly serve to increase the understanding of basic network structure, while there are a great many different metrics available in graph theory (Jamaković, 2008). Since the types of networks we are dealing with are multidimensional, most traditional metrics are not sufficient. Multidimensional metrics are developed in Chapters 6,7 and 8. Two basic networks that can be used, irregardless of a network's multidimensionality, are the Average Shortest Path Length and the Degree Distribution.

**Average Shortest Path Length** This is an important metric because it provides a fairly robust way to measure the networks *diameter*. Diameter should be understood fairly literally. A network with many nodes that is highly connected is literally very small, that is, it is easy to traverse from a node to any other very quickly. A network with few sparsely connected nodes is large, since it takes many steps to traverse it.

*Average Shortest Path Length*  $l$  is expressed by eq. A.1 and is valid for an undirected or bidirectional graph.

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} \quad (\text{A.1})$$

$N$  is the number of nodes, while  $d$  is the distance (hop count) between two nodes. In a case of a directed graph, the best procedure to calculate  $l$  is by using Dijkstra's algorithm (Dijkstra, 1959). This is a highly efficient algorithm to find the shortest paths from a single point to the rest of the points in a network. In other words, this is a methodology you could use to examine a map and find the shortest routes to all the surrounding cities from your house.

**Degree distribution** Degree of the node denotes the number of in- and outgoing edges a node has. Degree distribution characterizes a network by describing the histogram of the nodes with a specific degree. This metric is especially useful when comparing real-world networks to theoretical models. It provides an insight in the connection 'equity' between the system's subcomponents and offers structural insight. A particularly interesting distribution is the so-called power law (or scale-free) distribution. This type of distribution, also known as 20-80 distribution, is very common in natural and man-made systems (Clauset et al., 2007). In a network with power law distribution, a small number of nodes have very many edges, and a large number of nodes have very few edges. See Figure A.3 for an illustration of an observed power law distribution.



Figure A.3: Example of a power law distribution. The Y axis represents the number of saves per user at wiki.tudelft.nl.

### A.2.3 Important Properties

**Network growth** One of the dominant approaches to studying network evolution is through stochastic models of network growth (Kauffman and Weinberger, 1991; Kaufman, 1995). In such models, the model's nodes have a stochastic function that determines when and to which other node they connect to. The work in this thesis is based on Kauffman's basic idea of nodes searching for other nodes to connect to. However, most work done in the network growth field (Barabasi and Albert, 1999) does not contain any domain-specific notions. In order to understand the evolution of  $\lambda$ -systems, models based in relevant knowledge domains are needed. Multiformal aspects and realistic observed knowledge of decision-making processes, as well as technology descriptions of industrial clusters are used to describe how nodes search and connect to other nodes. This approach, embedded in the relevant domains, provides a much more realistic base for understanding network evolution.

**Network function** One of the main issues in studying industrial  $\lambda$ -system networks is the relation between network structure and network function. This is especially evident in the case of petrochemical clusters and their sustainability performance. Is it possible to maintain the industry's societal function, the production of chemicals and energy, while reorganizing its structure to deal with a resource problem and at the same time increase its sustainability performance? (Dijkema, 2004) This poses the problem of understanding network evolution.

**Network evolution** The relevant discipline studying industrial networks is Process Systems Engineering (PSE). In PSE it is well recognized that networks are not static. The flows through the network (of mass, energy, information, etc.) can vary in type and size. This is often referred to as network dynamics. Examples of network dynamics from other fields are: variable energy flows through the power grid, signals traveling through nerve cells during motion and the intensity changes in friendships during a lifetime. In PSE the network dynamics view is common. However, the evolution of the network structure has not been addressed adequately (Dijkema, 2004). Also the network structure is subject to change: new nodes are added to or

removed from the network and/or new edges emerge or disappear between existing nodes. This is referred to as network evolution. Example of network evolution from other fields are the addition of more cell phones to the network (nodes) that lead to more and different calls being made (edges). Another example is the broadening of a person’s social network by introducing friends to other friends.

**Topology** Topology describes the structure of the network. It can be understood as the structure of interaction between agents. In terms of network topology, two extreme types are possible (Barabási et al., 2001; Jamaković, 2008; Newman, 2003). They are differentiated by the type of process that generates them. Random attachment networks are created when a new node is connected to a randomly chosen other node in a network (see Figure A.4 a). Preferential attachment networks develop when a new node preferentially connects to highly connected nodes (Figure A.4 b). Topologies of most real-world networks are in between these two types.

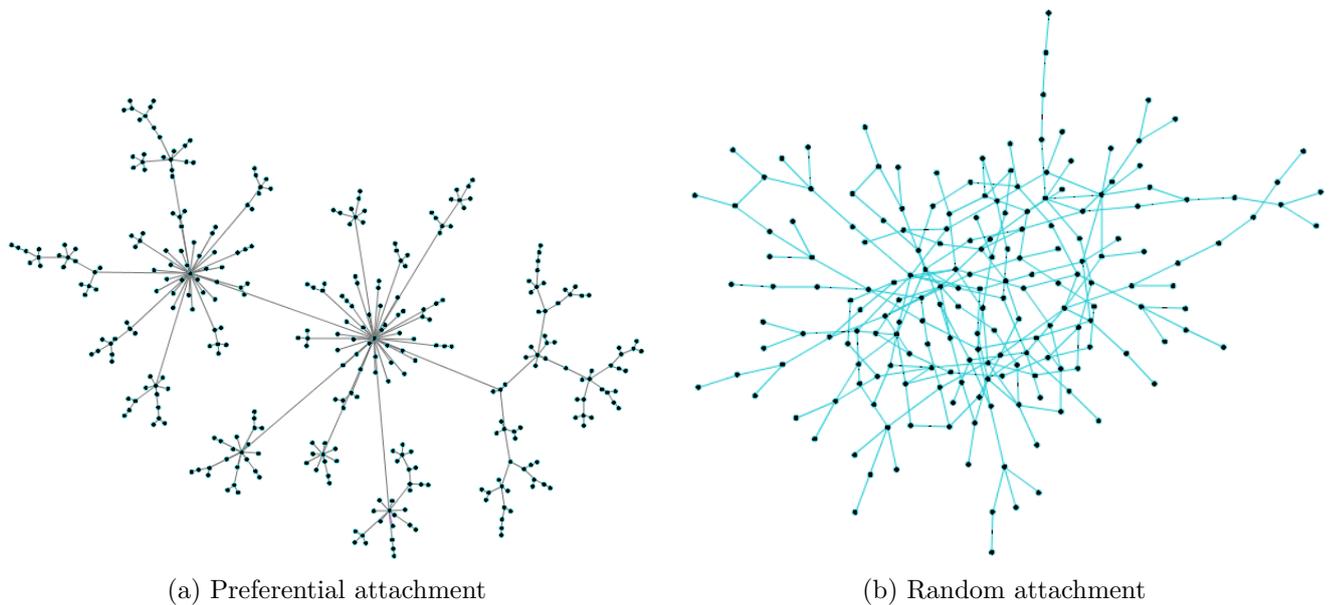


Figure A.4: Network topologies

### A.3 System

The system is introduced as the highest level, where emergent properties are observed. This level is conceptualized as an entity similar to the agent, with aggregate in- and outputs, and aggregate states and rules. Relevant properties identified at this level are emergent behavior, self-organization, path dependency, chaos, robustness and instability.

In this section we will examine system properties and behavior at the highest level, the overarching system level. All the considered definitions of complex systems contain the notion of emergent overall system behavior. This level describes the *whole* system: a human, the

world, a species, etc. The system level conceptualization has a structure similar to that of the agent level: input, rules and output. The properties we define at this level are: path dependency, robustness and instability (see Figure A.5. Non-linear dynamics (Prigogine and Stengers, 1984) has established a tradition of formally describing overall system behavior. Thus their vocabulary is most appropriate for our purpose of precisely defining the vocabulary useful in  $\lambda$ -system modelling.

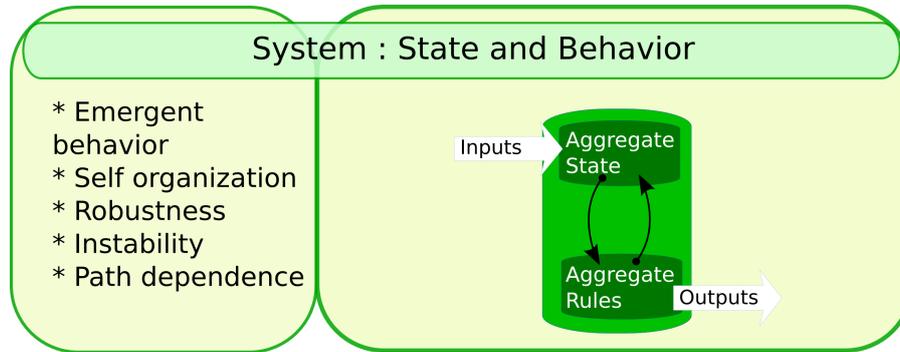


Figure A.5: System Level

### A.3.1 Aggregation

**System inputs and outputs** A system's aggregate inputs are a combination of all inputs coming from the environment. Even though the system consists of many agent level components, we can aggregate those inputs and treat them as a single input at the system level. The same is true for system output. For example, the brain sends electric pulses to the arm to raise it. The actual system input is a myriad of electrical signals to the individual muscle cells that all individually react to their own electrical signals. However, at the system level, the brain's command to raise the arm causes a single system output: a coordinated movement.

**Aggregate state and rules** The overall state of the system necessarily consists of the aggregates states of its consistent components at the agent level. The overall system's rules are the aggregate rules of the components. For example, a country's GDP is built up from the individual incomes of all individuals and firms within it. A flock of birds' response to flee from a predator is an aggregate rule based on individual fleeing behavior.

**System and agent similarity** The systems view introduced earlier does not prescribe the level of aggregation when describing a system and its components. The aggregation is dependent on the observer and the task at hand. Often, what is seen as the system level description in one formalization can be seen as the agent level in another. For example, a country can be seen as an agent if one observes international politics. If, however, one studies the interaction between municipalities, the country is viewed at the system level.

Aggregate states and rules, inputs and outputs are essentially the same concepts at the agent and system level. The part is a whole, and the whole can in turn be a part. This property is particularly important when developing models that are expected to change in scale as they are developed. For example, starting at the level of individual persons, one can construct a system

that represents a firm. By shifting scales and using firms as agents, their interactions will create a system level that is defined as a region. Shifting scales again, many regions will form a global industrial network. By each time treating the aggregate inputs/outputs and agent inputs, this scaling can be easily performed.

### A.3.2 Important Properties

In this section several important properties observable at the system level will be presented. These are emergent behavior, self-organization, path dependency, robustness and instability.

**Emergent behavior** The behavior that a complex adaptive system shows as a whole, the system's output - is called emergent behavior. Emergence is the process by which new characteristics arise once the system is constituted (Crutchfield, 1994; Morin, 1999). Newman (Newman, 2003) understands emergent behavior as processes on networks. Or as Jennings (Jennings, 2000b) puts it: emergent behavior is the behavioral phenomena that cannot be deconstructed solely in terms of the behavior of the individual agents. It is important to realize that emergence is *no magic*. That is, emergent behavior is the logical consequence of the interactions in the system and the organizational structure of the system which gives the parts qualities that they could not have if they were isolated from the organizing whole (Morin, 1999). Emergent behavior is simpler to understand - and more insightful - than the collection of processes that cause it. The emergent behavior itself, however, is not always predictable or obvious. Examples of emergent behavior are traffic jams, power blackouts and bankruptcy. Policy processes can also be viewed as the system output of the policy system. Policy processes are unpredictable due to incomplete information and unclear values (of the agents in the policy network). According to (Lindblom et al., 1980), an outcome emerges from the interactions among decision makers. As Cohen (Cohen et al., 1972) states, "a decision is an outcome or an interpretation of several relatively interdependent streams within an organization." Others argue that human consciousness is emergent behavior of the brain (Dennet, 1996). Emergent behavior has also been observed in the economic literature. Externalities are considered to be undesired emergent properties. These externalities can be positive: the lovely garden of the neighbors that adds value to your own house, or negative: sitting next to a person who is smoking.

Emergent properties are what we look for when studying  $\lambda$ -system evolution. Shaping evolution is all about causing desired emergent properties like sustainability to appear, while preventing the undesired ones such as pollution.

**Self-organization** Self-organization is specific form of emergent behavior (system output). Self-organization is a process by which a system achieves a different output through internal processes, without any external input (Kay, 2002; Prigogine and Stengers, 1984). For example, the morphogenesis (construction of shape) of embryos, a fully functional organism self-assembles from a single fertilized cell (Campbell, 2002). Bose-Einstein condensate, for example, (Anderson et al., 1995) is a unique state of matter that appears under very low temperatures and pressures, at which all atoms collapse to a single quantum state. With autopoiesis (self-steering), for example, society self-organizes in a way that limits the amount of individual choice, thereby providing a self-steering and a more predictable system (Luhmann, 1995). Please note that self-organization is different from adaptiveness, as self-organization originates from the system

itself and adaptiveness is a reaction of the agents to the changing environment. It is our goal when shaping  $\lambda$ -system evolution to gain as much self-organization as possible, since it is “for free”. Through relatively small modification of system components and their interaction, we can achieve a great degree of organization and regularity in the system.

**Path dependency** The informal conception of path dependency is that history matters (Buchanan, 2000): an accidental choice in the past determines the future path. A more formal conception is that there is a reinforcing effect; institutions, for example, are self-reinforcing. In economics the term path dependency is also known (Economides, 1996). Production and consumption decisions are based on the size of installed bases and on expectations of their increases over time. Path dependency in economics lasts because of the high switching costs involved. This is the case, for example, with our current infrastructures and industrial systems. We cannot change to an all-hydrogen economy overnight. Path dependency can also be found in politics. Once a political party chooses a statement, it is difficult to change; only when a new leader comes or elections are held can the statements be changed. Another form of path dependency, this time from management science, is group think (Janis, 1982). This is a situation in which the perspectives within a group are so aligned that deviation from the chain of thought is not possible. Teisman (Teisman, 2005) speaks of path dependency and bifurcation. Teisman assumes that human agents walk down a fixed path. Path dependency then means that a human agent’s behavior is based on the position of the agent and that this position is based on the path taken earlier. This could result in a lock-in situation when there is no switching point. However, when the environment changes, new opportunities appear and old goals become less important; this is called bifurcation. Human agents do then have the possibility to change their direction. Dependent on the internal mechanisms, the human agent will indeed choose a new direction. Both notions of path dependency will be used in this work. In  $\lambda$ -systems history matters, and it reinforces and limits the possible future states of the system.

**Chaos and robustness** A characteristic of the system output is that there are areas to which the system output ‘wants’ to go. These are called attractors. The system output has structure (Holland, 1997). Usually the system output contains multiple attractors and the system itself is sensitive to initial parameter values (system inputs). Slight changes in parameter value may lead it to another attractor. This extreme sensitivity to initial parameter conditions is also called chaotic behavior.

A system is robust (Callaway et al., 2000) when it is close to or at an attractor. Changes in certain parameters can not cause a deviation from the path to the attractor. Robustness is a relative concept, as some (large) parameter change cannot make the system deviate from its path to an attractor, whereas other slight changes might. The system is therefore robust against changes in specific parameters. It is a measure of how the system performs under stress when it is confronted with extreme inputs or with shocks from the environment for particular variables. Crash zones in cars enhance a car’s robustness and help save lives. The Internet is another example of a robust system. It is designed to function even if large parts of it are destroyed. Body temperature is relatively insensitive for large changes in the temperature of the environment. The economic situation of lock-in, in which a customer is so dependent on a supplier for products and services that he or she cannot move to another supplier without substantial switching costs, real and/or perceived, is another, albeit less positive, example.

Robustness can also be found in policy networks: if a conflict between parties escalates, the network structure (the ruling parties) should not immediately break down, and these networks should be able to recover from damage caused by conflicts (Kickert et al., 1997). This combination of chaos and robustness are important when attempting to shape the direction of  $\lambda$ -system evolution. Large intentional changes can have little or no effect, while small, accidental and unintended ones can dramatically affect the system.

**Instability** Instability is the capability to suddenly change over to another attractor with minimal parameter changes. Instability can be seen as the opposite of robustness. Note that robustness is not the same as stability, as systems can be simultaneously robust and instable. The terms stability and instability, however, make the properties sound mutually exclusive. The extreme sensitivity of a complex system to parameter values can be illustrated with the example of the exploded space shuttle Challenger. A difference of 13 ° Celsius at takeoff caused the Challenger to explode. The rubber sealing rings of the fuel tank cracked at this temperature. The management authorized the launch, even through NASA had the technical knowledge to be able to predict that this would happen. The human heartbeat is instable; for instance, the heartbeat can rapidly increase after hearing a sudden noise. Large crowds can also be instable, as they might suddenly erupt in riots under certain conditions. In structural engineering, a structure can become instable when excessive load is applied. Beyond a certain threshold, structural deflections magnify stresses, which in turn increase deflections. Examples of robust systems that become instable are: the human body, in which a tumor can become life threatening when it grows and presses into other organs; or the mobile phone network that periodically overloads on New Year's Eve. Just as is the case with robustness, instability of the system must be thoroughly understood before any change is effected in a  $\lambda$ -system.

## APPENDIX B

## CVI ONTOLOGY

Table [B.1](#) presents the entire CVI ontology structure and its scoring guide. 'Lobbyability' is a measure of how effective government lobbying is. 'Phaseability' is the ability to develop a technology in phases instead of at once.

Table B.1: CVI Ontology.

Aspect	Low score (1)	High score (5)
Safety landscape		
Kinetic energy	Much	None
Heat, smoke and/or heat radiation	Much	None
Hazardous materials	Much	None
Water (flooding, drowning, etc.)	Much	None
Electricity and/or electromagnetic signals	Much	None
Evacuability	Very difficult	Not an issue
Sense of safety	Very negative	Unaffected
Legal and Organizational		
Ownership of infra	Fragmented	Single owner
Sector regulation	Very specific	Absent
Competition on infra	Complete	None
Permit requirements	Many	None
Procedure duration	Very long	Very short
'Lobbyability' of regulator	Impossible	Easy
Spatial		
Barrier forming / Fractioning	Very large	None
Space requirement	Very large	None
Downward stackability	Only ground level	Deep tunnel
Upward stackability	Only ground level	Freely stackable
Trajectory constraint	Straight line only	Free form
Technical		
'Phaseability'	Very limited	Very high
Function specificity	High	Low
Modifiability	Very low	Very high
Maintainability	Frequent & complicated	Rare & easy

## APPENDIX C

## CVI INFRASTRUCTURES

Infrastructures identified by the Combination Of Infrastructures social process.

Type	Number	Name
Verkeer en Vervoer	1	Lokale Weg
	2	Ontsluitingsweg
	3	Snelweg
	4	Wegtunnel
	5	Weg op brug
	6	Parkeerterrein
	7	Bushalte
	8	Bewegwijzering
	9	Spoor
	10	Geëlektrificeerd spoor
	11	Spoortunnel
	12	Spoorbrug
	13	Spoorwegsemlacement
	14	Waterweg
	15	Waterkering
	16	Havenbekken
	17	"Parkeren" binnenvaart
	18	Luchthaven
	19	Helicopterbasis
	20	Voetpad
	21	Voetgangersbrug
	22	Fietspad
	23	Fietsbrug
	24	Tramspoor
	25	Tramhalte
	26	Metrospoor

*Continued on next page*

Type	Number	Name
	27	Metrohalte
Utilities	28	E-trace hoogspanning 50-380 kV
	29	E-trace sleuven 50-380 kV
	30	E-trace distributie 10-25 kV
	31	E-trace 380V 3-fasen
	32	E-trace Gelijkstroom
	33	Aardgas hoofdnet
	34	Aardgas distributienet
	35	Warmtenet
	36	ICT-kabels
	37	ICT-draadloos
	38	Drinkwater
	39	Rioolstelsel
	40	Regenwaterafvoer
	41	Industriewaterafvoer
	42	Kabels&Leidingentunnel
	43	Groenvoorziening
Industrie	44	Buisleidingtrace
	45	Leidingentunnel
	46	Industrieel parkeerterrein

This appendix will describe the simulation engine setup used in the thesis from the CostaDue project onwards. The design consists of several modular components. These are the hardware and operating system, knowledge management, simulation software, and data and knowledge processing & analysis. We view the simulation engine as an integrated physical-logical experimental device, allowing us to explore the synthetic universe contained within it. The engine design is guided by the guiding principles and requirements defined in Chapter 5.

The structure of the simulation engine is presented in Figure D.1.

## D.1 Hardware and Operating System

The relevant requirements at this level are open source, organic growth and modularity.

### D.1.1 Hardware Requirements

There are three main requirements for the hardware. It should:

- be as powerful as possible;
- be stable under load; and
- be interchangeable.

**Powerful** Currently, the main way to increase the performance of a computer is to increase the number of processing units (CPU). Therefore, multiprocessor hardware is used as much as possible, in order to allow model runs and analyses to be made parallel. Currently, the AMD Opteron CPU family is the preferred choice due to its ccNUMA<sup>1</sup> architecture. This architecture allows for very rapid memory-CPU interaction as compared to non-NUMA architectures present in Intel CPUs.

---

<sup>1</sup>Cache-Coherent Non-Uniform Memory Access

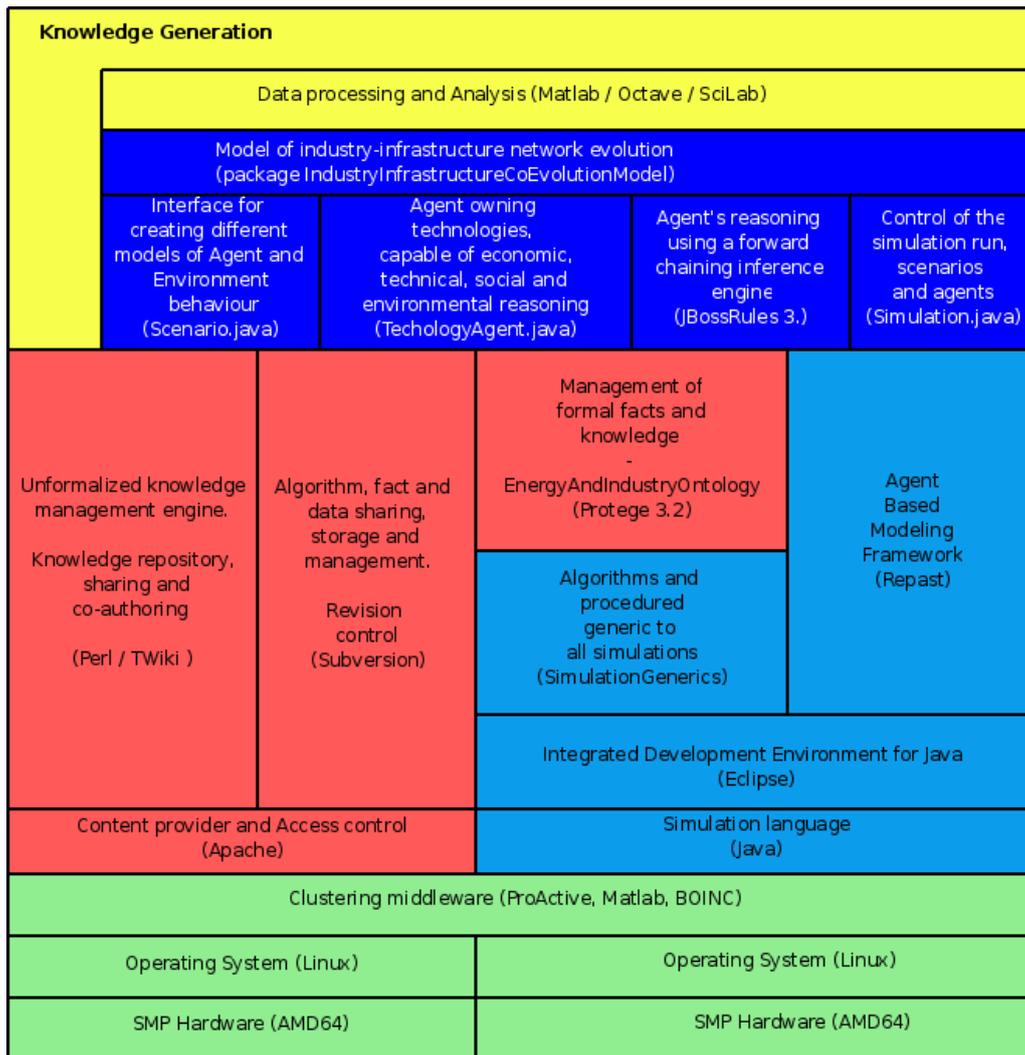


Figure D.1: The structure of the simulation engine. Green areas are the hardware and operating system. Red areas are the knowledge management components. Blue are the simulation software components, and yellow are the data and knowledge processing & analysis components.

**Stable** Hardware components of the simulation engine must be reliable under the severe system load experienced during computation. Therefore, commodity production grade servers are used. Such machines are designed to perform under high loads and have many redundant hardware features, such as hot-swap hard drives, hot-swap power supply units, etc.

**Interchangeable** The hardware platform should be interchangeable, since hardware is very quickly obsolete and the entire system must thus be designed with the idea that it will not run on the current hardware in the future. Designing a system based on industry standard, commodity hardware ensures this.

## D.1.2 Operating System Requirements

There are four main requirements that apply to the operating system. It should be as:

- stable as possible;
- secure as possible;
- scalable and flexible as possible; and
- interchangeable as possible.

**Stable and secure** This is an obvious requirement. No simulation is possible if the operating system constantly crashes or if it spends most of its time sending spam. Currently only UNIX-based systems offer these properties reliably.

**Scalable** The operating system needs to be able to deal with changing/expanding hardware without disrupting the simulation workflow after updates. It also must enable clustering of many physical machines into a single logical entity. The natural choice for scalable, flexible clusters is the GNU/Linux <sup>2</sup> operating system. An added advantage is that this is a modular, open source operating system that has excellent hardware support for the hardware needed for the task.

**Replaceable** Finally, the operating system should be replaceable. The simulation engine must be able to swap operating systems without too much effort. Using open standards and open source ensures that many alternatives exist for the same function.

## D.2 Knowledge Management

Based on the discussions on the SDM, it is clear that large amounts of knowledge are generated. This knowledge needs to be stored and managed. The knowledge management component of the simulation engine consists of several modular components (see the red areas in Figure D.1).

The knowledge management system needs to satisfy the following requirements (see Chapter 5). It must:

- provide enforceable authorship;
- provide an unchangeable historic record;
- be modular; and
- be open source.

### D.2.1 Apache

**Content provider** Apache<sup>3</sup> is the most widely used web (HTTP) server on the Internet today. It is a reliable, stable and modular system that provides content over the Internet and allows for easy extension. Apache provides the central entry point to both reading and writing of the knowledge repository over the Internet. In its default configuration, access to

---

<sup>2</sup><http://www.linux.org/>

<sup>3</sup><http://apache.org/>

content served by the Apache server is accessible to anyone on the Internet. The knowledge repository, however, needs to have enforceable authorship.

**Authentication** Enforceable authorship starts with enforceable identity. The web server maintains an identity database of all users allowed to access the knowledge repository. Authentication is based on Apache’s “Basic authentication”<sup>4</sup> method, based on a username and password combination. While this is a relatively weak authentication system, it is sufficient for our needs.

**Authorization** The system does not allow anonymous access. Once a user is authenticated, she is authorized to access all of the knowledge shared on the system that she has rights to. There are two types of user authorizations: permissive (everything except what is forbidden) and restrictive (nothing except what is permitted).

**Permissive** Users under permissive authorization are academic researchers. The permissive system allows access to everything except restricted areas where commercial projects knowledge is stored and for which non-disclosure agreements are required. This also means that any and all knowledge contributed by those who are willing to share is visible to everybody who is also willing to share.

**Restricted** Commercial users often require restricted access for their knowledge stored in the repository. Their access is restricted, meaning that they can access only their knowledge stored on the system and are thus excluded from the general knowledge pool. On the other hand, permissive users are not allowed to access their knowledge pool.

**Tit for tat** Users are treated with a tit for tat strategy (Axelrod, 1984). This strategy allows the users to feel comfortable about sharing their ideas, so that they do so at the maximum level. It ensures that users will receive the same treatment for their actions depending on which level of involvement they choose. Users who require secrecy receive that in exchange for reduced access of their own. Users who openly share are rewarded by others’ sharing. This way the knowledge repository keeps users aware of their rights and, because of authentication, responsible for their actions. Apache knows who did what when.

## D.2.2 Wiki

**What is a wiki** While Apache serves the content in the knowledge repository to users over the Internet, a system is needed to create, organize and manage that content. A description of what a wiki is is best left to Ward Cunningham, the creator of the first ever wiki, the Portland Pattern Repository<sup>5</sup>:

The simplest online database that could possibly work.

Wiki is a piece of server software that allows users to freely create and edit Web page content using any Web browser. Wiki supports hyperlinks and has a simple text syntax for creating new pages and cross links between internal pages on the fly.

---

<sup>4</sup><http://httpd.apache.org/docs/2.0/howto/auth.html#basic>

<sup>5</sup><http://c2.com/ppr/>

Wiki is unusual among group communication mechanisms in that it allows the organization of contributions to be edited in addition to the content itself.

Like many simple concepts, 'open editing' has some profound and subtle effects on Wiki usage. Allowing everyday users to create and edit any page in a Web site is exciting in that it encourages democratic use of the Web and promotes content composition by nontechnical users.

**TWiki** The wiki engine of choice is TWiki<sup>6</sup>. TWiki is a mature and feature-rich open source wiki engine. It is highly extensible through plug-ins and has a large and active developers' and users' community. TWiki is written in the Perl programming language. It runs under Apache and its internal Perl interpreter, `mod_perl`<sup>7</sup>. TWiki versions each change to a page via the RCS system<sup>8</sup>, providing an unchangeable historic record. Since each change to a page has been authenticated and authorized by Apache, a complete historic record of who did what when is created.

**Unformalized knowledge** Wiki is mainly used to store unformalized knowledge. This mainly means that text, images and other types of files can be stored and changed by anyone authorized to do so. The main property of this unformalized knowledge is the so-called *wabi-sabi*. It is defined by (Koren, 1994) as:

Wabi-sabi is the quintessential Japanese aesthetic. It is a beauty of things imperfect, impermanent, and incomplete. It is a beauty of things modest and humble. It is a beauty of things unconventional...

The unformalized knowledge stored on the wiki system is thus always changing and never complete. It enables and involves the community around it, and is as close to a repository of tacit knowledge as one can come in a written form.

**Formalized knowledge** In addition to the unformalized knowledge managed by the wiki, there is a need to manage formalized knowledge. There are two types of formalized knowledge that need managing: the "What Is" knowledge and the "How To" knowledge. The "What is" knowledge is maintained in a formal ontology, the use of which was previously discussed. The formal "How to" knowledge consists of computer algorithms that describe how things are to be done. This knowledge is stored in a source code versioning system, Subversion.

### D.2.3 Subversion

The best description of Subversion is given by its makers<sup>9</sup>:

Subversion is a free/open-source version control system. That is, Subversion manages files and directories, and the changes made to them, over time. This allows you to recover older versions of your data, or examine the history of how

---

<sup>6</sup><http://twiki.org/>

<sup>7</sup><http://perl.apache.org/>

<sup>8</sup><http://www.gnu.org/software/rcs/>

<sup>9</sup><http://svnbook.red-bean.com/en/1.4/svn.intro.what-is.html>

your data changed. In this regard, many people think of a version control system as a sort of 'time machine'.

Subversion can operate across networks, which allows it to be used by people on different computers. At some level, the ability for various people to modify and manage the same set of data from their respective locations fosters collaboration. Progress can occur more quickly without a single conduit through which all modifications must occur. And because the work is versioned, you need not fear that quality is the trade-off for losing that conduit - if some incorrect change is made to the data, just undo that change.

Subversion allows for the free experimentation with computer code and removes the fear of breaking things, as there is always the previous version. It is implemented as Apache module `mod_svn` and uses Apache's authentication and authorization system.

## D.2.4 Ontology

The formalized facts, or "What is" knowledge, is stored and managed through the Protegé, a free, open source ontology editor and knowledge-based framework <sup>10</sup>. It is the current *de facto* standard for ontology development. Protegé itself uses a XML database to store the classes and instances. The database files are stored in Subversion to ensure historic and authorship record of all changes to it.

## D.3 Simulation Software

Simulation Software components of the simulation engine are presented in blue areas in Figure D.1. The light blue areas are generic for all models developed on the simulation engine. The dark blue components are specific to the CostaDue model. Just as any other component, the simulation software needs to conform with the requirements presented in Chapter 5. I will first present the generic components before discussing the specific ones.

### D.3.1 Java

**Object Oriented** Java is a high-level Object Oriented programming language, initially developed by the Sun corporation and recently open sourced under the Gnu General Public License. One of the main features of Java is that its source code is not compiled to platform specific machine code, as other computer languages are, but to a higher level "byte code" that runs inside a Java Virtual Machine (JVM). JVMs are available on almost any imaginable hardware platform.

**Widely used** Java is widely used in the scientific community, and there is great availability of open source libraries. Java has high performance, even though the folklore claims otherwise <sup>11</sup>. It is a relatively high-level language. It is nonetheless relatively easy to use, since it abstracts away most low-level programming tasks, such as memory management.

---

<sup>10</sup><http://protege.stanford.edu/>

<sup>11</sup>[http://en.wikipedia.org/wiki/Java\\_performance](http://en.wikipedia.org/wiki/Java_performance)

### D.3.2 Eclipse

**Programming** Writing, debugging, compiling and running Java programs is a relatively complex tasks that has been greatly automated by the development of Integrated Development Environments, or IDEs. IDEs are a great help when less experienced programmers (such as PhD candidates and MSc students) are involved in model development. The most advanced open source IDE available for Java is Eclipse <sup>12</sup>. It is available on all major operating systems and has a large and active developers' community.

**Made easier** Eclipse has an integrated build infrastructure, compiling programs on the fly and detecting programming errors very quickly. It allows easy code factoring and change. It is tightly integrated with Subversion for easy management of code versions and the required libraries. Numerous plug-ins exist to further simplify common tasks and free the programmer to focus on the important model logic.

### D.3.3 Repast

The Recursive Porous Agent Simulation Toolkit (Repast) (N. Collier and North, 2003) is used as the basic Agent Based Model platform. It contains the agents, schedules their interaction, does all the data collection, facilitates parameter sweeps, etc. Repast can be seen as a meta-Agent Based Model that creates, runs and manages other Agent Based Model s.

### D.3.4 Simulation Generics

**Modularity and reusability** One of the important lessons from the learning cases presented in Chapter 6 is the need for modularity. Modularity reduces the time needed to develop a model and reduces the number of potential errors in models, since it reuses existing and tested code. This notion led to the development of Simulation Generics <sup>13</sup>, a collection of model components that can be reused in any model developed using the ontology.

**Specific vs. generic** Whenever a model is developed, especially when evolving from an earlier model, new functionalities are required - for example, a new type of graph. These improvements can be made specific to the model developed. These new features work well with the model and are relatively quick to implement. They are, however, not generic or modular and cannot be reused on other models. The challenge for the modeller is to recognize which components are interesting and/or useful enough to spend extra effort on in order to create a generic component. When modelling in groups, this becomes easier to decide, as the modeller is very likely to have used modules by others and it is more natural to then build components that are in turn useful to others.

---

<sup>12</sup><http://www.eclipse.org/>

<sup>13</sup><http://gux.tudelft.nl/svn/SimulationGenerics/trunk/src/>



LHS is usually applied to problems of up to 10 variables and a few hundred experiments. For the application in this thesis, we stretched the method to tens of parameters and tens of thousands of samples. This presented an unexpected computational challenge. When creating a LHS matrix for 10,000 samples over 30 dimensions, Matlab takes 7 minutes on a fast server with 2.6 GB of free memory. The memory requirement grows exponentially with the number of sample sizes. Machines equipped with 10 GB of memory can only generate 20,000 samples. If we want to have more samples, an alternative would be to use a random sample across the parameter space. The random sampling process of the same size takes 0.025 seconds and practically no memory space. This brings up the question of why we should use LHS at all. Figure [E.1](#) demonstrates the difference between samples generated by a random sampling and those generated by LHS. The histogram presents the frequencies of samples falling within a range of values. We can see that the LHS samples are very uniformly distributed across the parameter range, whereas the random samples vary greatly.

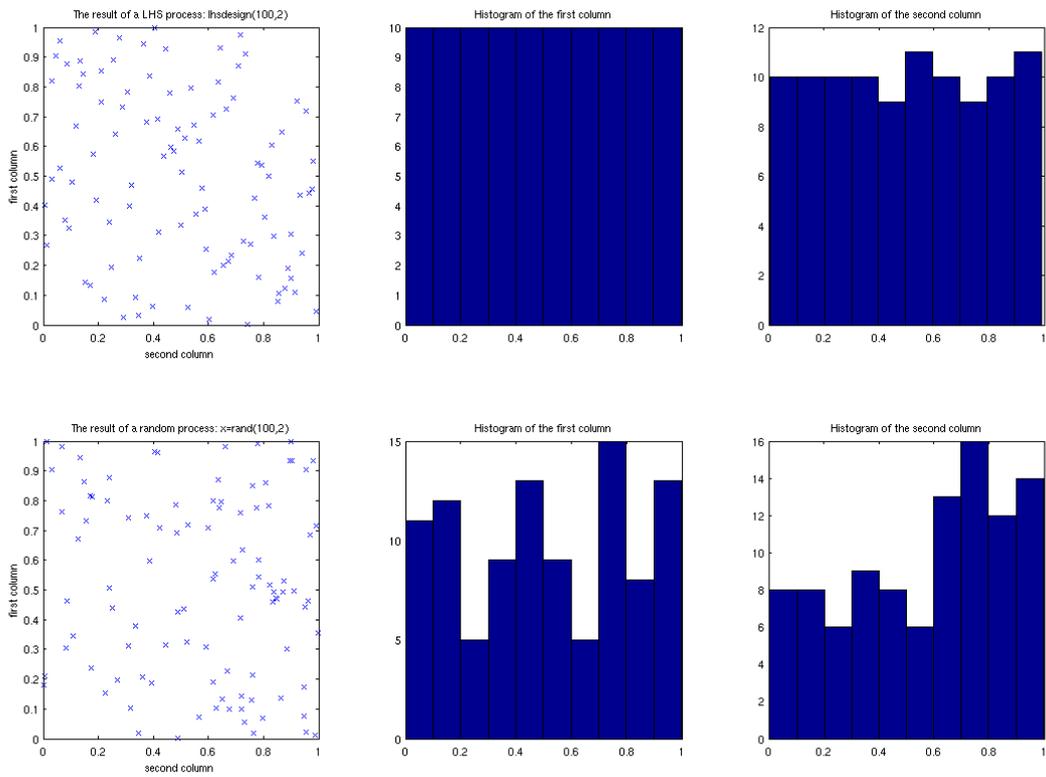


Figure E.1: Latin Hypercube Sampling (top row) vs Random Sampling (bottom row)

# APPENDIX F

## REQUIREMENTS TABLE

Table F.1: Overview of the case study performances

Requirement	Score	Explanation
<b>Flow Based Evolution</b>		
Non-Functional		
Open Source	Partly	A the time of writing the source code is available to peers, but not to the general public.
Sufficient community diversity	No	The model is developed as a first learning case, with the author as the only stakeholder.
Organically growing	No	The model is built as a conceptual test. While extension is kept in mind the current technical design does not allow for easy extension.
Recorded history	Yes	Versioning is initiated using CVS system. Due to technical problems the models early history is lost. More robust versioning needed.
Enforceable authorship	Yes	Personal accounts are used to track code commits.
Modular	No	As the model was a technology test, no modularity is implemented at this point.
Outcome		
Useful	Yes	The model is useful at the meta-level to the modeler.
Testable	Yes	During model development, model was not versioned, so repeatable testing of intermediate states is not possible. Experiments performed at the time of thesis writing are properly versioned, and can be repeated and tested.
<b>Combination of Infrastructures</b>		
Non-Functional		
Continued on next page		

**Table F.1 – continued from previous page**

Open Source	Partly	The source code of the calculation method is available to peers, but not to the general public. Social process has been published.
Sufficient community diversity	Yes	Expert and user diversity was sufficient. Quality of community was lacking.
Organically growing	Partly	Project requires external parties to proceed.
Recorded history	No	Attempted and failed.
Enforceable authorship	No	Attempted and failed.
Modular	Partly	The description of infrastructures and their landscapes and aspects is probably portable to other projects.
Outcome		
Useful	Yes	Useful both at the process level to the modeler, as well at the case level to the problem owner.
Testable	Yes	Social experiments are possible, observation of shifting insights and knowledge directly readable from the fitness landscapes.
<b>Chocolate Game Model</b>		
Non-Functional		
Open Source	Yes	All computercode, game design deliberations and materials and the ontology are available to the involved social network
Sufficient community diversity	No	Only students and staff from TU Delft involved in ontology, game and model creation.
Organically growing	Yes	Ontology, game and the model are direct conceptual descendents of the first case study, with generic modelling insights from the second case applied.
Recorded history	Yes	The entire modelling process, both formal interaction via code and ontology and informal interaction via wiki are versioned.
Enforceable authorship	Yes	Via authentication in subversion and wiki.
Modular	Yes	Ontology, agents and decision algorithms are designed with modularity and reuse in mind.
Outcome		
Useful	Yes	Provided plenty insights into ontology development and modelling process. Developed a usable system decomposition method.
Testable	Yes	The SDM can be repeated and results compared. Model is repeatable due to versioning and random seed record.
Continued on next page		

**Table F.1 – continued from previous page**

<b>Costa Due</b>		
Non-Functional		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	Yes	A wide range of domain experts and stakeholders were involved. While this is good a even greater diversity would contributed to better insights.
Organically growing	Yes	The SDM is based on the previously developed approach. The simulation engine was designed with insights obtained from the learning cases but did not include any computer code. It will serve as a basis for future organic growth.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	The Simulation engine is designed to be modular, and Simulation Generic are extensively used to encourage reuse of computer code.
Outcome		
Useful	Yes	The model has yielded both useful case level insights as well as provided a good bases for further method development. It can be considered a resounding success.
Testable	Yes	More than any other model so far, every component can be examined and tested because of the use of versioning.
<b>Bulk Biochemicals</b>		
Non-Functional		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	No	The case study is performed as a technological development step and no stakeholder was involved.
Organically growing	Yes	The case study is entirely based on the previous one.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful	Yes	Useful as a proof of principle for both LHS environmental mapping and in implementing MCA.
Testable	Yes	All outcomes are fully testable.
<b>Metals Network</b>		
Continued on next page		

**Table F.1 – continued from previous page**

Non-Functional		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	No	The case study is performed as a technological development step and no stakeholder was involved.
Organically growing	Yes	The case study is entirely based on the previous one.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful		Useful as a proof of principle for both encoding metal processing knowledge refinement of economic reasoning.
Testable	Yes	All outcomes are fully testable.
Bioelectricity		
Non-Functional		
Open Source	Yes	All unformalized and formal knowledge is accessible to all involved parties.
Sufficient community diversity	Partial	The case study is performed as an extension the work for the government committee. Only the scientific staff was involved in the case study.
Organically growing	Yes	The case study is based on the previous one and extends it organically.
Unchangeable history	Yes	Both the formalized and unformalized knowledge are fully versioned.
Enforceable authorship	Yes	All formalized and unformalized contributions have full authorship records.
Modular	Yes	Built on existing models. Additions created as modules.
Outcome		
Useful	Yes	Greatly increases the insight into environmental impacts of a simulated network and provides domain specific insights into bioelectricity production.
Testable	Yes	All outcomes are fully testable.
End of Table F.1		

APPENDIX G

OVERVIEW OF THE REVIEWED LITERATURE

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Bergholtz et al., 2003b)	Modelling institutional, communicative and physical domains in agent oriented information systems	Business, Knowledge Engineering	No	No	No	Yes	No
(Calmet et al., 2003)	A liberal approach to openness in societies of agents	AI, Sociology	No	No	No	Yes	No
(Noy and Schroeder, 2003)	Advancing profile use in agent societies	AI, Sociology	No	No	No	Yes	No
(Lanzola et al., 1999)	A framework for building cooperative software agents in medical applications	AI	No	Yes	No	No	No
(Hadeli et al., 2004)	Multi-agent coordination and control using stigmergy	Process Technology	Yes	Yes	No	Yes	No
(Tilman et al., 1999)	Modelling the actors in water supply systems	Water management	No	Yes	Yes	No	No
(Bresciani and Donzelli, 2004)	A practical agent-based approach to requirements engineering for socio-technical systems	AI, Requirements engineering	No	No	No	Yes	Yes
(Donzelli and Bresciani, 2004)	Improving requirements engineering by quality modelling - a quality-based requirements engineering framework	AI, Requirements engineering	No	No	No	Yes	Yes
(Kendall, 1999)	Role models - Patterns of agent system analysis and design	AI, Business	No	Yes	No	No	No
(Binbasioglu and Winston, 2003)	Knowledge-based management support: an application to the IS change agent role problem	Expert systems	No	No	No	No	No
(Clocksin, 2003)	Artificial intelligence and the future	AI	No	No	No	No	No
(Schneider et al., 2000)	Adaptation: Sensitivity to natural variability, agent assumptions and dynamic climate changes	Agriculture	No	Yes	Yes	No	No
(Giordano et al., 2003)	Specifying and verifying systems of communicating agents in a temporal action logic	AI	No	No	No	Yes	No
(Vishwanath and Goldhaber, 2003)	An examination of the factors contributing to adoption decisions among late-diffused technology products	Technology diffusion	No	No	No	No	No
(Tanimoto and Fujii, 2003)	A study on research performance in Japanese Universities: Which is more efficient - A professor who is leading his research group or one who is working alone? The multi-agent simulation knows	Academic society	No	Yes	No	No	No
(Bergholtz et al., 2003a)	Reconciling physical, communicative, and social/institutional domains in agent oriented information systems - A unified framework	Business, Knowledge Engineering	No	No	No	Yes	No
(Robbins and Robbins, 2004)	Simulation of the population dynamics and social structure of the Virunga mountain gorillas	Ecology, Biology	No	Yes	No	No	No
(Singh et al., 2004)	Toward understanding consumers' role in medical decisions for emerging treatments: Issues, framework and hypotheses	Medicine, Business	No	No	No	Yes	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Zhang, 2004)	Residential segregation in an all-integrationist world	Sociology	No	Yes	No	No	No
(Morishima et al., 2004)	The role of affect and sociality in the agent-based collaborative learning system	AI	No	No	No	Yes	No
(Carpenter and Brock, 2004)	Spatial complexity, resilience, and policy diversity: Fishing on lake-rich landscapes	Ecology, Policy	No	Yes	No	No	No
(Martino et al., 2004)	Statistical mechanics analysis of the equilibria of linear economies	Economy, statistics	No	No	No	No	No
(Lachman, 1998)	AI, decision science, and psychological theory in decisions about people - A case study in jury selection	AI, Sociology	No	Yes	No	No	No
(Dumont and Hill, 2004)	Spatially explicit models of group foraging by herbivores: what can Agent-Based Models offer?	Ecology, Biology	No	Yes	No	No	No
(Barreteau et al., 2004)	Suitability of Multi-Agent Simulations to study irrigated system viability: application to case studies in the Senegal River valley	Water management	No	Yes	No	No	No
(Fuentes et al., 2004)	Activity theory for the analysis and design of multi-agent systems	AI, Sociology	No	Yes	No	Yes	No
(Rodriguez-Achach et al., 2004)	Diffusion of innovations in a social network under mixed Pareto-Nash strategies	Economics	No	Yes	No	No	No
(Campos and Hill, 1998)	An agent-based framework for visual-interactive ecosystem simulations	ABM, Ecology	No	Yes	No	Yes	No
(Lee, 2004)	Applying domain knowledge and social information to product analysis and recommendations: an agent-based decision support system	AI	No	No	No	No	No
(Grandmont, 1998)	Expectations formation and stability of large socioeconomic systems	Economics	No	Yes	Yes	No	No
(Charterjee and Xu, 2004)	Technology diffusion by learning from neighbours	Technology diffusion	Yes	No	No	No	No
(Choe and Fraser, 1998)	A note on imperfect monitoring of agricultural environmental policy	Agriculture	No	Yes	No	No	No
(Kaminka et al., 2002)	Monitoring teams by overhearing: A multi-agent plan-recognition approach	AI	No	Yes	No	No	No
(Lustick, 2002)	Ps-i: A user-friendly agent-based modelling platform for testing theories of political identity and political stability	Politicology	No	Yes	No	No	No
(K.M.Carly, 2002)	Computational organizational science: A new frontier	AI	No	Yes	No	Yes	No
(Inchiosa and Parker, 2002)	Overcoming design and development challenges in agent-based modelling using ascape	ABM	No	Yes	No	Yes	No
(Gandon et al., 2002)	Multi-agent corporate memory management system	AI	No	Yes	No	Yes	No
(Chaidraa, 2002)	Causal maps: Theory, implementation, and practical applications in multiagent environments	Knowledge Engineering, AI	No	Yes	No	Yes	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Nehaniv and Dautenhahn, 2001)	Like me ? - Measures of correspondence and imitation	AI	No	Yes	No	No	No
(Henrickson, 2002)	Old wine in a new wineskin - College choice, college access using agent-based modelling	Sociology	No	Yes	No	No	No
(Bergler, 2001)	Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis	Agriculture	No	Yes	Yes	No	No
(Webster and Wu, 2001)	Coarse, spatial pricing and self-organising cities	Land use and planning	No	Yes	No	No	No
(Persson et al., 2001)	Understanding socially intelligent agents - A multilayered phenomenon	AI, Social intelligence	No	No	No	Yes	No
(Jonker and Treur, 2001)	Agent-based simulation of animal behaviour	AI	No	Yes	No	Yes	No
(Yu, 2001)	Agent orientation as a modelling paradigm	AI	No	No	No	No	No
(Zhong et al., 2001)	Dynamically organizing KDD processes	AI, Data mining	No	No	No	Yes	No
(Gasser, 2001)	Perspectives on organizations in multi-agent systems	AI	No	Yes	No	Yes	No
(Petta and Trapp, 2001)	Emotions and agents	AI	No	No	No	No	No
(Baxter et al., 2003)	Agent-based modelling - intelligent customer relationship management	CRM	No	Yes	No	No	No
(Sallach, 2003)	Social theory and agent architectures - Prospective issues in rapid-discovery social science	Sociology	No	No	No	No	No
(Lee, 2003)	Ijade surveillant - an intelligent multi-resolution composite neuro-oscillatory agent-based surveillance system	Business, AI	No	Yes	No	No	No
(Ross et al., 2002)	Speaking autonomous intelligent devices	AI	No	Yes	No	No	No
(Iizuka et al., 2000)	Learning of virtual words utilized in negotiation process between agent	AI	No	Yes	No	Yes	No
(Yaskawa and Sakata, 2003)	The application of intelligent agent technology to stimulation	ABM	No	Yes	No	Yes	No
(Weigand et al., 2003)	Specification by refinement and agreement: Designing agent interaction using landmarks and contracts	AI	No	Yes	No	Yes	No
(Ryokai et al., 2003)	Virtual peers as partners in storytelling and literacy learning	AI	No	Yes	No	No	No
(Filipe, 2000)	An organizational semiotics model for multi-agent systems design	AI	No	Yes	No	No	No
(Rossetti et al., 2002)	Using BDI agents to improve driver modelling in a commuter scenario	Traffic management	No	Yes	No	No	No
(Stathis et al., 2002)	Living memory: agent-based information management for connected local communities	Information Systems	No	Yes	No	No	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Faratin et al., 2002)	Using similarity criteria to make issue trade-offs in automated negotiations	AI	No	Yes	No	Yes	No
(Osowski and Omicini, 2002)	Coordination knowledge engineering	AI	No	No	No	Yes	No
(Pechoucek et al., 2002)	Coalition planning multi-agent system for humanitarian relief operations	AI	No	Yes	No	No	No
(Hercog and Fogarty, 2002)	Social simulation using a multi-agent model based on classifier systems: The emergence of vacillating behaviour in the "El farol" Bar problem	AI, ABM	Yes	Yes	No	No	No
(Andrasik and Krem-pasky, 2002)	Emergence of deterministic chaos: Limited passage in the evolution of economy	Economics	Yes	Yes	No	No	No
(Divina and Vogt, 2006)	A hybrid model for learning word-meaning mappings	AI	No	Yes	No	Yes	No
(Sansores and Pavon, 2006)	Agent based simulation for social systems: From modelling to implementation	AI	Yes	Yes	No	Yes	No
(Manson, 2006)	Land use in the southern Yucatan peninsula region of Mexico: Scenarios of population and institutional change	Land use	No	Yes	No	No	No
(Klein and Giese, 2006b)	Analysis and design of physical and social contexts in multi-agent systems	AI	No	No	No	No	No
(Guyot and Drogoul, 2005)	Multi-agent based participatory simulations on various scales	ABM	No	Yes	No	Yes	No
(Rotundo and Ausloos, 2007)	Microeconomic co-evolution model for financial technical analysis signals	ACE, Economics	No	Yes	No	No	No
(Serenko, 2007)	The development of an instrument to measure the degree of animation predisposition of agent users	Psychology	No	No	No	No	No
(Bandini et al., 2006)	Crowd modelling and simulation - The role of multi-agent simulation in design support systems	Architecture, Urban planning	No	Yes	No	No	No
(Moss and Norling, 2006)	Multi-agent-based simulation: Why bother?	AI	No	No	No	No	No
(Bharwani, 2006)	Understanding complex behavior and decision making using ethnographic Knowledge Elicitation Tools (knets)	Ethnography, Knowledge Engineering	No	No	No	No	No
(Franch et al., 2007)	Systematic construction of i* strategic dependency models for socio-technical systems	Knowledge Engineering, AI	No	No	No	No	No
(Rong and Liu, 2006)	A multi-agent architecture for CSCW systems: From organizational semiotics perspective	AI	No	No	No	No	No
(Santos et al., 2006)	Semantic web technologies applied to interoperability on an educational portal	AI, Ontology,	No	No	No	Yes	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Huang et al., 2006)	Developing an agent-based workflow management system for collaborative product design	AI, Ontology,	No	No	No	Yes	No
(Omicini et al., 2006)	Coordination artifacts as first-class abstractions for MAS engineering: State of the research	AI	No	No	No	No	No
(Eccies and Groth, 2007)	Wolves, bees, and football: Enhancing coordination in sociotechnological problem solving systems through the study of human and animal groups	Socio-Technical, Biology	No	No	No	Yes	No
(Fathian et al., 2007)	Application of honey-bee mating optimization algorithm on clustering	Data Mining, Biology	No	No	No	Yes	No
(Maher et al., 2007)	Agents for multidisciplinary design in virtual worlds	AI, Design	No	No	No	Yes	Yes
(van der Hoek et al., 2007)	Social laws in alternating time: effectiveness, feasibility, and synthesis	AI	No	No	No	Yes	No
(Lynam et al., 2002)	Adapting science to adaptive managers: Spherograms, belief models, and multi-agent systems modelling	Participatory modelling, Land use	No	Yes	No	No	Yes
(D'Aquino et al., 2003)	Using self-designed role-playing games and a multi-agent system to empower a local decision-making process for land use management: The SelfCorreas experiment in senegal	Participatory modelling	No	Yes	No	No	Yes
(Cheng et al., 2003)	Design and planning under uncertainty: issues on problem formulation and solution	Process Development	No	No	No	No	No
(Binder, 2007)	From material flow analysis to material flow management Part II: the role of structural agent analysis	IE, MFA, Social systems	No	No	No	No	No
(Smith and Conrey, 2007)	Agent-based modelling: A new approach for theory building in social psychology	Social Psychology	Yes	No	No	No	No
(Boisot et al., 2007)	Property rights and information flows: a simulation approach	Economics, Intellectual Property Rights	No	Yes	No	No	No
(Yamaguchi et al., 2007)	New model based on cellular automata and multiagent techniques	AI, CA	Yes	Yes	No	Yes	No
(Wanyama and Far, 2007)	A protocol for multi-agent negotiation in a group-choice decision making process	AI, Negotiation	No	No	No	Yes	No
(Jiang et al., 2007)	Agent alliance formation using ART-networks as agent belief models	AI, Ontology,	No	No	No	Yes	No
(Purnomo et al., 2005)	Developing multi-stakeholder forest management scenarios: a multi-agent system simulation approach applied in Indonesia	ABM, forestry management	No	Yes	No	No	No
(Alkemaded et al., 2007)	On social learning and robust evolutionary algorithm design in the Cournot oligopoly game	ACE, Economics	Yes	No	No	No	No
(Mantovani, 1996)	Social context in HCI: A new framework for mental models, cooperation, and communication	AI	No	No	No	No	No
(Reed et al., 2005)	Multi-agent patient representation in primary care	AI, Healthcare	No	Yes	Yes	No	No
(Carley, 1996)	Artificial intelligence within sociology	Sociology, AI	No	No	No	No	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Herrera, 2005)	Stochastic bycatch, informational asymmetry, and discarding	Ecology	No	Yes	No	No	No
(Kofod-Petersen and Cassens, 2005)	Using activity theory to model context awareness	AI	No	No	No	Yes	No
(Baark, 1994)	Technological entrepreneurship and commercialization of research results in the west and in china - comparative perspectives	Technology diffusion	No	No	No	No	No
(Fullam et al., 2005)	A design foundation for a trust-modelling experimental testbed	AI	No	No	No	No	No
(Rutkowski and Walle, 2005)	Cultural dimensions and prototypical criteria for multi-criteria decision support in electronic markets: A comparative analysis of two job markets	AI, Group decision	No	Yes	No	No	No
(Razavi et al., 2005)	Adaptive modelling: An approach and a method for implementing adaptive agents	Software Engineering, AI	No	Yes	No	Yes	No
(Lee et al., 1997)	Discovery and representation of causal relationships in MIS research: A methodological framework	Knowledge Engineering, AI	No	No	No	Yes	No
(Pezzulo and Calvi, 2005)	Designing and implementing MABS in akira	AI, ABM	No	Yes	No	Yes	No
(Shah and Pritchett, 2005)	Work-environment analysis: environment centric multi-agent simulation for design of socio-technical systems	Socio-Technical	No	Yes	No	No	No
(Upal, 2005)	Simulating the emergence of new religious movements	Sociology	No	Yes	No	No	No
(Ricci et al., 2005)	Environment-based coordination through coordination artifacts	AI	No	No	No	Yes	No
(Garcia-Flores, 2005)	Uses of agent-based modelling in innovation/new product development research	Innovation management	Yes	Yes	No	No	No
(Gaines and Norrie, 1997)	Coordinating societies of research agents - IMS experience	CAD/CAM	No	Yes	No	No	No
(Sabater and Sierra, 2005)	Review on computational trust and reputation models	AI, Trust	No	No	No	No	No
(Guyot et al., 2007)	Multi-agent participatory simulations between experimental economics and role-playing games	Participatory modelling	No	No	No	Yes	Yes
(Klein and Giese, 2006a)	Grounding social interactions in the environment	AI	No	Yes	No	Yes	No
(Torrens and Benson, 2005)	Geographic automata systems	GIS, cellular automata	No	No	No	No	No
(Shang et al., 2005)	An agent-based soft computing society with application in the management of establishment of Hydraulic Fracture in Oil field	AI, Engineering	No	Yes	No	Yes	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Ocampo and Munch, 2007)	The REMIS approach for rationale-driven process model evolution	Process evolution	No	Yes	No	Yes	No
(Troisi et al., 2005)	An agent-based approach for modelling molecular self-organization	Chemistry	Yes	Yes	No	No	No
(Bentahar et al., 2006)	A tableau method for verifying dialogue game protocols for agent communication	AI	No	No	No	Yes	No
(Castella and Verburg, 2007)	Combination of process-oriented and pattern-oriented models of land-use change in a mountain area of vietnam	Land use	No	Yes	Yes	No	Yes
(Westerberg et al., 1997)	Designing the process design process	Process Development	No	No	No	Yes	Yes
(Krohn and van den Daele, 1998)	Science as an agent of change: finalization and experimental implementation	Sociology	No	Yes	No	No	Yes
(Chaidraa, 1997)	Causal reasoning in multi-agent systems	AI	No	No	No	Yes	No
(Castella et al., 2007)	Combining top-down and, bottom-up modelling approaches of land use/cover change to support public policies: Application to sustainable management of natural resources in northern vietnam	Land use	No	Yes	Yes	No	Yes
(Lopez-Parades et al., 2005)	Urban water management with artificial societies of agents: The FIRMABAR simulator	ABM, Participatory modelling	No	Yes	No	No	Yes
(Sutcliffe, 2000)	Requirements analysis for socio-technical system design	Knowledge Engineering, AI	No	Yes	Yes	No	No
(Bharwani et al., 2005)	Multi-agent modelling of climate outlooks and food security on a community garden scheme in Limpopo, South africa	Agriculture, Sociology	No	Yes	No	No	Yes
(Boero and Squazzoni, 2005)	Does empirical embeddedness matter? Methodological issues on agent-based models for analytical social science	Sociology	No	No	No	Yes	No
(Lu and Botha, 2006)	Process development: a theoretical framework	Process Development	No	No	No	Yes	No
(Panbiango and Pahl-Wostl, 2006)	Modelling socio-technical transformations in wastewater treatment - A methodological proposal	Water management	No	Yes	Yes	No	Yes
(Bousquet et al., 2007)	Using multi-agent systems in a companion modelling approach for agro-ecosystem management in South-east asia	Agriculture, Landuse	No	Yes	Yes	No	Yes
(Pan and Tenenbaum, 1991)	An intelligent agent framework for enterprise integration	AI	No	Yes	No	No	No
(Kun et al., 2007)	Cellular automata for the spreading of technologies in socio-economic systems	Socio-Technical	Yes	Yes	Yes	No	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Doherty et al., 2006)	A re-conceptualization of the interpretive flexibility of information technologies: re-addressing the balance between the social and the technical	Information Systems	No	No	Yes	No	Yes
(Fuentes et al., 2006)	Requirements elicitation for agent-based applications	Knowledge Engineering, AI	No	No	No	No	Yes
(Carrillo-Hermosilla, 2006)	A policy approach to the environmental impacts of technological lock-in	Ecological Economics	Yes	Yes	No	No	No
(Carley et al., 2006)	Biowar: Scalable agent-based model of bioattacks	Biosafety, Terrorism	Yes	Yes	Yes	Yes	No
(Hill et al., 2006)	Improving ROSE with an enriched modelling framework	Knowledge Engineering, AI	No	No	No	No	Yes
(Manson, 2005)	Agent-based modelling and genetic programming for modelling land change in the Southern Yucatan Peninsular Region of Mexico	Land use	No	Yes	Yes	Yes	No
(Fehler et al., 2005)	Techniques for analysis and calibration of multi-agent simulations	AI	No	No	No	Yes	No
(Penker and Wytzens, 2005)	Scenarios for the Austrian food chain in 2020 and its landscape impacts	Urban planning	No	No	Yes	No	Yes
(Pahl-Wostl, 2005)	Information, public empowerment, and the management of urban watersheds	LSTS evolution	Yes	No	Yes	No	Yes
(Berger et al., 2007)	Capturing the complexity of water uses and water users within a multi-agent framework	Water management	No	Yes	Yes	No	Yes
(Croke et al., 2007)	Integrated assessment of water resources: Australian experiences	IA, Water Management	No	No	Yes	No	Yes
(Bryson et al., 2007)	Agent-based modelling as scientific method: a case study analysing primate social behaviour	Biology, Meta analysis	Yes	No	No	No	No
(Tesfatsion, 2007)	Agents come to bits: Towards a constructive comprehensive taxonomy of economic entities	ACE, Economics	Yes	Yes	No	No	No
(Doherty et al., 2006)	A re-conceptualization of the interpretive flexibility of information technologies: re-addressing the balance between the social and the technical	Psychology	No	No	Yes	No	Yes
(Keirstead, 2006)	Evaluating the applicability of integrated domestic energy consumption frameworks in the UK	Domestic energy policy	Yes	No	Yes	Yes	Yes
(Panebianco and Pahl-Wostl, 2006)	Modelling socio-technical transformations in wastewater treatment - A methodological proposal	Wastewater	Yes	Yes	Yes	No	No
(Janssen and Ostrom, 2006)	Empirically based, agent-based models	ABM, Ecology	Yes	No	No	No	Yes
(Berger and Schreinemachers, 2006)	Creating agents and landscapes for multi-agent systems from random samples	ABM, Ecology	Yes	Yes	Yes	No	No
(Ottens et al., 2004)	Modelling engineering systems as socio-technical systems	Philosophy	Yes	No	Yes	No	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Cysneiros and Yu, 2003)	Requirements engineering for large-scale multi-agent systems	MAS, Requirements engineering	No	Yes	No	Yes	Yes
(Padgett et al., 2003)	Economic production as chemistry	ABM	Yes	Yes	No	No	No
(Becu et al., 2003)	A methodology for eliciting and modelling stakeholders' representations with agent based modelling	AI, Knowledge Engineering, ABM	No	No	No	Yes	Yes
(Zhang, 2003)	Growing Silicon Valley on a landscape: an agent-based approach to high-tech industrial clusters	Industrial Districts	Yes	Yes	No	No	No
(Mylopoulos et al., 2002)	Agent-oriented software development	AI	No	No	No	Yes	No
(Davidsson, 2001)	Multi agent based simulation: Beyond social simulation	ABM	No	Yes	No	No	No
(Downing et al., 2001)	Understanding climate policy using participatory agent-based social simulation	ABM	No	Yes	Yes	No	Yes
(Pahl-Wostl, 2002)	Towards sustainability in the water sector - The importance of human actors and processes of social learning	Water management	No	No	Yes	No	Yes
(Moss and Edmonds, 2005)	Sociology and simulation: Statistical and qualitative cross-validation	Sociology	No	Yes	No	No	No
(Reich, 2004)	Reasoning about other agents: A plea for logic-based methods	ABM, AI	No	No	No	Yes	No
(Ramanath and Gilbert, 2004)	The design of participatory agent-based social simulations	ABM, Sociology	No	Yes	No	No	Yes
(Zhang et al., 2005)	Dynamic game theoretic model of multi-layer infrastructure networks	LSTS, Infrastructures	No	Yes	Yes	No	No
(Castella et al., 2005)	Participatory simulation of land-use changes in the northern mountains of Vietnam: the combined use of an agent-based model, a role-playing game, and a geographic information system	Sociology, Ecology, Land use	No	Yes	No	No	Yes
(Pahl-Wostl and Hare, 2004)	Processes of social learning in integrated resources management	Water management, collaboration	No	Yes	Yes	No	Yes
(Aldea et al., 2004)	The scope of application of multi-agent systems in the process industry: three case studies	Process Technology	No	Yes	No	Yes	No
(Guyot and Honiden, 2006)	Agent-based participatory simulations: merging multi-agent systems and role-playing games	ABM, Sociology	No	Yes	No	No	Yes
(Banks et al., 2002)	Making computational social science effective - Epistemology, methodology, and technology	Social Science, Modelling	No	No	No	Yes	No
(Boero et al., 2004)	Micro behavioural attitudes and macro technological adaptation in industrial districts: an agent-based prototype	Industrial Districts	Yes	Yes	Yes	No	No
(Schenk et al., 2007)	Agent-based simulation of consumer behavior in grocery shopping on a regional level	Spatial Planning	Yes	Yes	No	No	No

Continued on next page

Ref	Title	Domain	CAS	ABM	LSTS	Evolution	Social process
(Jarupathirathana and Zahedi, 2007)	Dialectic decision support systems: System design and empirical evaluation	Decision Support	No	No	No	No	No



**Introduction** Our human society exerts great pressure on Earth's carrying capacity, leading to exhaustion of natural resources, loss of habitats and biodiversity, and causing a resource and climate crisis. To avoid a sustainability crisis, we urgently need to transform our production and consumption patterns. Given that we are part of a complex and integrated global system where and how should we begin this transformation? And how can we ensure that our transformation efforts will lead to a sustainable world?

This thesis focuses on industrial systems, which many claim are partly responsible for the global sustainability crisis. Rather than taking a traditional engineering science perspective, this thesis studies industrial systems as a Large-Scale Socio-Technical Systems, or  $\lambda$ -systems, composed of interconnected social and technical networks embedded in a geological and biological context. The  $\lambda$ -systems perspective implies we recognize global, national and regional  $\lambda$ -systems as systems that grow and evolve as a result of the decisions and actions undertaken by social entities or actors (individuals and organizations) in relation to the implementation, operation, design and innovation of the technical elements of these systems.

The research theme is thus "to increase our understanding of industrial networks and help steer their evolution towards sustainability," which implies a - methodological - research question: How can we model the evolution of Large-Scale Socio-Technical Systems such as industrial networks?

This work is aimed at developing a better understanding of regional industrial networks, with the goal of supporting Regional Development Authorities (RDA). We realize that this work only begins to capture the complexity of the global sustainability challenge, implicit in the research theme. The main hypothesis of this thesis, encompassing the larger theme, is that the use of adequate models of  $\lambda$ -system evolution will improve the ability of decision makers in industrial networks to make sustainable development choices. The hypothesis leads to the main research question: How can we create a model for exploring the evolutionary patterns of  $\lambda$ -systems? Three subquestions were derived from the central research question. First, how can a generativist Complex Adaptive Systems perspective be operationalized in models that capture  $\lambda$ -systems evolution? Second, what are the content specifications of such models in terms of the relevant formalisms (knowledge domains)? Third, what are the specifications for a process that would create such models?

**Theoretical foundations** As part of answering the first two research questions, we referred to the theory on Complex Adaptive Systems, which stipulates that overall  $\lambda$ -system behavior can be understood as an emergent property of interactions between autonomous agents at the lowest level of the system - in our case the firms and their technologies.

In our framework for understanding  $\lambda$ -system evolution, a system is conceptualized as consisting of three levels: agent, network and emergent system. At the agent level, the agent is defined as the smallest system component; its state and rules determine its behavior in terms of conversion of inputs to outputs and its adaptivity. Agents are similar in their interaction abilities and diverse in their states and rules. At the network level, interactions between the agents create a network with a certain topology. Changes in the intensity of the connections between agents determine the network dynamics; the addition and removal of nodes and edges makes the network develop and evolve. The system level is where the entire system manifests itself as an entity with aggregate in- and outputs, this entity having an aggregate state and aggregate rules. In reality, the system's properties are emergent; they are the result of low-level interactions. The system exists within a larger environment, it self-organizes, is robust, can be unstable, and its description is observer-dependent.

The presented framework is holistic, as it looks at all aspects of a system - from the smallest individual elements to the highest level of system aggregation. It considers systems in their entirety and only reduces elements to smaller elements if they are fully interrelated with other elements. It is generativist, as it understands a system to be the result of a continuous process of emergence across multiple levels, starting with the lowest level elements. It is multiformal, as it allows different languages to be used to describe different levels and agent properties.

An important insight gained from Complex Adaptive Systems theory is that predicting the exact outcome of the evolution of any  $\lambda$ -system is impossible, due to the intractability of the evolutionary process; there exists no faster way of predicting a systems outcome than to simply allow it to run its course. Simulation, however, can be used to generate emergent patterns of possible futures and identify system attractors.

Agent Based Modelling (ABM) was identified as the only generative modelling tool able to represent the structure and dynamics of evolving regional industrial networks. Individual companies and their technical installations are conceptualized as the smallest system elements or agents. Suitable representations of the systems economic environment (world market), natural environment (physical sources and sinks of materials) and policy environment (laws and regulations) are included in the model. Process Systems Engineering (PSE) is used to represent technical elements as a mass conserving black box characterized by its inputs and outputs. Corporate Finance offers suitable descriptions of the boundedly rational behavior of firms that own and operate the technical installations.

The number and diversity of agents in any industrial network is substantial, and needs to be adequately captured. Descriptions from technical, social, financial, ecological, engineering and other domains must be incorporated. These fields represent an equal number of distinct and often incompatible languages or formalisms. Knowledge management and artificial intelligence literature demonstrate that interconnecting this 'Tower of Babel' is not a trivial exercise. Building a shared multiformalism requires an engineered social process that involves domain experts of diverse backgrounds and expertise. To facilitate the communication between different vocabularies and formalisms, ontologies were identified as a means to create an interface between experts, while retaining the domain-specific vocabularies.

**Modelling Foundations** To answer the third research question an evolutionary modelling method was constructed, progressing through a series of case studies. Based on principles from collaborative modelling and insights from Complex Adaptive Systems and the evolutionary formalism, six requirements for the modelling process were formulated: open sourcedness of the tools used and created, sufficient diversity in the community of modellers, the organic growth of models, recorded history, enforceable authorship, and modularity of the software stack.

Model development is a co-evolutionary process that progresses in generations, or case studies, and which takes place in a dynamic, socially constructed environment of modellers and stakeholders who determine what is useful and what is not.

Co-evolution means that an element in an evolving system never exists in isolation but is always in interaction with others. A change in the fitness (survival ability) of any component has a direct effect on all other elements with which it shares its environment. In our method for model development there are four modelling aspects that co-evolve:

1. Technical aspects of the model, namely which software and hardware systems to use, how to organize the modelling software components, how to store data, how to analyze results, etc.;
2. The social process of involving the stakeholders in identifying and collecting relevant knowledge and providing feedback on the model's outcomes - this process involves the selection of the right participants, the execution of the collaborative process, the manner feedback is organized, etc.;
3. Formalized and encoded knowledge domain representations of  $\lambda$ -systems - microeconomics, chemical engineering and psychology are examples of relevant knowledge domains; and
4. Factual information that describes the components of the  $\lambda$ -system, their interactions and the overall system behavior - for example: specific processing plants, their in- and outputs, economic performance data, etc.

The modelling method was anticipated to lead to the evolution of increasingly richer and better models and tools, while at the same time providing insights per case study. The series of seven case studies explored in this work included three that we call 'learning case studies' because they explore the general applicability and usefulness of the modelling process, the lessons of which are later applied to the four practical case studies.

**Learning case studies** To begin with, the Flow-Based Evolution model was created to investigate whether the chosen conceptualization is good enough; in other words, whether or not we can eventually meet our objectives by representing a  $\lambda$ -system as an input/output flow-based network in an ABM in which the industrial network consists of producers and consumers (nodes) connected by mass flows (edges). Developing the model also helped us to determine what types of facts need to be collected and led us to the conclusion that technology descriptions and economic decision-making processes must be implemented as separate but connected modules.

In the Combination Of Infrastructures case study, the focus was on designing a social process for multiformal knowledge and fact collection. The model built was intended to help

elucidate the spatial combinability of infrastructures. Through this social process combinability was defined to consist of social, legal, safety and technical aspects. In the social process a proto-ontology of infrastructures and a parameterization of combinability aspects were created. The facts collected from the stakeholders allowed the construction of a fitness landscape that described the combinability of different infrastructures.

The final learning case, the Chocolate Game, involved the development and playing of a serious game and developing a model based on it. The game identified the information needed to create an ABM of a chocolate supply chain, that serves as an analogy of the chemical process industry. The analogy and associated information were extracted from the game using the System Decomposition Method (SDM), resulting in an ontology. This ontology serves as a formal, machine-readable representation of the language used for reasoning and communicating by the agents in the ABM. The SDM was designed as a collaboration script that consists of a group modelling exercise in which experts in relevant knowledge domains and formalisms interact. It offers a template of interfaces as well as a procedure to transform and encode their knowledge into a single multiformalism that defines the states and rules of an Agent Based Model. In the technical design aspect the conceptualization of flows was changed from discrete to continuous. The technical implementation of the ontology proved to be relatively inflexible.

**Main case study** In the CostaDue case study the SDM was improved and completed. A full-scale simulation engine for ABM modelling of  $\lambda$ -systems was developed, and knowledge and facts about chemical and biochemical processes were encoded. The model was used to explore the evolutionary patterns related to the transformation of the Groningen Seaports industrial cluster from a chlorine to a bio-based network. The agents had realistic economic properties and modular descriptions of technology, and they respected the mass balance. Basic economic reasoning was implemented through contract selection and price setting mechanisms. The possibility for the transition from a chemical to a bio-based industry was created by adding new bio-based technological options to the simulation. These options had been identified through a social process involving different stakeholders. The ensuing network evolution was studied under different economic scenarios representing various selection pressures. The main conclusions were that the bio-based options as identified by the stakeholders do not appear to lead to a diverse biomaterials-based network in the Groningen Seaports region. An enrichment of the existing network with bio-energy options is possible, the extent of which appears to depend on the survival of the incumbent energy-intensive industry. The importance of path dependency in network development is clearly demonstrated, as is the limited power that the Regional Development Agency has in steering this evolutionary process.

**Further case studies** To provide a robustness and applicability test of the modelling process, model and simulation engine, three additional case studies were completed. In the Bulk Biochemicals study the performance and evolution of a bio-refinery network was investigated across a large economic scenario space. Latin hypercube sampling was implemented as a technique to examine this large parameter space. Multi Criteria Assessment was implemented as a rationalization of the RDAs network development process. This network is likely to emerge and be successful under the majority of economic conditions examined. Testing a variety of RDA strategies revealed that an increased rationality of the RDA does not improve the performance of the network, due to the limited number of technological options available.

In the Metals Network case the evolution of a global aluminum and copper production network was studied under different economic conditions and different agent investment strategies. The case study extended the agents reasoning with Net Present Value and Internal Rate of Return calculations, as well as added a dynamic world market with global interest rate developments, next to encoding a wealth of metallurgical processes. Global economic conditions and agent investment policies appear to have little effect on the development of this network.

The Bioelectricity study was completed to examine the evolution of the Dutch bioelectricity production portfolio under different CO<sub>2</sub> emission taxation levels and under different agent reasoning strategies. The incorporation of Life Cycle Assessment (LCA) enabled agents to reason about their environmental impact across their supply chain. Incorporation of the EcoInvent LCA database enabled the World Market to provide goods with associated environmental impacts from 3000 different production processes. The case solved a number of complex algorithmic and computational challenges in combining a static analysis tool (LCA) with a dynamic pattern generation tool (ABM). The main methodological outcome was a practical way to combine LCA with ABM. The study showed that high levels of CO<sub>2</sub> emission taxation allow for structural change in the way bioelectricity production is organized.

**Results** Returning to the idea of the four co-evolving modelling aspects, the main accomplishment in the technical dimension is the design and implementation of the modular, open source simulation engine. Recognition of the necessity of recording its development history is important to the science and engineering of modelling. In the social dimension, the results are the System Decomposition Method (SDM), the emergent community sharing the practical knowledge on co-evolutionary modelling processes and the models that have been developed and recorded in a wiki system. The results of the knowledge formalization process are the formalizations of the PSE, microeconomics, corporate finance, industrial clusters, evolution and Complex Adaptive Systems domains. This knowledge is formalized in an ontology. The practical outcome of the fact collection process is the encoding of large numbers of industrial processes, their flows and economic properties.

**Domain insights** On the basis of the modelling, case study results, insights from Complex Adaptive Systems and evolution theory, seven general guidelines for shaping and steering the evolution of industrial networks can be given.

First, network development is strongly path dependent, and RDAs must strive to develop an understanding of networks' future development patterns. The order in which firms appear matters. Second, once established, the network structure is relatively robust. Third, the social, legal, institutional and regulatory contexts can make or break a industrial network, even if the right firm and technology mix is present. Fourth, given the importance of past decisions and the chaotic nature of the evolutionary process, it is inevitable that mistakes will be made and the wrong type of firm will be allotted space in the region. Consequently, RDAs must at all times retain spatial control of their region. Fifth, RDAs must be aware of the importance of diversity - in the types of firms and their physical installations, as well as in the options available for the same functions within their network. Without diversity, evolution is impossible. Sixth, the importance of the long-term view must be emphasized. In their planning, RDAs must be able to look ahead several generations of firms or technical installations. Given the average installation lifetime of 15 years or more, RDAs need to use a multi-decade planning perspective. Finally,

RDAs must strive for balance. Too much top-down control will stifle change, and too much bottom-up initiative will destroy the networks coherence.

**Conclusions** Concluding, the execution of seven case studies in an evolutionary modelling process has allowed us to answer the research questions and to conclude that the posed hypothesis that “the use of adequate models of  $\lambda$ -system evolution will improve decision-making abilities in industrial network development” was not falsified.

Central to this thesis has been the development of a method for creating models. Operationalizing insights from complex adaptive systems theory and evolutionary thinking, has led to the development of requirements for the modelling method and the method itself. It has been used to create consecutively “good enough” models of  $\lambda$ -systems evolution. These models are “good enough” in the sense that they provide useful insights that can support strategic decision makers involved in industrial network development. The main practical result presented is the description of the modelling process, a modular, expandable simulation engine, a collection of domain knowledge formalized in an ontology and the encoding of a large number of facts on industrial network elements.

One of the main strengths of the models presented in this work is that their representation of  $\lambda$ -systems is intuitively understood by users and modelers. Furthermore, ABMs represent an exciting new paradigm, through which collective understanding of  $\lambda$ -systems can be expressed. Their main weaknesses are the large data requirement for realistic models and relatively complicated implementation. Main strengths of the modelling process are that it is a socially inclusive, adaptive process for long term capacity building with a high scientific output. Its weaknesses are that its high cost in terms of people and time, the slow takeoff phase, the dependence on the quality of the social network and its divergence from the modeling paradigms that decision makers are familiar with. It takes an imaginative decision maker to truly appreciate ABM.

The co-evolution of the technical design of the models, the social process involved, the knowledge and facts encoded have been set in motion. As this body of knowledge gathers speed and momentum it will continue to increase our understanding of  $\lambda$ -system evolution. We hope that it will ultimately contribute to a more sustainable development of the human species on planet Earth.

**Thema en onderzoeksvragen** De belasting van het systeem Aarde door de mensheid leidt tot uitputting van natuurlijke hulpbronnen, de teloorgang van habitats en tot verlies van biodiversiteit. Inmiddels dreigt een grondstoffen- en klimaatcrisis. Alleen door verandering van onze productie- en consumptiepatronen kan een duurzaamheidscrisis worden afgewend. Echter, wij maken deel uit van een complex en geïntegreerd wereldomspannend systeem waarvan de onderdelen op allerlei wijzen (via geld-, massa- en informatiestromen) met elkaar zijn verbonden, over enorme afstanden en tijdschalen. Waar en hoe kunnen we beginnen aan de benodigde transformatie? En hoe kunnen we zeker stellen dat onze inspanningen zullen leiden tot een duurzame wereld?

Dit proefschrift is gericht op industriële systemen; systemen die op zijn minst ten dele debet zijn aan de mondiale duurzaamheidscrisis. In dit onderzoek beschouwen we industriële systemen niet vanuit het traditionele ingenieursperspectief, maar als Grootschalige Socio-Technische Systemen. Dergelijke systemen, ook wel  $\lambda$ -systemen genoemd, bestaan uit onderling verbonden sociale en technische netwerken, die op hun beurt ingebed zijn in een biologische en geologische omgeving. Het  $\lambda$ -systemen systeemperspectief impliceert bovendien dat we de mondiale, nationale en regionale industriële systemen zien als systemen die groeien en evolueren als gevolg van handelingen en beslissingen die door sociale entiteiten, actoren (individuen en organisaties), worden genomen over ontwerp, implementatie, werking, gebruik en innovatie van deze systemen.

Zo werd het centrale thema van dit proefschrift: “inzicht te verkrijgen in het gedrag van industriële netwerken om sturing van hun evolutie in meer duurzame richting mogelijk te maken”. Dit thema impliceert een methodologische onderzoeksvraag: “Hoe kunnen we de evolutie van Grootschalige Socio-Technische systemen modelleren?”

De focus van dit werk is het ontwikkelen van een beter begrip van regionale industriële netwerken, beseffend dat dit maar een begin is van de uitwerking van het centrale thema. Onze hypothese is dat het gebruik van adequate modellen van de evolutie van  $\lambda$ -systemen strategische beslissers in deze industriële netwerken kan helpen om “hun” systeem in de richting van duurzame ontwikkeling te sturen. Deze hypothese leidt tot de hoofdonderzoeksvraag: “Hoe kunnen we modellen maken om de evolutionaire patronen van  $\lambda$ -systemen te verkennen?” Uit deze onderzoeksvraag zijn drie subvragen afgeleid: (1) Hoe kan een generatief perspectief op Complexe Adaptieve Systemen worden geoperationaliseerd in modellen van  $\lambda$ -systeem evolutie? (2) Wat zijn de inhoudelijke specificaties van zulke modellen in termen van relevante formalismes (domeinkennis)? (3) Wat zijn de specificaties van het betreffende modelleerproces?

**Theoretische achtergrond** Een deel van het antwoord op de eerste twee vragen vonden wij in de theorie van Complexe Adaptieve Systemen. Deze geeft aan dat het gedrag van  $\lambda$ -systemen kan worden beschreven als een emergente eigenschap van de interacties van autonome agenten op het laagste systeemniveau - in ons geval bedrijven en hun technologie.

Het raamwerk dat we ontwikkelden om de evolutie van  $\lambda$ -systemen te kunnen begrijpen bestaat uit drie conceptuele niveaus: agent, netwerk en emergent systeem. Op het laagste niveau is de agent gedefinieerd als het kleinste systeemelement; zijn toestand en gedragsregels bepalen hoe zijn inputs worden omgezet in outputs, en hoe de agent zich aanpast aan zijn omgeving. De agenten zijn gelijk in hun mogelijkheden tot interactie, maar verschillen in de toestanden die ze kunnen aannemen en de gedragsregels die ze hanteren. Op het netwerkniveau ontstaat door wisselwerkingen tussen de agenten een netwerk met een bepaalde structuur. Veranderingen in de intensiteit van verbindingen tussen agenten leiden tot dynamiek in het netwerk. Het toevoegen of verwijderen van verbindingen of agenten leidt tot evolutie van het netwerk. Op het emergente systeemniveau kan het systeem opgevat worden als een geaggregeerde agent, met geaggregeerde inputs en outputs, en met geaggregeerde gedragsregels. In werkelijkheid zijn alle systeemeigenschappen op dit niveau echter emergent, immers ze vloeien voort uit de wisselwerking tussen de componenten op lagere systeemniveaus. Het systeem bevindt zich in een omgeving, is zelf-organiserend, robuust, en kan instabiel zijn; de wijze van beschrijving is afhankelijk van de toeschouwer.

Dit is een holistisch raamwerk omdat het alle aspecten van een systeem in beschouwing neemt, van het kleinste element tot het algehele systeem. Het beschouwt het systeem in zijn geheel, en ontleedt systeemelementen slechts in kleinere onderdelen als deze volledig in verbinding staan met elkaar. Het is een generatief raamwerk, omdat het systemen beschouwt als het resultaat van een continu proces van emergentie dat begint bij de kleinste systeemonderdelen. Het raamwerk is multiformeel, dat wil zeggen het staat het gebruik van verschillende “talen” (uit verschillende kennisdomeinen) toe bij de beschrijving van systemen op verschillende aggregatieniveaus van de verschillende typen eigenschappen van de systeemonderdelen.

Een belangrijk inzicht uit de theorie van Complexe Adaptieve Systemen is dat het onmogelijk is om de precieze uitkomst van een evoluerend systeem te voorspellen. Evoluerende systemen zijn onnavolgbaar en daarmee onvoorspelbaar. Er is geen snellere manier om de uitkomst van een evolutieproces te berekenen dan het proces zelf te laten verlopen. Simulatie van evolutieprocessen heeft echter wel degelijk zin om te verkennen welke patronen zouden kunnen ontstaan, om daarmee als het ware de “mogelijke toekomst” van het systeem te identificeren, zonder enige pretentie om een exacte uitkomst te voorspellen.

Agentgebaseerde Modellen (ABM) zijn geïdentificeerd als de enige modelleeraanpak die generatief is én in staat is de structuur en dynamica van evoluerende regionale industriële netwerken te beschrijven. In onze ABM worden bedrijven en hun technische installaties gezien als de kleinste onderdelen van het systeem, de agenten. In de beschrijving van het systeem wordt de relevante economische omgeving gerepresenteerd (wereldmarkt), evenals de natuurlijke omgeving (bron en ontvanger van materiestromen) en de bestuurlijke omgeving (relevante wetgeving en regulering). Met behulp van proceskundige kennis worden de relevante technische eenheden/agenten beschreven als een “black box” met in- en outputs; bedrijfskundige inzichten en noties van “beperkte rationaliteit” zijn gebruikt om het gedrag van de bedrijven die deze technische installaties beheren te beschrijven.

Industriële netwerken bevatten een aanzienlijk aantal agenten van divers karakter die allen op een geschikte manier dienen te worden beschreven. Om dit voldoende nauwkeurig te kun-

nen doen zijn inzichten uit een groot aantal disciplines nodig, zoals de technische, sociale en bedrijfskundige wetenschappen. De verschillende onderzoeksvelden hanteren even zoveel verschillende (vaak incompatibele) talen of formalismen, die met elkaar moeten worden verbonden. De literatuur op het gebied van kunstmatige intelligentie en kennismanagement laat zien dat het smeden van verbindingen in deze wetenschappelijke “Toren van Babel” geen sinecure is. Het creëren van een gedeeld formalisme, of taal, die door domeinexperts uit verschillende disciplines wordt begrepen en gebruikt, vereist een zorgvuldig ontworpen sociaal proces. Om dit proces te faciliteren en de verschillende talen die nodig zijn om de modellen te maken, zodanig te verbinden dat betekenisvolle communicatie over disciplinegrenzen heen mogelijk wordt, is gebruik gemaakt van ontologieën. Ontologieën vormen de interfaces tussen verschillende kennisgebieden, terwijl ze toelaten dat de betrokken experts hun eigen taal gebruiken.

**Modelleren** Om de derde onderzoeksvraag te beantwoorden is een co-evolutionaire modelleermethode ontworpen, die vordert in generaties, per casus. Gebaseerd op inzichten van “collaborative modelling”, complexiteits- en evolutietheorie zijn zes eisen geformuleerd voor de modelleermethode: (1) de brongegevens van de gebruikte (software) gereedschappen dienen openbaar te zijn; (2) het sociale netwerk van modelleurs en experts is voldoende divers; (3) modellen kunnen organisch groeien (4) de ontwikkelgeschiedenis wordt volledig vastgelegd en is altijd opvraagbaar (5) de identiteit van alle deelnemers aan het proces is bekend; (6) de ontwikkelde software is modulair.

Modelontwikkeling is een co-evolutionair proces, dat over generaties, casus na casus, vordert en plaats vindt in een dynamische en sociaal geconstrueerde omgeving van modelleurs en belanghebbenden, die gezamenlijk bepalen welke verbetering nuttig is en welke niet.

Co-evolutie betekent dat een element in een evoluerende omgeving nooit is geïsoleerd, maar steeds in wisselwerking staat met andere elementen. Verandering in “fitness”, of zijn vermogen tot overleven, is daardoor altijd gekoppeld aan en heeft altijd invloed op alle andere co-evoluerende elementen. In de beschreven modelleermethode zijn er vier elementen die co-evolueren:

1. *Technische model aspecten.* Welke software en hardware systemen worden gebruikt, hoe is de software georganiseerd, hoe wordt data beheerd en opgeslagen, hoe worden de uitkomsten geanalyseerd, enzovoorts.
2. *Het sociale proces* dat de belanghebbenden en de modelleurs helpt om de relevante kennis en feiten te identificeren en feedback te leveren in het modelleerproces. Het sociale proces bestaat uit de selectie van relevante deelnemers, de feitelijke samenwerking, de vorm(en) van terugkoppeling, en het selectiemechanisme van de co-evolutie.
3. *Geformaliseerde en gecodeerde kennis* over  $\lambda$ -systemen, hun elementen, hun interacties en het emergente systeemgedrag. Voorbeelden van relevante kennisgebieden zijn psychologie, bedrijfskunde, scheikundige technologie, enzovoorts.
4. *Feitelijke informatie* die de relevante entiteiten en hun interacties in het  $\lambda$ -systeem beschrijft. Bijvoorbeeld de technische specificatie van een fabriek, de economische prestaties daarvan, enzovoorts.

De verwachting was dat deze modelleermethode steeds rijkere en nuttiger modellen zal doen ontstaan, naast het genereren van nieuwe inzichten in elke casus. Dit proefschrift laat een serie

van zeven casus zien; de serie begint met drie “leercasus” waarin de toepasbaarheid en het nut van de modelleermethode zelf verkend worden. De in het proces ontwikkelde en verbeterde methode is daarna toegepast op vier praktische probleemgedreven casus.

**Leercasus** Om te beginnen is een generiek model ontwikkeld dat “stroom-gebaseerde evolutie” beschrijft en kan simuleren. Doel van de casus was te verkennen of de gekozen systeemconceptualisatie überhaupt geschikt is. De zo geteste conceptualisatie van  $\lambda$ -systemen gaat er van uit dat een industrieel netwerk voorgesteld kan worden als een netwerk van in- en uitgaande massastromen tussen producerende en consumerende agenten. De modelontwikkeling in deze casus hielp om te bepalen welke typen feitelijke kennis noodzakelijk zijn. De resultaten leidden tot de conclusie dat de beschrijving van de technische elementen voldoende nauwkeurig was, maar dat de bedrijfseconomische beslissingsprocessen moesten worden geïmplementeerd als modulaire elementen.

Het ontwerp van het sociale proces voor het coderen van multiformele domeinkennis en -feiten stond centraal in de “Combinatie van Infrastructuren” casus. Daarvan was het doel de ruimtelijke combineerbaarheid van infrastructuren te verkennen. In het sociale proces werd vastgesteld dat combineerbaarheid afhangt van sociale, wettelijke, veiligheidskundige en technische aspecten. Het proces leverde een proto-ontologie op; de verzamelde feitelijke informatie maakte het mogelijk om een combineerbaarheidslandschap te maken waarmee de “fitness” van de combinatiemogelijkheden van verschillende infrastructuren wordt gevisualiseerd.

Het “Chocoladespel” was de laatste leercasus. Deze bestond uit het ontwikkelen van een “serious game” waarvoor daarna een agentgebaseerd model werd ontwikkeld. Het spel verkende de noodzakelijke kennis om een productieketen voor chocola te beschrijven. Deze beschrijving dient als een analogie van de productieketens in de chemische industrie. De analogie en de noodzakelijke feiten werden verkregen door ontwikkeling van een “Systeem Decompositie Methode” (SDM). Deze is ontworpen als een sociaal script dat bestaat uit een groepsmodelleerproces waarin domeinexperts hun kennis delen en formaliseren. De SDM leidde vervolgens tot de ontwikkeling van een ontologie die gebruikt wordt als de taal waarmee de agenten in het model kunnen communiceren. De ontologie laat het coderen van verschillende formalismen toe (technische en bedrijfskundige); tevens maakt ze een eenduidige beschrijving van agenten mogelijk. Wat betreft het modelleertechnische aspect is in deze casus het werken met discrete stromen verkend; daaruit bleek dat de gekozen technische implementatie van de ontologie niet flexibel genoeg was.

**Hoofd casus** In de “CostaDue” casus is de SDM verbeterd, is de agentgebaseerde simulatieomgeving ontwikkeld, is een model gemaakt en zijn kennis en feiten van chemische industrie en bioprocindustrie vastgelegd in de ontologie. De casus omvat een verkenning van de mogelijke evolutiepatronen van het industriële netwerk in en rond Delfzijl, meer in het bijzonder van de mogelijkheden om een transformatie te bewerkstelligen van een chloor- naar een biomassa-gebaseerd netwerk. De beschreven agenten hebben realistische, modulaire economische en technische beschrijvingen die voldoen aan het criterium van een gesloten massabalans. De bedrijfskundige aspecten van de agenten bevatten beslissingen over de prijsstelling van hun producten en methodes voor de selectie en het sluiten van contracten. De evolutie naar een biomassa-gebaseerd netwerk werd gesimuleerd door agenten die biomassa (kunnen) verwerken toe te voegen aan het bestaande netwerk van chloorchemie-gerelateerde agenten. De set van biomassa-agenten werd bepaald in een sociaal proces met een keur van deskundigen en lokale

stakeholders. De resulterende evolutionaire ontwikkeling van het netwerk werd verkend voor verschillende economische scenario's. Uit de simulatieruns volgt dat het niet waarschijnlijk is dat er een robuust en divers biomaterialencluster zal ontstaan in Delfzijl; wel zijn er mogelijkheden dat een bioenergiecluster zich zal ontwikkelen. Echter, dat is afhankelijk van het voortbestaan van de huidige, energieintensieve, chloorgebaseerde industrie. Uit de simulaties blijkt bovendien dat de padafhankelijkheid groot is en dat het vermogen van het havenbedrijf om als regionale ontwikkelingsautoriteit de evolutie van het netwerk te beïnvloeden beperkt is.

**Overige casus** Om de robuustheid en algemene toepasbaarheid van de co-evolutionaire modelleermethode, het model en de simulatie omgeving te verkennen zijn nog drie casus uitgevoerd. In de “Bulk-biochemicaliën” casus werden de evolutionaire ontwikkelingspaden inclusief de economische prestaties van een (gedistribueerde) bioraffinaderij gesimuleerd onder een groot aantal verschillende economische omstandigheden. De “Latin Hypercube Sampling” methode is gebruikt om de zeer grote parameterruimte systematisch te verkennen. Multicriteria analyse is gebruikt om het sturingsproces van de gebiedsmanager te rationaliseren. De hoofduitkomst was dat de bioraffinaderij onder de meeste economische omstandigheden waarschijnlijk winstgevend zal zijn. Een verkenning van de verschillende sturingsstrategieën van de regiomanager liet zien dat het rationaliseren van de beslissingen weinig effect heeft op de prestaties van het netwerk. Dat komt vooral doordat het aantal technische opties beperkt is.

In de “Metalenproductienetwerk” casus is de evolutie van het wereldwijde aluminium- en koperproductienetwerk verkend onder verschillende economische omstandigheden en investeringscriteria gehanteerd door de agenten. Daartoe zijn de bedrijfskundige redeneerprocessen van de agenten uitgebreid met Netto Contante Waarde en Investeringsrendement overweginen. Verder is het model van de wereldmarkt uitgebreid met dynamisch prijsstellingsgedrag en renteontwikkelingen en is een groot aantal metallurgische processen en mijnbouwtechnieken toegevoegd aan de ontologie. De simulaties van het netwerkevolutieproces brachten aan het licht dat economische condities en de investeringsstrategieën van de agenten weinig invloed hebben op de uiteindelijke netwerkstructuur omdat er maar een beperkt aantal technische opties beschikbaar is.

De laatste casus was het “Bioelektriciteits” model. Hierin is de evolutie van de Nederlandse elektriciteitsproductieportfolio bestudeerd onder verschillende regimes voor CO<sub>2</sub> emissiebelasting en investeringsstrategieën van agenten. Levenscyclusanalyse (LCA) is gecombineerd met het agentgebaseerde model zodat agenten hun beslissingen (mede) kunnen baseren op de milieueffecten over gehele leveringsketens. Koppeling tussen het model en de EcoInvent LCA database maakt het de agenten mogelijk om de milieueffecten van meer dan 3000 producten te overwegen. In de casus is een aantal complexe algoritmische problemen opgelost die voortvloeien uit de combinatie van een dynamische model met een statische database. Het belangrijkste methodologische resultaat is dan ook het verbinden van een statische methode (LCA) met een dynamische simulatie (ABM). De casus liet ook zien dat hoge CO<sub>2</sub> belastingniveaus leiden tot structurele veranderingen in de elektriciteitsproductieportfolio, en dat die verandering gepaard gaat met het verdwijnen van vervuilende technologie.

**Resultaten** Terugblikkend naar de vier co-evoluerende aspecten is het belangrijkste resultaat in de technische dimensie het ontwerp en de implementatie van de modulaire, open-bron simulatieomgeving. De erkenning van de noodzaak tot vastleggen van het ontwikkelproces is een belangrijke toevoeging aan de wetenschap van modelleren. De belangrijkste resultaten wat

betreft de sociale dimensie van het modelleerproces zijn het ontwerp van de Systeem Decompositie Methode (SDM), het emergente sociale netwerk waarbinnen praktische kennis van ABM modelontwikkeling wordt gedeeld en de modellen zelf die ontwikkeld en beschreven zijn op het gebruikte wiki platform. Het belangrijkste resultaat van het proces van kennisformalisering is de ontologie die de brug slaat tussen de vele betrokken kennisgebieden (o.m. scheikundige technologie, bedrijfskunde, economie, netwerktheorie en complexiteitstheorie). Het praktische resultaat van het proces van feitenverzameling is de codering van een groot aantal industriële processen, hun stromen en economische eigenschappen.

**Domeinspecifieke inzichten** Op basis van de ontwikkelde modellen, de resultaten uit de casus en inzichten uit de complexiteits- en evolutietheorie kunnen zeven generieke aanbevelingen worden gegeven voor het sturen van de ontwikkeling van industriële netwerken.

Ten eerste, de ontwikkeling van industriële netwerken is sterk padafhankelijk. De volgorde waarin bedrijven tot een netwerk toetreden is sterk bepalend voor de netwerkontwikkeling. Dat betekent voor de gebiedsmanager dat inzicht in de mogelijke patronen van ontwikkeling essentieel is, afhankelijk van de aard en volgorde waarin bedrijven zich aandienen. Ten tweede, een industrieel netwerk is, eenmaal ontwikkeld, zeer robuust en moeilijk te veranderen. Ten derde, de sociale en institutionele context kunnen een netwerk maken of breken, zelfs bij de aanwezigheid van de juiste mix van technologieën. Ten vierde, gegeven de padafhankelijkheid en het chaotische karakter van evolutieprocessen is het onvermijdelijk dat een gebiedsmanager fouten zal maken. Het is daarom uitermate belangrijk dat een gebiedsmanager als grondeigenaar of -uitgever controle over zijn gebied behoudt. Vijf, diversiteit in zowel technische installaties als soorten bedrijven in een netwerk is van groot belang voor de veerkracht van het netwerk. Een gebiedsmanager zou deze diversiteit moeten stimuleren en bewaken, zodat daarmee het vermogen tot evolutionaire aanpassing van het netwerk behouden blijft. Zes, het belang van een lange-termijn visie moet benadrukt worden. Om de evolutie van industriële netwerken te kunnen sturen is een visie over meerdere generaties van vernieuwing nodig. Gezien het feit dat grootschalige technische installaties tien tot veertig jaar meegaan, vereist dit een langetermijn gebiedsontwikkelings visie. En tenslotte zouden gebiedsmanagers moeten streven naar balans. Te veel top-down sturing of te veel bottom-up initiatief kan de coherentie van een industrieel netwerk ernstig verstoren.

**Conclusies** De zeven evolutiemodellen die met behulp van de co-evolutionaire modelleermethode tot stand zijn gekomen, hebben ons in staat gesteld om de drie onderzoeksvragen te beantwoorden en te concluderen dat de hypothese dat “het gebruik van adequate modellen van de evolutie van  $\lambda$ -systemen zal leiden tot verbeterde en meer duurzame beslissingen in deze industriële netwerken” niet gefalsificeerd is.

Het centrale thema van dit proefschrift was de ontwikkeling van een co-evolutionaire methode voor het ontwikkelen van simulatiemodellen voor de evolutie van complexe socio-technische systemen. Het operationaliseren van inzichten uit de complexiteits- en evolutietheorie heeft geleid tot de ontwikkeling van eisen voor de modelleermethode en voor de modellen zelf. De methode is toegepast om een serie modellen van  $\lambda$ -systemen te maken die “goed genoeg” zijn, in de zin dat ze nuttige inzichten leveren die gebruikt kunnen worden om strategische beslissers te ondersteunen in hun streven de evolutie van de industriële clusters in “hun” gebied in duurzame richting te sturen. De praktische resultaten van dit proefschrift zijn het modelleerproces, de modulaire, uitbreidbare simulatieomgeving, de collectie van domeinkennis geformaliseerd in een

ontologie en de codering van een groot aantal feitelijke gegevens over de elementen (procesinstallaties, technologieën) van industriële netwerken.

Het belangrijkste voordeel van de modellen beschreven in dit proefschrift is dat hun structuur intuïtief wordt begrepen door de gebruikers en modellers. ABM is een nieuw en spannend modelleerparadigma; het gebruik ervan laat het ontstaan van een collectief begrip van  $\lambda$ -systemen toe. Het belangrijkste nadeel is dat ABM een relatief grote dataintensiteit hebben en dat hun implementatie relatief gecompliceerd is. Het belangrijkste voordeel van de gepresenteerde modelleermethode is het sociaal inclusieve karakter; een adaptief proces leidt tot capaciteitsontwikkeling bij de modellers en de gebruikers. De belangrijkste zwaktes zijn dat het een relatief duur proces is in termen van tijd en mankracht, dat de opstartfase (daardoor) traag is, dat het sterk afhankelijk is van de kwaliteit (diversiteit) van het sociale netwerk en dat de gebruikers die vertrouwd zijn met traditionele modelleermethodes, ABM nog moeten leren appreciëren.

Het co-evolutionaire proces van het technische modelontwerp, het sociale procesontwerp, de codering van kennis en data is in gang gezet, en neemt gestaag toe in snelheid. We hopen dat dit proces uiteindelijk zal leiden tot meer inzicht in de evolutie van  $\lambda$ -systemen en daarmee zal bijdragen aan een kleine stap voorwaarts in de richting van een duurzame ontwikkeling.



## CURRICULUM VITAE

Igor Nikolić was born on the 24<sup>th</sup> of July 1975 in Novi Sad, former Yugoslavia. After completing his high school in Novi Sad, Budapest, and Arnhem, he started his MSc studies at the Delft University of Technology where he graduated with honors as a chemical- and bio-process engineer. He was one of the first students to get a “Sustainable technology” annotation on his degree. In his M.Sc. thesis he presented an Agent Based Model of gene flow from GMO crops to surrounding plant populations. After his graduation, he spent several years as an environmental researcher and consultant at University of Leiden, Institute for Environmental Science (CML), where he worked on life cycle analysis (LCA), material/substance flow analysis (MFA/SFA), photo identification of whales and has specialized in industrial ecology. He started his PhD thesis in 2004 at the Energy and Industry section of the Faculty of Technology, Policy and Management at the Delft University of Technology. His thesis work focused on the design of a co-evolutionary method for constructing Agent Based Models of the evolution of Large Scale Socio-Technical systems. In his research he specializes in applying complex adaptive systems theory, agent based modeling and evolutionary theory to model industry–infrastructure network evolution. He takes a heavy hint from evolutionary biology and ecosystem behavior in his understanding of industrial ecology. Next to his scientific work he has organized, set up and managed [wiki.tudelft.nl](http://wiki.tudelft.nl). He is an active networker and promoter of open source and social software that enables collaborative, multidisciplinary research work.

Currently, he is an Assistant Professor at the Energy and Industry group, Faculty of Technology, Policy and Management faculty, Delft University of Technology where he studies Large Scale Socio-Technical system evolution in all of its aspects. An up to date profile is available on <http://www.linkedin.com/in/igornikolic>.

## NGInfra PhD Thesis Series on Infrastructures

1. Strategic behavior and regulatory styles in the Netherlands energy industry  
Martijn Kuit, 2002, Delft University of Technology, The Netherlands.
2. Securing the public interest in electricity generation markets, The myths of the invisible hand and the copper plate  
Laurens de Vries, 2004, Delft University of Technology, The Netherlands.
3. Quality of Service Routing in the Internet: Theory, Complexity and Algorithms  
Fernando Kuipers, 2004, Delft University of Technology, The Netherlands.
4. The role of power exchanges for the creation of a single European electricity market: market design and market regulation  
François Boisseleau, 2004, Delft University of Technology, The Netherlands, and University of Paris IX Dauphine, France.
5. The ecology of metals  
Ewoud Verhoef, 2004, Delft University of Technology, The Netherlands.
6. MEDUSA, Survivable information security in critical infrastructures  
Semir Daskapan, 2005, Delft University of Technology, The Netherlands.
7. Transport infrastructure slot allocation  
Kaspar Koolstra, 2005, Delft University of Technology, The Netherlands.
8. Understanding open source communities: an organizational perspective  
Ruben van Wendel de Joode, 2005, Delft University of Technology, The Netherlands.
9. Regulating beyond price, integrated price-quality regulation for electricity distribution networks  
Viren Ajodhia, 2006, Delft University of Technology, The Netherlands.
10. Networked Reliability, Institutional fragmentation and the reliability of service provision in critical infrastructures  
Mark de Bruijne, 2006, Delft University of Technology, The Netherlands.
11. Regional regulation as a new form of telecom sector governance: the interactions with technological socio-economic systems and market performance  
Andrew Barendse, 2006, Delft University of Technology, The Netherlands.
12. The Internet bubble - the impact on the development path of the telecommunications sector  
Wolter Lemstra, 2006, Delft University of Technology, The Netherlands.
13. Multi-Agent Model Predictive Control with Applications to Power Networks  
Rudy Negenborn, 2007, Delft University of Technology, The Netherlands.
14. Dynamic bi-level optimal toll design approach for dynamic traffic networks  
Dusica Joksimović, 2007, Delft University of Technology, The Netherlands.

15. Intertwining uncertainty analysis and decision-making about drinking water infrastructure  
Machtelt Meijer, 2007, Delft University of Technology, The Netherlands.
16. The New EU Approach to Sector Regulation in the Network Infrastructure Industries  
Richard Cawley, 2007, Delft University of Technology, The Netherlands.
17. A functional legal design for reliable electricity supply, How technology affects law  
Hamilcar Knops, 2008, Delft University of Technology, The Netherlands and Leiden University, The Netherlands.
18. Improving Real-Time Train Dispatching: Models, Algorithms and Applications  
Andrea D'Ariano, 2008, Delft University of Technology, The Netherlands.
19. Exploratory modelling and analysis: A promising method to deal with deep uncertainty  
Datu Buyung Agusdinata, 2008, Delft University of Technology, The Netherlands.
20. Characterization of Complex Networks: Application to Robustness Analysis  
Almerima Jamaković, 2008, Delft University of Technology, Delft, The Netherlands.
21. Shedding light on the black hole, The roll-out of broadband access networks by private operators  
Marieke Fijnvandraat, 2008, Delft University of Technology, Delft, The Netherlands.
22. On Stackelberg and Inverse Stackelberg Games & Their Applications in the Optimal Toll Design Problem, the Energy Markets Liberalization Problem, and in the Theory of Incentives  
Kateřina Staňková, 2009, Delft University of Technology, Delft, The Netherlands.
23. On the Conceptual Design of Large-scale Process & Energy Infrastructure Systems: Integrating Flexibility, Reliability, Availability, Maintainability and Economics (FRAME) Performance Metrics  
Austine Ajah, 2009, Delft University of Technology, Delft, The Netherlands.
24. Comprehensive models for security analysis of critical infrastructure as complex systems  
Fei Xue, 2009, Politecnico di Torino, Torino, Italy.
25. Towards a single European electricity market, A structured approach for regulatory mode decision-making  
Hanneke de Jong, 2009, Delft University of Technology, The Netherlands.
26. Co-evolutionary process for modelling Large Scale Socio-Technical Systems evolution  
Igor Nikolić, 2009, Delft University of Technology, The Netherlands.

Order information: [info@nginfra.nl](mailto:info@nginfra.nl)

Igor Nkolić

# Co- Evolutionary Method For Modelling Large Scale Socio- Technical Systems Evolution

Exactly predicting the future of an evolving large scale socio-technical system is impossible. Yet, if we are to sustainably manage the industrial and infrastructure systems our society depends on, we must understand how the actions we take today will affect the evolution of these systems. Simulating how the social and technical networks co-evolve over time allows us to explore possible system futures. This knowledge can help today's decision makers to steer the system away from undesirable evolutionary pathways.

Creating models that capture the complexity of socio-technical systems co-evolution requires multiple formalisms to be encoded in a modeling framework that itself evolves. This thesis presents a method for creating Agent Based Models that suitably represent complex evolving systems. The method involves a co-evolution between the technical aspects of model development, the social process involving the stakeholders, the collection of relevant domain knowledge and the encoding of facts. Through seven case studies the method is demonstrated to yield subsequent generations of richer and ever more useful simulation models.

## The Next Generation Infrastructures Foundation

represents an international consortium of knowledge institutions, market players and governmental bodies, which joined forces to cope with the challenges faced by today's and tomorrow's infrastructure systems. The consortium cuts across infrastructure sectors, across disciplinary borders and across national borders, as infrastructure systems themselves do. With the strong participation of practitioners in a concerted knowledge effort with social and engineering scientists, the Foundation seeks to ensure the conditions for utilization of the research results by infrastructure policy makers, regulators and the infrastructure industries.

[www.nginfra.nl](http://www.nginfra.nl)

