NTNU
Innovation and Creativity

Simulation of Innovation in Mobile Communication Markets

Martin Andreas Børke

Master of Science in Computer Science
Submission date: May 2007
Supervisor: John Krogstie, IDI
Co-supervisor: Øyvind Strømme, Accenture

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

Many methods and algorithms for simulation of business processes exist today. In many cases it may be difficult to decide which model is best fitted for simulating a specific scenario. Knowledge of this requires knowledge of both economy/strategy and algorithms for modelling. Further it's needed to have the ability to combine these fields in such a way that the result of the simulation is perceived as a value increase.

This task will look into simulation models for business processes. It shall be argumented why one model is chosen before the others, and a solution shall be designed and implemented. Analysis of the results related to expected results will be a concluding part of the thesis. Possible methods to investigate are including, but not restricted to; Agent-based modelling, Non-linear modelling, Social network models, System dynamics, and NK-modelling.

It is expected that the student in cooperation with supervisors and other relevant persons decides upon the most adequate method to simulate a scenario for actors in the mobile industry. Value chains and value networks that constitute this business are complex, and there are many dependencies between actors. Simulation and visualization of these processes may contribute to an increased understanding of behavior in this business, and how the actors may respond to sudden events and organize themselves to optimize their own market position. Implementation will be done in cooperation with other students, and it is expected that the student (Martin) has a leading role in the progress related to which models to implement and how results are presented/visualized.

Assignment given: 15. January 2007
Supervisor: John Krogstie, IDI

Abstract

Mobile communication markets are known for frequent innovations with potentially high network effects. The conceptual work in economics and innovation studies show how the growth pattern for innovations in such markets could vary depending on the competition and market characteristics. However, the empirical research within this field is limited.

This thesis introduces a computer simulation model for analyzing the development, adoption and diffusion of innovations in a mobile market. The model is based on Agent-Based Computational Economics (ACE), and makes use of behavioral theory of firms, economics, and sociology, to leverage the theoretical understating of the diffusion of innovations.

The results of the simulation runs on the developed simulation software show that the topologies of social networks have strong effects on diffusion. However, it is also found that in situations where several competing companies launch their innovations sequentially, a winner-takes-all outcome is the most likely when the actors are completely rational. Further, when the information in the market is imperfect, the topology of social networks can create equilibria where the market is shared between several providers. Finally, the variance in consumer characteristics is shown to affect both the rate and the outcome of innovation diffusion.

The thesis reaffirms that computer simulation is an effective way to combine the sociological and economic theories of innovation diffusion. The results show that there is still a need for more research on the field to better understand why some innovations fail, while other succeed and becomes accepted in the market. The outcome of an innovation launch is shown to be affected by several factors, including timing, network structure, market noise, and consumer characteristics. By using the simulation model to study the influence of such factors in a specific market, service providers may improve their competitive power.

Preface

This document is the report of my Master's Thesis in Computer Science at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU) in Trondheim. The thesis is written for the Innovation Lab at Accenture Norway.

I wish to thank my supervisors, professor John Krogstie (NTNU) and Øyvind Strømme (Accenture), for providing input and ideas as well as valuable feedback on the research and the writing of the report. Further I wish to thank Babak Farshchian(Telenor R&I) and Ulas Burkay (BI) for their contributions and ideas on the subject.

Trondheim, June 1. 2007

Martin Andreas Børke

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *An* ***innovation*** *is the implementation of a new or significantly improved product (good or service), or process, a new marketing method, or a new organisational method in business practices, workplace organisation or external relations.*

This is how the Oslo Manual[TGG05], endorsed by Organisation for Economic Co-Operation and Development (OECD) and Eurostat, defines an *innovation*. The definition encompasses a wide range of possible innovations, with a minimum requirement that the product, process, marketing method or organisational method must be *new to the innovator*.

Innovation research is often described as a socioeconomic science, with branches in both sociological and economic theories. Most *social science research* proceeds by building simplified representation of social phenomena, often purely verbal. On the other hand, in *economic research* the representation is usually more formal and often expressed in terms of statistical or mathematical equations. There is however a third way of representing the phenomenas, namely through *computer simulation* or *computational modelling*. Gilbert and Terna argue that this approach is "...particularly good at modelling processes and although non-linear relationships can generate some methodological problems, there is no difficulty in representing them within a computer program"[GT00]

A *product* is in the Oslo Manual defined as "a good or a service", and the focus of this project has been *services* in the mobile market. While goods may be more observable in the daily life, services have other characteristics that makes their diffusion in markets unique. For instance, the *cost of reproduction* is often very low. This report describes the development of a model for simulating how new competing service innovations diffuse in a market of consumers, and thereby affect the competition between service providers, based on recognized economic and sociological theory and simulation methods.

1.1 Motivation, goals and contributions

In mobile commerce, voice communication has reached its saturation point several years ago in countries such as Norway where the penetration rates of mobile phones are high[CHRW05]. Operators in these countries complain that they suffer from a declining Average Revenue Per User (ARPU). While the hardware manufacturers see a continuous growth in sales as new equipment is launched, the market for operators and service providers has nearly stagnated, and price competition forces prices down. The mobile industry has for a long time dreamt about the "killer application", the one application which will increase the profit in the industry. To be characterized as a "killer application", a service must (1) generate inordinately high traffic volumes, (2) produce especially high revenue, (3) produce high margins, and (4) drive the construction of sufficient infrastructure to support yet unknown future applications[1]. In addition to messaging, Value Added Services (VAS) are now typically seen as promising for market growth.

There has been performed little research in the field of service innovation compared to *classical* product innovation where the good was physical and easily observable. The main problem is that these two fields can not easily be compared[Küp01]. Well known theories, such as *diffusion theory* and *Technology Acceptance Model*, have also been suspected to be unable to explain the acceptance and future growth of mobile services *by themselves*[CHRW05].

1.1.1 Motivation

The motivation of this thesis is to gain better understanding of competition in innovative markets, and especially in the market of mobile services. Through computer simulation, features of *consumers*, *social networks*, *mobile innovations*, *externalities*, and *firm strategy* can be analyzed to see how they affect competition and innovation diffusion in such markets.

1.1.2 Goals

The main goal of this thesis is to design and implement a simulator for competitive innovation in the mobile market. The simulator will embody the behavior of consumers and service providers, and address the characteristics of the market for mobile services. Further, the simulator shall be able to simulate a variety of scenarios through different input, and it shall have a general interface to allow it to be used in several settings.

[1]From "ARPU is not the best way to measure success" (Visited 2007/03/01); `http://www.3gnewsroom.com/3g_news/jun_03/news_3481.shtml`

1.1.3 Contributions

The main contribution of this project is the elaboration and design of a simulator for mobile service innovation. The developed software will serve as one example of an implementation of the design, and shall serve as a proof-of-concept. Further, the thesis will give thorough descriptions of the consumers, service providers, and market characteristics of the mobile industry - which by themselves serve as a contribution. The thesis also gives a description and evaluation of four simulation methods that may be used to simulate innovation. The extensive bibliography contains references to much of the most recent work done in the field, and will therefore serve as a contribution. Finally, the thesis will give some conclusions based on the simulation of a set of scenarios.

1.2 Research method

The research method of the thesis is built around the incremental development of the simulator. Each increment, or iteration, was performed as a small development project containing all the usual phases: planning, requirements analysis, design, coding, testing, and documentation. This improved the understanding of the domain throughout the project, while ensuring that a working prototype existed at the end of each iteration.

Before the development could start, a thorough state-of-the-art study was necessary to understand the basic mechanisms behind what was to be implemented. This study included innovation theory, economic models, game theory, and simulation. The phase ended with the choice of a simulation method to be used.

With the theoretical background in place it was possible to start the basic design of the model. Design, implementation and testing/verification was done incrementally. In the first round of the design phase, the basic requirements for the simulation was elicited, and hypothesis and research questions were prepared. The first algorithms were also designed along with the basic architecture. Throughout the following iterations, the requirements were revisited, algorithms were improved, and the design was refined.

The implementation phases consisted of implementation of the simulation model and adjustment of the algorithms. For each iteration, a few of the most important remaining requirements were selected for implementation. The basic requirements were implemented first, while refinements were implemented in following iterations. In parallel with this thesis, a project at Norwegian School of Information Technology (NITH) was implemented a surrounding framework that used the simulation engine[2] and presented it through a web interface.

At the end of each iteration, the simulation engine was tested, verified and validated to ensure that it acted as expected and that the requirements were fulfilled.

[2] *Simulation engine* is the word used for the main simulation software developed in this thesis.

The framework was also tested by the NITH students for each increment.

The results of this thesis is this report describing the work and results, and the simulation software developed in accordance to the requirements specification. The research method is illustrated in Figure 1.1, where the pink boxes (*Implementation of Surrounding Framework* and *Test of Framework*)are parts of the external project at NITH.

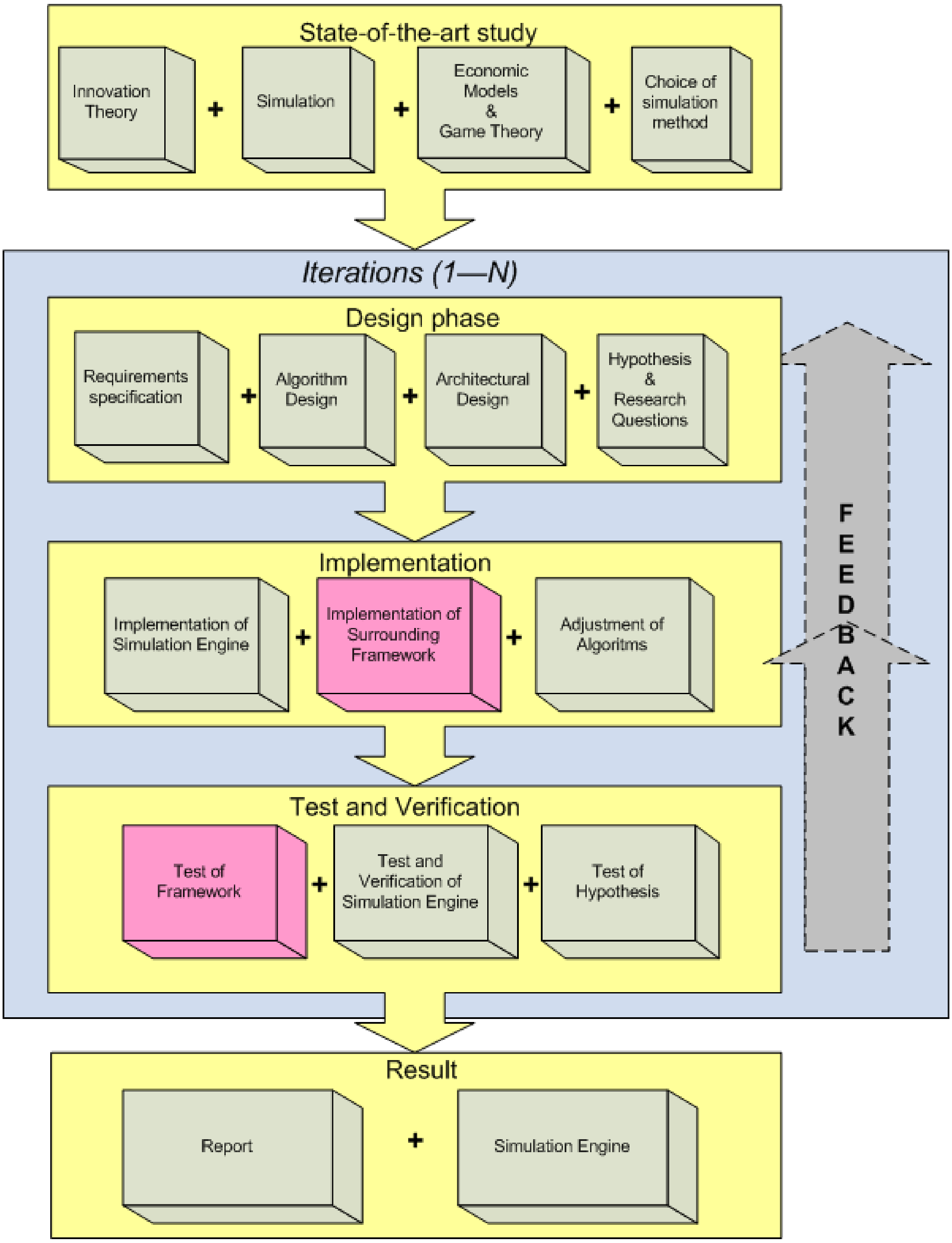

Figure 1.1: Research method

1.3 Report outline

This thesis consists of the following chapters:

Chapter 2 elaborates the problem and describes important aspects of the task. The properties of *network industries* are described, and the innovation model that will be used in the thesis is presented. The chapter describes what is the novelty of this thesis, and concludes with the research questions and hypothesis.

Chapter 3 presents four known methods for simulating innovative activities. The chapter concludes with a choice of simulation method.

Chapter 4 gives the theoretical background for the model that is implemented, where both economic-, computer scientific-, and sociological aspects are covered.

Chapter 5 presents the requirements specification for the implementation of the model. Both an overall description, functional requirements, and quality requirements are presented.

Chapter 6 presents the architecture of the implemented software. The architectural drivers are given along with quality tactics to assure representation of these drivers in the architecture. Further, the architectural description of the software is presented through the "4+1 view model"[Kru95]. To assure understanding of the relationships among agents, their properties, externalities and actions, a separate *agent view* is also presented in this chapter.

Chapter 7 describes the implementation of the software by defining what was implemented in each iteration of the development.

Chapter 8 presents the results of testing of the simulator. The most important findings from simulations are presented here, along with discussions of the results and comparison with similar work.

Chapter 9 gives an evaluation of the results achieved. It presents the contributions of the thesis, and also gives guidelines for further work.

Chapter 2

Problem Elaboration

"If you don't know where you're going,
you're unlikely to end up there."
- Forrest Gump

The mobile market is known to be a highly dynamic market with rapid service innovations. Mobile technology and mobility has made our daily life easier since first introduced in the early 1980s[1], and is now seen as a pervasive part of our lives. As in other economic markets, companies' primary existence is based on the value they can offer to customers utilization of the product or service. However, as mentioned in Chapter 1, Norway has now become a country where most citizens allready own a mobile phone with a subscription. Market penetration through *new* subscribers is thereby difficult, and at the same time price competition between companies has brought the prices down making it even harder to differentiate on price.

This chapter describes some of the characteristics of mobile markets, and thereby puts the foundation for the research questions. Starting with a description of network industries gives insight in what differs mobile services from many other kinds of innovations. Next, the innovation model that is to be used as a basis for the simulation is presented. The environment that is to be simulated is also described. Further, theory on how computer simulation programs can be verified and validated is given. At the end of the chapter the novelty of this thesis is described, and the research questions and hypothesis are presented.

[1]The first fully automatic mobile phone system was the 1981 Nordic Mobile Telephone system.

2.1 Network Industries

The mobile market can be characterized as what we call a *network industry*. Several other markets for goods and services also satisfy the characteristics of this industry, including telephone, email, Internet, computer hardware, computer software, music players, music titles, video players, video movies, banking services, airline services, and legal services[Shy01]. The main characteristics of these kinds of markets, that separate them from other kinds of markets, are:

- Complementarity, compatibility and standards
- Consumption externalities
- Switching costs and lock-in
- Significant economies of scale in production

2.1.1 Complementarity, Compatibility and Standards

The consumers in markets of network industries are shopping for *systems* rather than individual products. For instance, computers would be of no value without software, and a mobile service would be of no use without a telephone subscription, a cellphone and the network supporting the communication. Such products are said to be *complementary*.

In order to be complementary, products must be *compatible*. Software must compatible with the operating system, and the mobile services must be compatible with the cellphones. A keyword in this matter is *standardization*. If the value of a network depends on its size, standardization becomes an important strategic decision[VFS05]. In general, the dominant firms with established networks will prefer not to interconnect, as it may be more profitable to refuse access to their own services.

2.1.2 Externalities

In network industries, the utility of using a good is often affected by the number of other people using similar or compatible products[Shy01]. For instance, no one would use e-mail knowing that nobody else does. These kinds of externalities are **not** found in, for instance, markets for groceries. We call such externalities *network externalities*.

When launching a new technology[2] with network externalities in the market, the outcome depends on the consumers expectations on the size of the network of users. Two typical equilibria that may appear are “all consumers adapt the technology” and “no consumers adapt the new technology”. The more important the network externalities are compared to the intrinsic value of the product, the

[2] I here use the word *technology* for both physical goods and services

more the users expectations and beliefs form the outcome. We often talk about a *critical mass* that must adopt the new technology before it becomes generally accepted. If this critical mass is not reached, the technology will stagnate and eventually die out. However, if reached we see what has become known as the "tipping point" of the technology[Gla01], where the technology suddenly starts diffusing at a very high rate.

2.1.3 Switching Costs and Lock-In

If the user is faced with direct or indirect costs by switching technology or provider, we say that the user is *locked-in*. For instance, changing from a Windows operating system to Unix will lead to costs in the sense of learning to manage the new system. We call such costs *switching costs*, and they can take many forms. In the mobile industry, the providers often operate with *contracts*, binding the customer to be loyal for a certain time after signing the contract. Further, there is a *search cost* related to finding the best alternative. There may also exist *customer programs* that gives long-term users some benefits. Finally, the providers often operate with *binding fees* for new subscribers, which may be seen as an external lock-in effect since it is produced by other providers than the one the consumer is currently subscribing to.

2.1.4 Economies of Scale

Generally speaking, service innovations may be seen as pieces of software. As with other software, they are characterized by a huge sunk cost for the first copy, while the next copies cost almost nothing to reproduce. The research and development (R&D) costs of innovations are often very high, especially if we include the costs of all the innovations that are never launched in the market. When the marginal cost is almost negligible, the average cost function will decline sharply with the number of "copies" sold (see Figure 2.1). This implies that a *competitive equilibrium* does not exist, and these markets will often be characterized by dominant leaders capturing most of the market[Shy01] (also called a "winner-takes-all" equilibrium). An interesting question is then whether or not it is *possible* for several competing innovations to coexist in such a market at all.

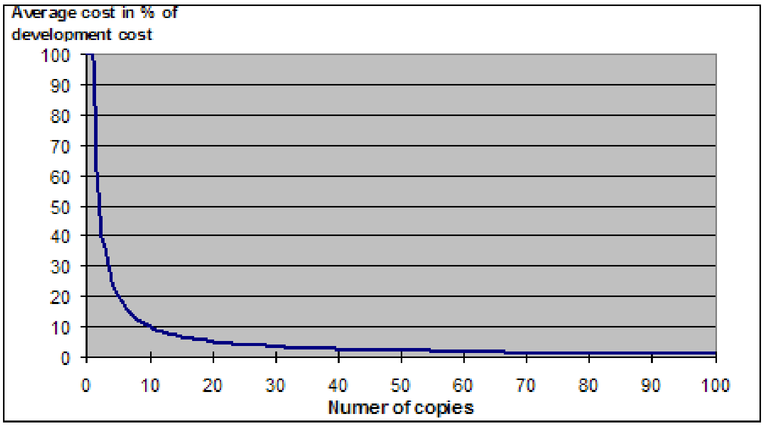

Figure 2.1: Average cost of the innovation

2.2 The Innovation Model

The simulation will be based on the innovation model illustrated in Figure 2.2. The model shows how the company resources are used to produce innovations, some of which are launched in the market based on selection. The decision of which innovations to launch and which to reject is usually based on expectations and current trends. New research tends to explain the development process as *chaotic*[Cm96], whereas classical theory usually described it as *random*. The differences will be further explained in succeeding chapters.

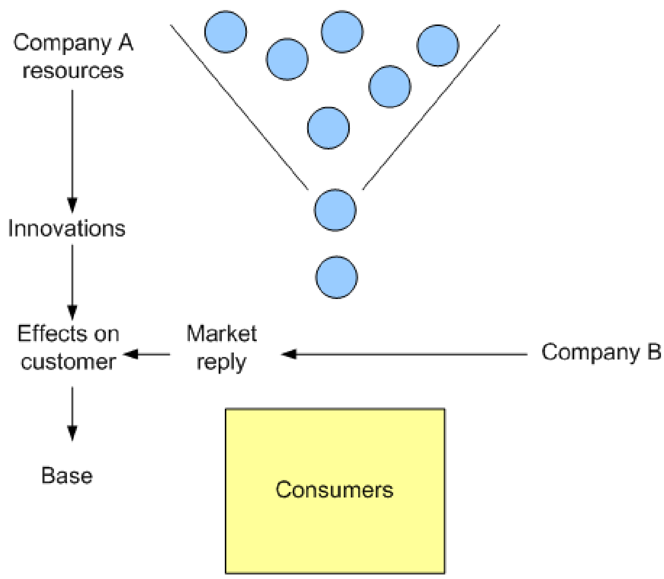

Figure 2.2: Innovation Model

Other companies in the same industry will typically reply to the innovation with competing innovations and thereby affect the market in some way. The competing innovation may be in the same genre as "our" innovation, or it may be another unique innovation which our company does not have access to. In any matter, the consumers will face a decision between the companies based on the value they see in the new innovations. The companies may also use other strategies than innovating, for instance differentiating on price. For some consumers, lower prices may be more attractive than new innovations.

The consumers are usually connected to one or more social networks, through which information and reputation flows. While marketing may affect the consumers knowledge about providers and services, modern theory in social science claim that the final decision on whether to adopt or not is mostly based on information from the social networks. The consumers base their decisions on this information and try to select the technology that give them most value. While some consumers are eager to try out all new technology that is launched in the market, others are more expectant and will not buy the technology until it has become common. The differences in consumer behavior must therefore be included in the simulation model.

2.3 The Simulation Environment

The Norwegian mobile industry is distinguished by a small number of service providers serving a saturated market. Most consumers already have a subscription connected to a service provider, and new subscribers are mostly younger people getting their first mobile phone. Even if the providers have different binding fees, subscription fees and variable costs (price per SMS, unit cost etc.), the share of consumers between the providers seems to be more or less stable with small variances as seen in Figure 2.3.

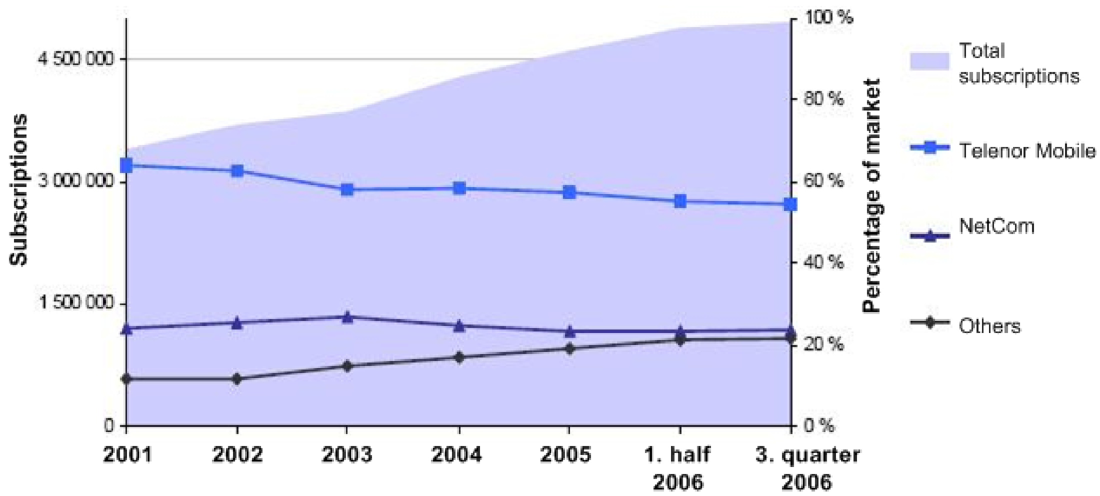

Figure 2.3: Share of subscribers in the consumer market [ot06]

As seen in the figure, the new subscribers seem to be "eaten up" by other providers than the two dominating ones (Telenor and NetCom). However, this is not the whole truth. According to statistics, in average 50.000 subscriptions change provider **each month**. This constitutes 600.000 changes each year, 12.5% of the subscriptions. So what attracts these people to the new provider?

Historically, we can see that the prices in the mobile industry are decreasing on almost all fields. When a new and cheaper provider enters the market and is able to stay there with the same prices, the tendency is that the prices in general are also decreasing. The following assumption is therefore made:

> **Assumption 1: The consumers valuation of a *basic subscription* is a function of the prices for available subscriptions in the market *that the consumer is aware of.***

The last part of the assumption is important, since the average consumer does not have a complete overview of all available providers and subscriptions. However, if this was the only influence on the prices they would decrease much faster than what we see today. Telenor, which is one of the most expensive providers, still holds the largest share of the subscribers. This leads us to the next assumption:

> **Assumption 2: The consumers valuation of *one specific subscription* is a function of the valuation of a basic subscription *and* the valuation of additional services this subscription can provide to the consumer.**

What this means for the providers is that they may maintain their current prices, and may even offer subscriptions with *higher* subscription costs than what they currently do, if they can *innovate with new services*. By adding services to their subscriptions, they also increase the values of these subscriptions.

The described environment lays the foundation for what is to be simulated. We have a limited set of providers which, for simplicity, each hold one type of subscription. The consumers are in general distributed between these providers, but "unbound" consumers may also exist. Each provider can then add new innovative services to their subscriptions to increase the consumers valuation, or they can use other strategies such as decreased prices to hold their market position.

2.4 Verification and Validation

Verification and Validation (V&V) are two important aspects of simulation research. Verification refers to determining that the simulation software performs as intended, whereas validation is concerned with whether or not the *conceptual simulation model* is an accurate representation of the system simulated. The "conceptual model" is the mathematical, logical or verbal representation(s) of the problem entity developed for the study. The model is developed through analysis and modelling of the real-world system that is in focus.

2.4.1 Verification

Verification consists of checking if the computer code contains any programming errors (also called 'bugs'). Kleijnen [Kle95] discusses four ways of verifying the simulation model:

1. Through general good programming practice, such as modular programming, object oriented programming, chief programmers approach, structured walk-throughs, and correctness proofs.
2. Through verification of intermediate simulation output, for instance verification of each module, each increment, etc.
3. By comparing final simulation outputs with analytical results. A simplification of this may be to verify the simulation response by running *simplified* versions of the simulation program with *known analytical solutions*. For instance, one can try to reproduce "textbook theories" on the subject, and compare the results to recognized formulas.
4. Through animation of the results. A dynamic system can be animated through dynamic displays of the simulated system. It is then possible to detect both programming errors and conceptual errors, although the latter concerns validation more than verification.

2.4.2 Validation

Even if the computer code is verified, it is not assured that the simulation behaves as expected; *correct programming is no assurance of correct assumptions*. It is therefore important to check whether the conceptual simulation model is an accurate representation of the real system in focus. The first step toward validating a model is to obtain real-world data. However, this is not always easy and may in fact be impossible or inappropriate (for instance in a simulation of a nuclear war). In the cases where data is not possible to access, it may be shown that the exact values of input and output data are not critical, and that other validation methods may be used [Kle95]. If however some data is possible to obtain, statistical procedures should be used to compare the simulation data with real-world data.

Whether or not real-world data is obtained, a sensitivity analysis may be performed. Such analysis determine whether the model's behavior agrees with "... the judgment of experts (users and analysts)"[Kle95]. We may then find which inputs are really important to the model, and which are less important. By using commonly accepted facts and theories as hypothesis for simulations, one can controll that the simulation behaves as expected.

Finally, many simulation studies typically base their conceptual models on *common sense* and *direct observations* of the real system[Kle95]. We call this *white* *box simulation.* Kleijnen [Kle95] argues that animation is a good means to "... obtain *face validity* of white box simulation models."

2.5 Novelty of Work

In their 1997 article *"Social Network Effects on the Extent of Innovation Diffusion: A Computer Simulation"*[AR97], Abrahamson and Rosenkopf describe how traditional diffusion theories should be enriched by theories on social networks. They explain how the structure of social networks can affect the bandwagon theories, where a positive feedback loop is created as the number of adopters increase.

In the concluding part of the article they state:

"Increasing Returns theories generally assume a context in which an innovation's costs decline or its returns increase with the number of its adopters. ...information is assumed to be available and unambiguous, and it need not to be complicated through a social network before it can prompt more adoptions.
...
We find only one exception to the general claim that social networks cannot enrich Increasing Returns theories as they are currently formulated. This exception occurs when increasing returns are generated by communication networks; the more potential adopters adopt a communication standard or device, such as an electronic-mail facility, the greater the returns to adopters because they can communicate with more adopters."

The last part of the statement refers to theories by Katz and Shapiro[KS85, KS94] and Farrell and Saloner[FS85], in articles describing what is known as *network effects* or *network externalities.*

Further, in the same article they state:

"We also examined only the diffusion of single innovations, rather than of competing variants of an innovation, and we focused on the adoption of innovations rather than their rejections. This focus leaves open several additional directions for future theorizing about social network effects on the extent of innovation diffusion. Future research could explore, via computer simulation, the simultaneous diffusion of competing variants of an innovation across networks with varying structures."

Loch and Huberman partly responded to the latter gap in research through their 1999 article *"A punctuated-Equilibrium Model of Technology Diffusion"* [LH99], where they present an evolutionary model of simultaneous technology diffusion in which both an old and a new technology are available, and thereby competing. They also allowed both technologies to improve their performance incrementally. However, they have not included social networks in their model and thereby do not directly respond to the gap Abrahamson and Rosenkopf pointed out.

Some recent work on the field (see Appendix C) was published December 2006 by Lee, Lee, and Lee[LLL06]. In this article they describe a simulation model for competition between two incompatible technologies with network effects. Through this model they argue that the validity of the "winner-takes-all" hypothesis, popularized for such environments, depends on how customers interact. This article is a direct extension of the work by Abrahamson and Rosenkopf, and address both the effects of social networks on Increasing Returns theories and competing products. In the conclusion of the article they state:

"...An interesting and important extension of our model would be to consider sequential entry, where one technology is introduced first and a new incompatible technology arrives later. We believe such a sequential entry model would provide valuable theoretical and practical implications. Second, we could not extend the analysis beyond two competing models. Consequently, we do not know whether the main findings of this study remain intact in a general case of many competing, incompatible products. Third, we used the Watts-Strogatz model as a representation of diverse social networks. The key assumptions of this model are: All relationships among customers are symmetric and of equal significance; they are assumed to be fixed over time; and no hubs exist."

Further, in their article they consider economic strategies used to gain market share to increase the network effect of the innovations. Since their model does not explicitly specify price, they use the individuals basic willingness to adopt to incorporate such strategies in their model.

This was the most recent work found on the field, and it creates the entrance point for this thesis. By combining the economic theories of innovation diffusion and company behavior with the sociological theories of social networks and consumer behavior, the thesis describes the design and implementation of a model for competition in the mobile market. To succeed the previous work mentioned in this section, the model must incorporate:

1. Subsequent arrival of innovations
2. More than two competing innovations
3. Advanced and realistic modelling of social networks
4. Explicit use of prices

2.6 Research Questions

The main objective of this thesis is to:

> **Design and implement a simulator for innovative competition in the mobile industry.**

However, to be able to fulfill this objective, several *research questions* need to be answered. By combining the answers to all these questions, the main objective may be fulfilled. The research questions for this thesis are:

1. Which simulation method is best suited for simulating innovation in the mobile market?
2. Which economic theories should be included in the simulation model?
3. Which sociological theories should be included in the simulation model?
4. How should the consumers be modeled to best reflect a real market?
5. How should the service providers be modeled to best reflect a real market?
6. How can the actors be parameterized in a way that 1)makes the simulator as realistic as possible, and 2)minimizes the configuration of each simulation?

2.7 Hypothesis

The hypothesis are the *assumptions* that are to be tested through simulation. Several hypothesis should be possible to test through the simulator, but the following are the initial hypothesis that the simulator **must** support:

1. Innovation diffusion will differ when the services are valued through social networks instead of globally.
2. The "winner-takes-all" equilibrium is not the only equilibrium for companies with competing innovations
3. If the companies compete equally through prices, and no innovation efforts are present, no company will stick out as the natural leader in the market.

Chapter 3

Simulation Methods

"*Computers are useless. They can only give you answers.*"
- Pablo Picasso

The first step in designing the simulator, is to choose which simulation method to use. Four approaches that have been used in prior work to simulate different aspects of the innovation process are *System Dynamics*[For61], *Agent-Based Computational Economics*[Tes02], *the NK-model*[Kau93], and *Fractal Landscape Modelling*[CDW07]. This chapter describes the four simulation methods, and concludes with a choice of method.

3.1 System Dynamics

System dynamics is a method to enhance learning in complex systems[Ste00]. It is fundamentally interdisciplinary, grounded in the theory of nonlinear dynamics and feedback control developed in mathematics, physics, and engineerings. When used to study social and economical models, it also draws on cognitive and social psychology, economics and other social sciences.

One of the fundamental skills in the art of system dynamics modelling is discovering and representing the feedback processes. These may be positive (self-reinforcing) or negative (self-correcting) feedback loops. While positive feedback loops tend to reinforce or *amplify* some system behavior, the negative ones counteract and oppose change. Even though there are only two *types* of feedback loops, complex models may easily be composed by thousands of them - coupled to one another with time delays, nonlinearities, and accumulations. When such systems are to be analyzed, our intuition will not be enough, and usually computer simulation must be used to deduce the behavior of the models.

As mentioned, the goal is to enhance *learning* in complex systems. But in the real world, learning is limited by several factors:

- Unknown structure of the real world
- Complexity of dynamics
- Time delays between action and reaction
- Failure of decision implementation
- Selective perception of information
- Ambiguity
- etc.

To be able to disregard some of these limitations, we use what is called *virtual worlds*. These are models of the real world, in which decision makers can conduct experiments and foresee the results of their decisions. Time and space are allowed to be compressed or dilated, actions can be repeated with the same or different conditions, and dramatical strategies can be tested without any risks.

Several modes of dynamic behavior exist, and Sterman [Ste00] mentions exponential growth, goal seeking, oscilation, and process point as the most fundamental. Each of these are generated by simple feedback structures. When nonlinear interactions exist between the fundamental feedback structures, we experience other common modes of behavior including "S-shaped growth", "S-shaped growth with overshoot and oscillation", and "overshoot and collapse"(see [Ste00]).

Modelling the system dynamics is an important step in the process toward learning. However, simulation is the only practical way to test these models. Typical conceptual models are way too large and complex to simulate mentally, and without proper simulation the only way to test the models are through the real world. For important strategies and decisions, the latter is of course not preferable. While the advantages of simulation may seem obvious, there are also people that don't believe that human behavior can be simulated. The main argument against simulation of human behavior is that there are no reliable laws of human behavior that can be compared to those of ,for instance, physics. The counterargument to this view is that it overestimates our understanding of the nature and underestimates the regularities in human decision making.

3.2 Agent-Based Computational Economics

Agent-Based Computational Economics (ACE) is the computational study of economies modeled as evolving systems of autonomous interacting agents [Tes02]. The approach starts by defining and constructing an initial set of agents (the population). The term "agent" is here used to describe self-contained programs, or parts of programs, which can controll their own actions based on their interpretation of the environment. These may be both economic agents, and agents representing some kinds of social or environmental phenomena. The former are for instance traders, financial institutions, etc., while the latter may be a government, land, and so on. The agents are initialized by specifying the attributes ruling their behavior, before the economy is “set free” to evolve over time without further intervention from the modeler.

By building computational laboratories it is possible to create artificial economies where the agents are acting independent and without assistance from the modeler. The models can easily be tailored to suit specific research agendas, or to test some theories and hypothesis. The agents themselves act according to some set of internal rules, and they may learn to optimize their own behavior by implementing memory. It is also possible to create evolutionary scenarios, where new agents inherit the knowledge and the best behavior of the former ones. One example of a conducted experiment is the one of McFadzean et al.[MST01], which uses a computational laboratory to investigate “...evolutionary trade network formation among strategically interacting buyers, sellers and dealers”.

One of the major benefits of ACE in comparison to System Dynamics is its grounding in the interactions of autonomous adaptive agents[Tes02], including economic, social, and environmental entities. This means that the dynamics of the model is governed by the interaction amongst these agents, not merely by systems of equations. This more closely resembles the sociological view of economic models, and allows the researcher to more closely inspect the reasons for each macro level effect through the internal states of each agent (micro level). ACE also allows for relaxation of some of the assumptions of typical economic models, such as *common knowledge*, *rational expectations*, and *perfect capital markets*. However, as each simulation may lead to different results, it is important to use statistical methods to ensure the robustness of the findings.

3.3 The NK-model

Technological innovation can be seen as a complex system composed by elements and their interactions. For instance, to use the example provided by Frenken [Fre01], a particular artefact as a vehicle technology can be described by the elements *engine*, *suspension*, and *brake*, with respective alleles[1]. In a technological system, there is usually more than one option for each element. If we assume two options for each element, the number of possible combinations (designs) adds up to 2^N, where N is the number of elements in the system. Through this view we can see a technological innovation as the designer moving from the current location in design space to a new one, preferably with better performance or ’fitness’.

The NK model was first proposed by Kauffman [Kau93], and was originally developed as a model of biological evolution (hence the use of the word *alleles*). In the

[1] Allele is the word used by Frenken[Fre01] to describe a combination of properties (options/choices) for each element. (The real meaning of the word is any one of a number of viable DNA codings that occupies a given position on a chromosome.)

NK-model we use the metaphor of a *fitness landscape*, which is the mapping of fitness values to the possible combinations of elements. In the sense of technological innovation, the fitness can be seen as the efficiency or quality of the artefact being studied. Each design can be labeled by a *string* of alleles $s_1 s_2 s_3 ... s_n$, and the sum of all these strings forms a set S which is the complete design space. The number of alleles for element n is noted A_n, and it is usually set A=2 for all elements for simplification in illustrative examples.

The complexity of a system comes from the dependencies between the elements constituting it. Kauffman calls the dependencies between elements in complex systems "epistatic relations"[Kau93]. These relations means that a mutation in one element affects the functionality of the related elements as well as the element itself. Kauffman uses the parameter K to define the number of elements possibly affected by a mutation of one element[2]. We can construct a fitness landscape by drawing randomly the fitness of an element w_n from the uniform distribution between 0 and 1 each time the element itself is mutated, or a epistatically related element is mutated. The total fitness of a string s is then defined as $W(s) = \frac{1}{N} \cdot \sum_{n=1}^{N} w_n(s)$. The landscape will now consist of peaks called local and global optima. Local optima are strings where there exist no neighbouring[3] string with higher fitness. The global optima is the string with the highest fitness in the landscape. In the context of technological innovation, two properties of the NK fitness landscape are especially interesting [Fre06, Kau93]:

- The number of local optima increases with increasing K and with increasing N. This means that the probability of the found optimum during a search being local also increases.
- The mean fitness of local optima is highest for systems with a positive but moderate complexity (where $2 \leq K \leq 4$). This reflects that higher K-values "...pose increasingly more incompatible design constraints resulting in poorer performance of local optima"[Fre06].

With regards to innovation, Altenbergs [Alt94] model of constructional selection may also be highly relevant. This model considers the inclusion of *new* elements, not only changes in the properties of existing elements. This may be seen as the introduction of an entirely new technology into the existing model, and the model may now be used to see how this affects the system as a whole. Also, in technological evolution, design spaces usually enlarge over time as the number of elements that constitue an technological artefact typically increases. For instance, a car today consists of more elements than one did a few decades ago. Following the presented logic of fitness landscapes, the introduction of a new element will generate new fitness values w_i to all the elements affected by the new element. Altenberg showed that the probability of increasing the total fitness of a string by adding a new element is *inverely related* to the number of elements affected

[2]For instance, if N=3 and K=2, then all mutations will affect both the mutated element and all of the other elements.

[3]Neighbouring here means a string that shares all alleles except one.

by the new element[4]. The more elements affected, the more elements must also be assigned new fitness values - with mean value 0.5. If only a few elements are affected, it becomes more likely to obtain a "lucky" draw and thereby give a new total fitness value for a string that exceeds the old one.

3.4 Fractal Landscape Modelling

The Fractal Landscape Model [CDW07] approach is similar to the NK-model in that it also uses the ruggedness (landscape) metaphor. However, this model takes the metaphor more literally, by actually representing the landscape as a surface over a plane. Landscapes are generated by use of technologies from fractal geometry, characterized by a small number of parameters. An advantage compared to the NK-model is that the landscape may be easily visualized since it is restricted to three dimensions (compared to the N dimensions of the NK-model). Further, it offers the notion of *preferred direction* of a search. The model in addition includes several "natural" considerations:

- Search is represented as costly
- The payoff for a given firm also depends on the number of other firms and what they have found
- The "tightness" of the environment can be varied (from "winner takes all" to highly forgiving toward inferior performance).

An experiment conducted by Cattani and Winter [CDW07] used the *random midpoint-displacement algorithm* to generate the landscape, but the authors indicate that other algorithms may also be used. The paper however focus its discussion around the parameters used in the mentioned algorithm, and explains how the landscape is generated according to these. In this landscape, the "sea level" may be seen as the level of fitness that is just at the threshold of "ecological" fitness[5]. At this level there is no market feedback, comparable to an early stage of the emergence of a new technology.

In the model, each firm is described by its x,y coordinates. Each firm is also assigned a direction vector which is the firms preferred direction to pursue when performing a search. This also allows us to model the heterogeneity in initial conditions across firms, since a firm with a direction vector pointing toward a global maximum has better initial conditions than one pointing another way. The vectors are assigned randomly, usually within a wedge or with some standard deviation. Further, to capture the role of market feedback, each firms relative shares in the current total payoff is normalized on current maximum fitness. More concrete, a firms "share of the pie" is computed as $[F(i)/Fmax]^p / \sum_n^{i=1}[(F(i)/Fmax)^p]$

[4]According to Frenken[Fre06], this can be understood from the disruptive effect a new element has when it affects several other elements.

[5]Cattani and Winter[CDW07] use the example that of "...a productivity level just low enough so that even a monopolist could not find a buyer".

where p provides a means of studying different degrees of "tightness" in the environment. A high value of p corresponds to a "winner-takes-all" environment.

Through adjusting the parameters of the model, one can simulate different kinds of environments and see how the firms behave in these. The parameters are not all described in this section, since a thorough specification of them would be too comprehensive. However, the results from the simulations performed by Cattani and Winter show that the simulation approach can "*...explicitly model the dynamic interplay between firm heterogeneity (e.g., pre-adaption), foresight and market feedback*"[CDW07].

3.5 Choice of Simulation Method

A *model* represents our understanding about a phenomenon, in this case innovation diffusion in a market. The model links the different aspects of the phenomenon together, showing how they influence each other. Classic economic theories usually assume a mathematical relationship between the aspects at an aggregated level, and do not directly model the actual phenomenon and the underlying logic.

This thesis is intended to use low-level theories about the phenomenon to build up a higher level understanding. To enable this kind of approach the simulation method must allow for the following:

- We **cannot** assume that consumers in the market are completely rational
- We must take **uncertainty** into account
- We must be allowed to model low-level relationships, for instance between consumers
- We must be able to abstract analysis of the behavior from the model

After examining the four methods, the choice has fallen on ACE. Instead of creating a simple mathematical model, the agent-based approach bases its underlying model on the interactions of various interacting agents. This also seems more natural, since both economic- and sociological theories deal with processes that *are* results of individual agents actions and assumptions. The choice is amongst others based on the following observations:

- ACE is less predictable than System Dynamics, and better reflects a market where uncertainty and choices are important factors.
- The NK-model and Fractal Landscape Modelling allows us to model the innovation process and the behavior of firms. However, they do not allow for a thorough analysis of the actors in the system.
- ACE better allows us to specify the behavior through parametrization of the actors/agents.

The remaining part of this thesis will thereby focus on the creation an agent-based model for simulation of innovation diffusion in mobile markets. The next chapter will look further into the micro-level assumptions and economic and sociological theories that will be used as a foundation for the model.

Chapter 4

Theoretical Background

> "*The first step towards knowledge is to know that we are ignorant.*"
> - Richard Cecil

Based on the findings of Chapter 3, the simulation will make usage of *Agent-Based Computational Economics (ACE)*. This means that we must define the *agents* of the economy, the *environment* they operate within and their *properties*. Together, this will serve as the *conceptual model* of the simulation.

This chapter starts off by taking a closer look at ACE and its relation to the challenge at hand. Then some theory on diffusion of innovations is presented as a foundation for the further work. Further, the behavior of the agents (consumers and service providers) along with the characteristics of the market are presented. The chapter ends with a short summary of the agent properties.

4.1 Agent-Based Computational Economics - Revisited

The term *"agent"* may lead to some confusion, considering the various usages of the term in modern computer science. *Agents* are used to search the Internet, as "wizards" in application programs such as word processors, and in many other contexts[GT00]. In Agent-Based Computational Economics, and in *Multi Agent Systems* in general, the term is used to describe processes implemented on a computer that have:

- Autonomy (controll of their own actions)
- Social ability (interact with other agents)
- Reactivity (can perceive their environment and respond to it)

- Pro-activity (able to undertake goal-directed actions)

The first step in modelling an economy through agents is to "*... define the cognitive and sensory capabilities of the agents, the actions they can carry out and the characteristics of the environment in which they are located*"[GT00]. This is what *this* chapter is trying to solve. The next steps are to design and implement the software that can simulate this model, and finally to observe the results of the simulation to see whether some phenomenas emerge.

4.1.1 Agent Construction

An agent can usually be designed as a production system consisting of three parts:

1. A set of rules
2. A working memory
3. A rule interpreter

The *rules* consist of i) a *condition* specifying when the rule is to be executed, and ii) an *action part* to determine the consequence of an execution. Further, the *memory* is used to maintain information about previous and current states, and may be used for learning. The *rule interpreter* uses the rules and the memory to check whether the conditions for the rules are present, and possibly performing the indicated actions.

4.1.2 The Agents Environment

The agents should be located within some sort of *social environment* consisting of one or more networks of interactions. The networks may be direct (friends, colleagues, neighbors) or indirect (marketing, rumors, news). These networks set the limitations for spread of information between the agents.

4.1.3 Conceptual Model of the Agent Economy

Figure 4.1 shows a conceptual model of the economy to be simulated and the agents operating in it. The consumers and service providers are agents fitting in to the definition given earlier in this chapter. The innovations cannot be considered as agents; although their properties may change over time (quality, maturity etc.) they do not act autonomously, nor do they interact with other agents by themselves. They are even so included in the figure to show how they fit into the larger picture.

The **consumers** are the agents within the economy that *purchase* and *use* the services/innovations. As in the real world we cannot assume that all consumers

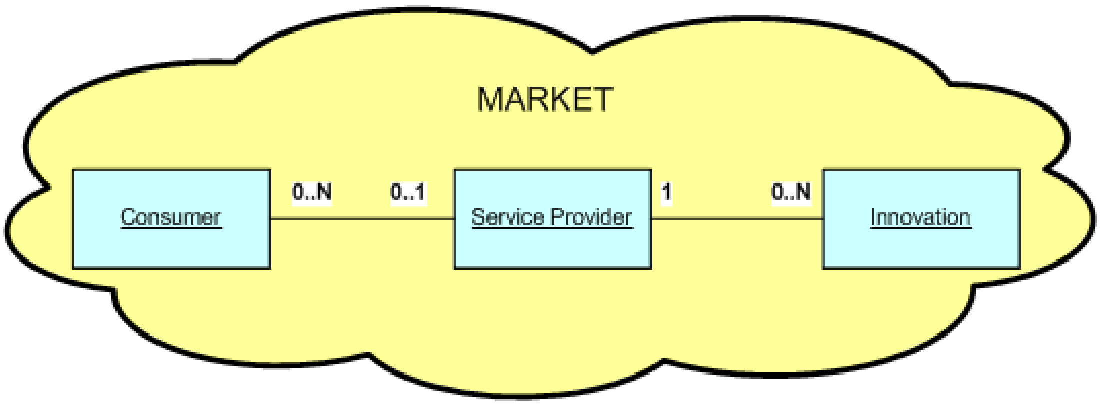

Figure 4.1: Conceptual model of the economy

behave equally, and must therefore introduce theories that describe how the consumers are scattered in a real-world economy. Further, we shall use these theories to disperse the consumers *behavioral models*. The *main task* of the consumers in our economy is to make decisions that gives them the best possible benefits at the time of the desicion.

The consumers are attached to zero or one **service provider**. The service providers are the agents innovating with new services which they launch in the market. The autonomous service provider agents will respond to trends in the market and try to optimize their own behavior according to performance parameters and selected strategies. Each service provider will have its share of the consumer market, but may also be pushed out of the economy leaving it with no consumers. A consumer that is bound to a service provider is said to be a *subscriber* of that provider.

The service providers will create **innovations** to attract the consumers. The consumers will then base their decision of which service provider to use on the properties of the service providers *and* their innovations.

As in the real world, all information goes through **the market**. The important information in our economy is the information affecting the decisions of the agents. Some information is relatively transparent, such as the cost of subscription establishment. This information is generally available through the marketing of the service providers. Other information, such as the number of other consumers subscribing to a specific service provider, is generally not as easily accessible. The differences in the quality of information must be reflected in the simulation.

4.2 Diffusion of Innovations

Diffusion is the process of acceptance of new technologies or ideas in a market. It is rather unrealistic to expect that a new technology will be adopted by an entire market at once, but rather it goes through stages of adoption based on how fast the news spread from one consumer to the next. Several models try to explain how technology spreads in a population. Adherents of the "trickle-down effect" claim that user adoption is affected by the fact that new products are so expensive at first that only the most wealthy people can afford it. When the product in time becomes cheaper, larger parts of the population can afford it. However, this theory is not directly applicable to innovation in the mobile industry since other price strategies are often applied as we shall see in Chapter 4.4.2. Also, in our case the *costs* of producing the innovation lies mostly in front of the launch of the innovation, and producing a new copy of the service is virtually free.

A model that suits better for our use is the "Diffusion of Innovations theory" by Everet Rogers[Rog03]. This theory claims that there are five categories of product adopters: *Innovators*, *Early Adopters*, *Early Majority*, *Late Majority*, and *Laggards*. The population is separated into the categories as illustrated in Figure 4.2. The curve in the figure is called the Technology Adaption Lifecycle, originally developed in 1957 at Iowa State College[1]. This was the curve that Rogers based his theory upon six years later[2].

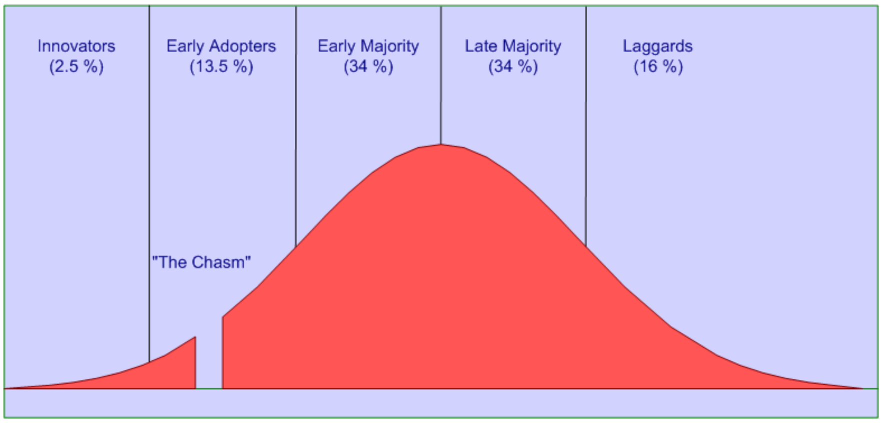

Figure 4.2: Technology Adaption Lifecycle

[1]The *original* purpose of the model was to track the purchase patterns of hybrid seed corn by farmers.

[2]Figure 4.2 differs from the original curve in that it includes a "chasm" between innovators and early adopters. In the book *Crossing the Chasm*[Moo02], Moore argues that visionaries and pragmatists have very different expectations. This difference leads to the chasm, or gap, that is presented in the figure. However, the theories of Moore are only valid when dealing with *disruptive* or *discontinuous* innovations. When the innovation is continuous, the chasm can be removed - leading to the original Technology Adoption Lifecycle.

4.2.1 Adopter Categories

Each of the adopter categories consists of individuals who share some characteristics, but most importantly they share the *level of innovativeness*. Innovativeness is the degree to which an individual is "... relatively earlier in adopting new ideas than other members of a system"[Rog03]. However, constraining the consumers in our model to behave according to time constraints would not make sense as this would only *provoke* the expected results. Another approach that is used in simulations is to give each category a unique threshold for adoption, as in [MF02], but this still feels like provoking expected results. Finally, as mentioned in Chapter 1, the theory is not likely to be able to describe the diffusion of service innovations by itself.

A more interesting approach is thereby to define the consumers according to what distinguish the categories, and then observe whether or not this will produce the expected results when the consumers are acting as autonomous agents. In this way we make use of the fundamental findings of the theory (the adopter categories), while being independent of the conclusions. The following sections present some of the *general* characteristics of the adopter categories, while Appendix D presents more specific characteristics.

4.2.1.1 Innovators

The innovators are almost obsessively venturesome. Their interest for new technology and ideas lead them out of their local social network and into more cosmopolite social relationships. They usually have substantial financial resources to absorb the potential losses from investing in unprofitable innovations. The innovators also have the ability to understand the technical aspects of the innovations, and are thereby better suited to evaluate the possible utility or value.

4.2.1.2 Early Adopters

Early Adopters are more integrated in the local social networks than the innovators. This category also has a large influence on other people, as members of later categories often look to early adopters for advise and information. The members of this category are highly respected by their peers, and know that they must make well-considered decisions to keep this respect.

4.2.1.3 Early Majority

The early majority is the group of individuals adopting the innovations just before the average members of the system. They interact frequently with their peers, but they don't have the same respect as the early adopters in convincing other individuals. The time from knowledge of an innovation to actually adopting it is longer for this category than the innovators and early adopters.

4.2.1.4 Late Majority

The late majority adopt the innovation just after the average member of the system. The adoption may both be a result of economic necessity, as well as of increasing peer pressure. To be willing to adopt, these individuals require that most of the members of their social network have already done so.

4.2.1.5 Laggards

The laggards are the last people in the social system to adopt the innovation. They have smaller social networks than individuals in the other categories, and their limited resources require them to be certain that the innovation will not fail.

4.2.2 Criticism

A counter-argument to diffusion theory is that diffusion research traditionally has a pro-innovation bias[CHRW05]. The theories are based mainly on successful examples of innovation and diffusion, since our current knowledge on ignorance and rejection is limited. However, since the conclusions from the theory are not used, but merely the statistical findings related to consumer categorization, this should not be a weakness for this thesis. Further, some argue that the same person may be an early adopter of one technology, but a late adopter of another one[3]. The innovations of this simulation model are of similar nature, and hence such ambiguity in categorization should not influence the model.

[3] See for instance `http://www.technobility.com/docs/article032.htm` (Last visited 2007/05/28)

4.3 Consumer Behavior

The diffusion theory described earlier is often recognized when *one* new innovation is launched in the market. However, in our case the innovation is not necessarily launched in isolation. What we deal with is a set of innovations that compete for the same consumers. In the end, it is the consumers that choose between the service providers they have knowledge of, based on an evaluation of their services. We therefore need to further define the decision process and behavior of the individuals. This section describes how the consumer agents are profit driven but may be unable to optimize because of bounded rationality and noise.

4.3.1 Valuation of Subscriptions and Innovations

Mahler and Rogers[MR99] points out that *"...an individual adopting a telephone today knows that almost all other households in the nation already have telephone service. Thus the utility of adopting depends almost entirely on factors internal to the individual, rather than such of externalities..."* This means that we should use measurements *internal* to the consumers to evaluate the valuation of a *basic subscription*, since the same statement is true for the basic services of mobile subscriptions today (call service, SMS, MMS, etc.).

Assumption 1 and 2 in Chapter 2 stated that the *basic utility*, or *intrinsic value*, of having a phone subscription is perceived as a function of the prices for subscriptions *known to the consumer*. If the consumer knows only of one subscription possibility with price X, the consumer will assume that the value of owning such a subscription is X. However, if the consumer gets to know another subscription which has price Y<X, then the assumed value will be somewhere between these prices.

In addition to the basic subscription, the service providers can innovate with new services to add to their subscriptions. Mahler and Rogers argue that for *interactive telecommunications services* that are *new* (the innovations), the adoption by other consumers is crucial[MR99]. Such services give more value to a consumer the more of his social connections are allready using the service. We can then distinguish between two basic types of services:

1. Services that have *network effects*, where the value increases with the number of users (interactive)
2. Services that *add value*, regardless of the number of users (non-interactive)

Network effect utility vary from the basic utility of a good, and need not to be present at all. In some cases, such as with the fax machine, the network effect utility is often seen as the main utility of the good. In other cases, such as for instance *food*, network effects are irrelevant. In the mobile services industry network effects are highly relevant and are often the main reason why people adopt new technologies and services[4].

The following sections will describe the two kinds of innovations.

4.3.1.1 Innovations with Network Effects

For some technologies the utility of buying the technology depends on the number of people already using it. For instance, a fax machine would be useless if you were the only person owning one. We call these kinds of effects “network effects”, and we can distinguish between direct and indirect network effects [KS85, VFS05]:

Direct network effects are the effects resulting from compatibility between system elements. They are direct physical effects of the number of purchasers on the quality of the product. As an example, the number of users of the relatively new 3G mobile network in Norway will have a direct effect on the quality; for instance new users can only perform video conversation with other users already in this network.

Indirect network effects are the effects coming from an assumed positive dependency between the spreading of a technology and the appropriate offers of complementary goods. One example is the market domination Apples music player *iPod*, which has now gained an enormous base of complementary products by other vendors (clothing, shoes, car audio, etc.).

We can now introduce the *network effect factor* **Q**, as proposed by Buxmann [Bux01], to measure the strength of the network effect utility (**C**) in relation to basic utility(**b**) of a subscription:

$$Q = \frac{C}{C + b} \tag{4.1}$$

The higher number we get on Q, the more important are the network effects compared to the basic utility of the subscription. Service providers may increase their Q by innovating with new services that gives value to users according to the number of other users adopting them.

Further, if the service provider offers **N** services, the total network effect factor Q_{total} of the services will be calculated as:

$$Q_{total} = 1 - \prod_{n=1}^{N} (1 - Q_n) \tag{4.2}$$

[4]During the writing of this thesis, I actually received a call by a salesman for a new service provider. This service provider offered free calls between its subscribers, and would have the same subscription fee as my current subscription. However, based on the fact that none of my friends were subscribing to this provider, the *network effects* that the “free call” service added to the subscription were not present and I rejected the offer.

which gives Q_{total} a value between 0 and 1, increasing as more innovations are added. Equation 4.3 is then used to calculate the network effect utility c_{total} for the innovations. Figure 4.3 shows the increase in the total network effects factor Q_{total} when several services with network effect 0.2 are added.

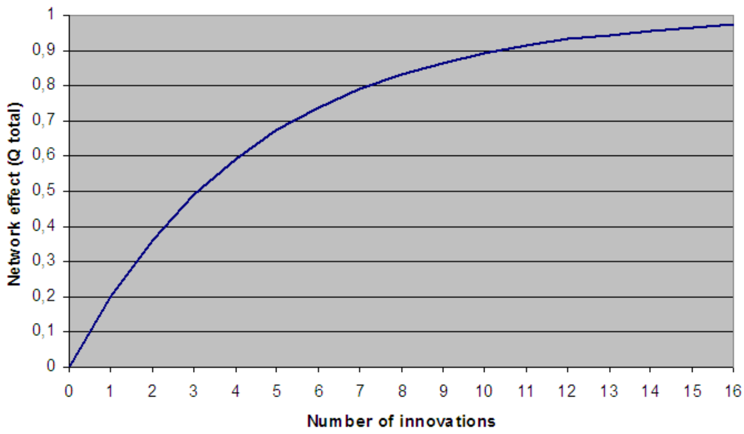

Figure 4.3: Increase in total network effects factor when more services are added

When there are totally **N** users in a consumers social network, the network effect produced by each existing user of the services is calculated as (see [Bux01]):

$$c = \frac{Q * b}{(1 - Q)N} \tag{4.3}$$

From this we can make the following formulation of a criteria for valuation of a service provider[5]:

$K_{x\tau}$**:** The benefit of subscribing to provider x in period τ

$p_{x\tau}$**:** The binding fee the user has to pay to provider x in period τ

b_{xt}**:** The intrinsic value (basic utility) the user perceives from having a subscription of provider x in period t

c_x**:** The network effect utility of the services of provider x for each user of the technology

k_{xt}**:** The subscription fee for subscription x in period t

$n_{x\tau}$**:** The number of users[6] of the services of provider x in period τ (the consumer assumes this to be stable)

[5] The terminology and formulation is based on the work by Buxmann[Bux01]

[6] One can vary the formula by letting the consumers assume that other consumers that have not selected technology yet will select this technology. This will make n_{xt} equal to the current users of the technology *plus* the "unbound" users. Another variation is to only include the users within this consumers direct social network.

r**:** The calculation interest rate

T**:** The planning horizon of the consumers

$$K_{x\tau} = -p_{x\tau} + \sum_{t=\tau+1}^{T} (b_{xt} + c_x * n_{x\tau} - k_{xt}) * (1 + r)^{-(t-\tau)} \tag{4.4}$$

When deciding between two or more service providers, the consumer will now select the provider x which has the highest benefit if this benefit is above zero. If the consumer is already using provider x at time τ, then $p_{x\tau}$ will naturally be 0.

4.3.1.2 Innovations with Intrinsic Value

As mentioned earlier, not all services will have network effects. Some services that are added to a subscription give value on their own, regardless of the number of other users. For instance, adding mail-reading capability to your mobile phone will not necessarily produce network effects since this is a part of an already diffused innovation (mail reading capability).

The problem with such services is that we cannot easily anticipate the value they give to the consumers. To include such innovations in the model, the following assumption is made:

> **Assumption 3: Innovations with intrinsic value increase the consumers perceived intrinsic value of the subscription with a certain percentage.**

This assumption allows us to incorporate the value of these innovations in our model by increasing the intrinsic value of the service provider owning the innovation. We say that each value adding service has a "value increasing factor" **V** between 0 and 1. The valuation(**v**) of each value adding service is then:

$$v = V * b_{xt} \tag{4.5}$$

and the total valuation (v_{xTotal}) of all N value adding services of provider **x** is:

$$v_{xTotal} = \sum_{n=1}^{N} V_{xn} * b_{xt} \tag{4.6}$$

Equation 4.4 can then be revised to include value adding services. The following equation will be used for valuation of a subscription, which includes both kinds of services:

$$K_{x\tau} = -p_{x\tau} + \sum_{t=\tau+1}^{T} (b_{xt} + v_{xTotal} + c_x * n_{x\tau} - k_{xt}) * (1 + r)^{-(t-\tau)} \tag{4.7}$$

4.3.2 Payback of Investments

Simulation of the consumers personal economy is not intended to be a direct part of this thesis. However, we can make some assumptions about their financial *behavior*. One such assumption is that they require that their investments have shown profitable before considering changing provider. As shown in Figure 4.4, an investment in our case starts with paying the binding fee in period 0.

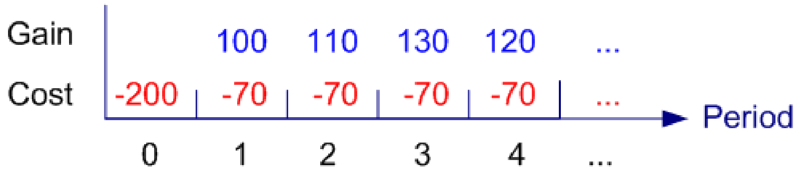

Figure 4.4: Payback of investment

The following periods the user receives some virtual gain from the investment, and also pays a subscription fee. The gain is the benefit as perceived for the consumer. Although this is not easily calculated, we can assume that it takes into account the network effects of the period as well as the perceived intrinsic value of owning the service. The latter may change during time, and is thereby not assured to be the same as at the time of investing. If the consumer gets information about cheaper providers *after* investing in the chosen provider, the perceived value will decrease.

The *payback period* is the time required for the return on the investment to "re-pay" the sum of the original investment. We can assume that the consumer is not searching for alternatives in this period. This will put limitations on the consumers economic capabilities, since the model does not contain information about the economic strength of each consumer. Without such restrictions, all consumers would be completely economically independent, able to change to whichever provider that is best *at the moment*.

4.3.3 Imperfect Information and Noise

As mentioned introductorily in this chapter, all information between service providers and consumers go through the market. Fisher Black states in his article *Noise* [Bla86] that "*Noise makes financial markets possible, but also makes them imperfect*". What he means by this statement is that without noise, the *imperfection* of information in a market, there would be very little trading in individual assets. In financial markets, we seldom see *win-win* situations - and thereby we can assume that one parts outcome of the trade is worse than the other parts outcome. If both parts had perfect information, the part coming worst out of the trade would not perform the trade in the first place.

What this means for us is that without noise, we could assume that all consumers would always choose the service provider giving them the highest utility or profit (hence a "winner-takes-all" situation). Through observing the real world, we instantly see that this is not a correct assumption. Even for small products with almost zero quality differences, like bottled water[7], there is still a market for price differentiation. By adding noise to our model, the consumers will observe the information in a way that better reflects the real world. Tastes, preferences, goal ambiguity, and other factors affecting the decisions in "irrational" ways (see March [Mar78]) are then encapsulated in the model.

Returning to the decision algorithm of Equation 4.7, we have the situation illustrated in Figure 4.5.

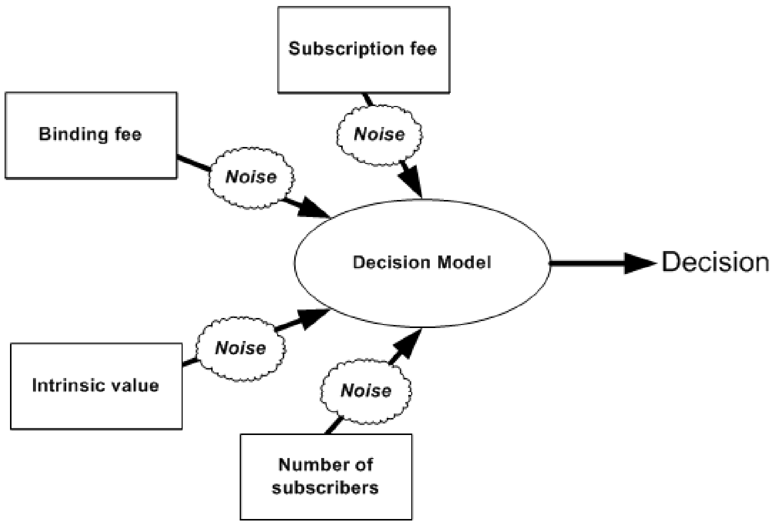

Figure 4.5: Noise in the decision process

4.3.3.1 Noise Related to Number of Subscribers

The network effects are calculated based on the number of other consumers using the service. However, there are two ways of implementing this. Traditionally, work in this field have used the *total* number of consumers in the market. Another approach, announced by amongst others Abrahamson and Rosenkopf[AR97], is to use the *social network* of the consumer as the basis of calculation. These two cases create different needs for implementing noise.

The first case, where the consumers in the calculation are *all consumers* in the market, has a problem that companies seldom announce the number of subscribers publicly. We can assume that the more information sources the consumer has, the more accurate information about the size of the subscriber base he or she has. This means that innovators will typically have better information about the number of subscribers of each service provider at a certain point of time. "Laggards", described as having neighbors and friends as main information sources, will typically have more noise added to the estimate. As visualized in Figure 4.6, the noise factor widens the "sample space" for the estimates of the consumers.

[7]This is not meant as an offense to the producers of bottled water, but is intended as an example of products with very small *quality differences* perceived by the consumers.

Laggards will assume that the number of subscribers is somewhere between zero and the real number, while innovators will have more accurate estimates.

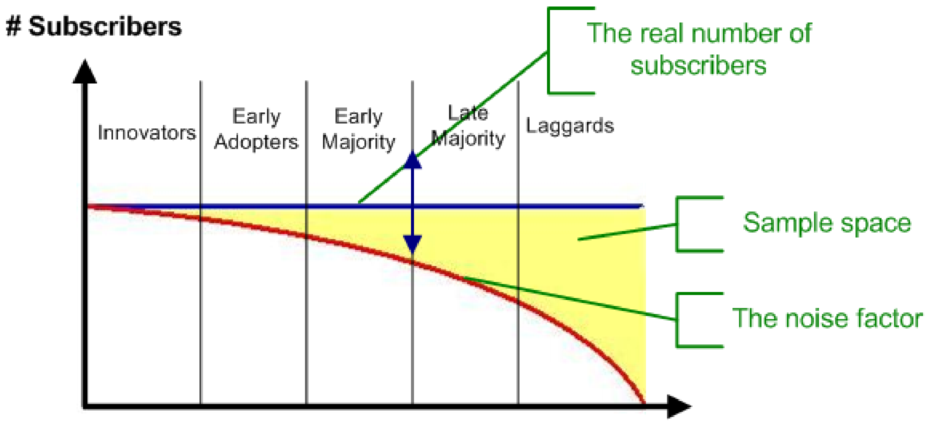

Figure 4.6: Noise related to number of subscribers

In the other case, where the basis of calculation is the social network of the consumer, we can assume that the information is not affected by noise. However, in both cases we must assume that only providers and services that the deciding consumer has *knowledge of* are considered in the calculations.

4.3.3.2 Noise Related to Intrinsic Value

The **intrinsic value** is the assumed *value*, *utility* or *benefit* the subscription would give the consumer by itself, regardless of other services and the number of other consumers subscribing to it. This value is hard to estimate, but is still an important property of the decision process. In this model, the assumption is that there exist an average assumed intrinsic value which is reasonably close to the "real" value. This value is, as mentioned earlier in this report, calculated as a function of the subscription rates of the known providers.

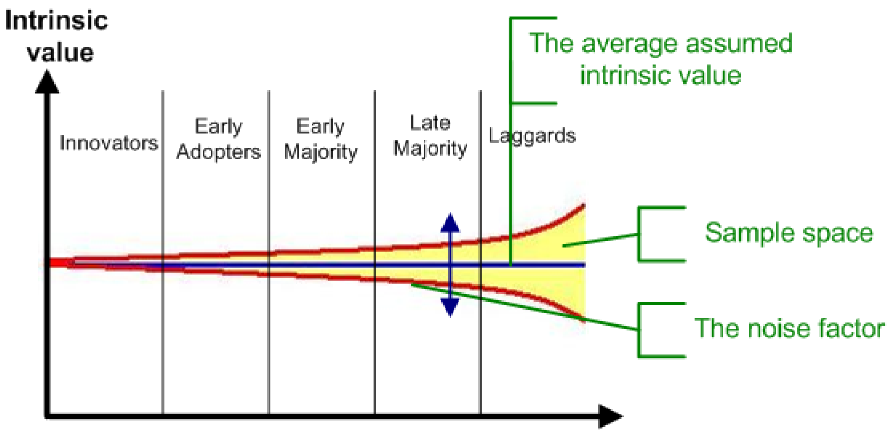

Figure 4.7: Noise related to the intrinsic value

The *more* and *better* information a consumer has about possible providers, their services (innovations) and the market, the more capable he or she is to make a correct estimate. This means that "innovators" will estimate more "precise" values, while the "late majority" and "laggards" will usually either under- or over estimate the values (see Figure 4.7). Another possible noise factor to bring in here is that heavy marketing may increase the possibility of over valuation, while modest marketing may lead to an increased possibility of undervaluation.

4.3.3.3 Noise Related to Subscription- and Binding fees

As mentioned earlier, the costs of the subscription is usually well known through advertisements and marketing. Also, when a consumer is in a decision process we should assume that he or she tries to get information about the costs of each alternative. However, it is known that the marketing effort of service providers differ and that some are more conspicuous than others. If noise is to be added to these parameters, it should be distinguished so that consumers with more information channels also have more accurate values.

4.3.4 Consumer Memory

To be able to make more rational decisions, we can give the consumers a memory. This memory must hold information we can assume that consumers remember. First, it should hold information about all service providers that the consumer has information about. Each service provider can have an internal representation which includes the "reputation" this provider has for the consumer. For instance, a service provider recently abandoned should be given a quarantine time before it is considered as an alternative once again. Further, the memory can include information about former prices and decisions. It should also hold information about the social connections of the consumer, which are the other consumers this consumer communicates with.

4.4 Service Provider Behavior

As with the consumers, the service providers will also act as autonomous agents. This means that their behavior needs to be defined. The service providers make decisions to ensure that the performance measurements are as good as possible. Their strategies to ensure this are amongst others to vary the prices of the services and innovating with new services.

4.4.1 Company Performance Measurement

The service providers main task is to maximize the *profit* (for the owners), but it is also important to remember the other performance measurements used by investors and financial analysts. In the real world, the performance of service providers in the mobile industry is usually measured by *ARPU*, *customer churn rate* and *the number of subscribers.*

4.4.1.1 ARPU

To gain a market share is actually quite easy. For instance, if you want to sell a luxury car you just need to be willing to sell it with a huge discount and its sold in a minute. In the same way it's not very hard for a service provider to sign up subscribers if they allow access to their services for a price much lower than the competitors. However, these measurements (number of cars sold, number of subscribers, etc.) do not say anything about *how much money* you make of your business.

ARPU is a useful measurement of how well companies in the telecommunications industry are accessing their subscribers' revenue potential. The standard way of calculating the ARPU is by dividing the aggregate amount of revenue by the total number of subscribers who provided that revenue. More than often, the measurement is used as the *main performance variable* when evaluating companies in this industry.

We define the following parameters:

X_t = the number of subscribers at time t
N_t = the number of new subscribers between time t and t+1
L_t = the number of lost subscribers between time t and t+1
BF_t = the binding fee at time t
SC_t = the subscription cost at time t

Based on these parameters, the number of subscribers at time t+1 equals:

$$X_{t+1} = X_t + N_t - L_t \tag{4.8}$$

We can then calculate the ARPU as:

$$ARPU_t = \frac{(X_t * SC_t) + (N_t * BF_t)}{X_t + N_t} \tag{4.9}$$

4.4.1.2 Customer Churn Rate

Another useful measurement is the customer churn rate, or *attrition rate*, of the service provider. This is a measurement of how many individuals that are moving into or out of a collection over a period of time. When using discrete time steps, the churn rate (CR) is calculated as:

$$CR_t = \frac{X_{t-1} - X_t}{X_{t-1}} \tag{4.10}$$

where X_t is the number of subscribers, or customers, that the service provider has at time t.

4.4.1.3 Number of Subscribers

One of the oldest measurements of the success of service providers is the number of subscribers connected to them. Even though this measurement is not the best, as argued in the section on ARPU, it still says something about the strength of the provider. With few subscribers, the provider is more vulnerable to changes since each subscriber is responsible for a larger part of the revenue.

4.4.2 Service Provider Strategies

From a user perspective, what creates the *value* of service providers are which services they provide that affect this user. We can assume that all companies provide the basic services (call, SMS, MMS, etc.), and can based on this be compared as equal. The consumers choice of service provider is based on Equation 4.7. From this equation, we see that there are three ways of increasing the perceived benefit $K_{x\tau}$:

1. Decreasing the binding fee $p_{x\tau}$. This will probably lead to other providers decreasing their binding fees, giving decreased lock-in effects for all providers in the market.

2. Decreasing subscription fee k_{xt}. This will both affect perceived benefit directly, and through the *intrinsic value* calculated as a function of all subscription fees known to the consumer. This means that the effect of decreasing the subscription fee will be lower than the price it was decreased with.

3. Innovating, and thereby creating a higher network effect utility c_x, or higher intrinsic value through v_{xTotal}. This strategy is however more risky, since it is high costs and uncertainty related to innovation.

4.4.2.1 Binding Fee Strategies

Binding fee is a known notion for most consumers. When changing from one provider to another it is usual for the consumer to pay a “switching cost”. The reason for this has been the “paperwork” required to move the consumer from one provider to another, but there is also a second reason. When all service providers have binding fees, the lock-in effects will naturally reduce the customer churn in the market. Although the cost is not very high compared to the savings of changing provider, the consumers planning horizon may not be long enough to see the benefits of changing.

If a service provider sees that its consumer base is stagnating, or possibly decreasing, lowering or removing the binding fee for a period may be an effective strategy. The good thing about binding fees is that they are not a part of the “contract” with the consumer, meaning that they can be increased and decreased day-by-day if necessary.

4.4.2.2 Subscription Fee Strategies

The subscription fee is the monthly fee a subscriber is required to pay for a mobile subscription. As mentioned, this parameter affects the consumers valuation of a service provider in two ways: the consumer sees lower costs, and at the same time starts to value a subscription lower than before.

The thing about decreasing subscription fees as a strategy is that they may not easily be raised again. The subscription fee may be seen as a “contract” between service provider and consumer, and a change in the terms are not easily accepted by the subscribers if it is not to their advantage. However, if other providers have lower subscription fees and/or better services, lowering the subscription fee may be the only way to survive in the market.

4.4.2.3 Innovation Strategies

To this point, the innovations have been explained to give a positive effect to the valuation of a subscription, since they may either provide network effects or direct value. A theory of how knowledge of the already developed innovations spreads through the market has also been selected. However the development process of innovations, from idea to launch in the market, is not yet defined.

Many attempts have been made to model the development process of innovations. A well known approach is to adopt an evolutionary metaphor, assuming that innovation is a stochastic process of exogenous random events [CMO72, TA86].

However, as argued by Cheng and Van de Ven [Cm96], this kind of explanations means that we must assume that the source of innovation is exogenous to the system being examined. Another, more satisfying, explanation is that the process is not random but *chaotic*. This means that the process is a stable and deterministic non-linear system that produces behavior that *appears* irregular. The variables in the model will then be a function, at least partly, of the earlier values of the variables.

Further, it is often observed that the process of learning in innovation development seems random or unpredictable in early periods, but not in later. This can be explained by the high level of exploration (search, discovery, experimentation) instead of exploitation (choice, refinement, implementation and execution) in the early periods, while this balance changes through learning.

We can say that the total resources available for innovating at any time must be divided between exploration and exploitation. The more resources available, the faster the exploration will decrease and the exploitation will increase. When the sum of exploitation has reached a certain level, the innovation is ready for launch in the market.

Before launch, the innovation must be given either a network effect factor or a valuation factor. Since the development process will be fictitious, these factors can be assigned either randomly or as a function of some parameters in the development process.

4.4.3 Service Provider Memory

The memory of the service provider can maintain information helping the provider achieve its goals. First, a history of economic measurements should be kept: revenue, number of subscribers, and so on. These may again be used to calculate the key measurements. Further, information about current and previous strategies should be maintained.

4.5 Market Characteristics

As in other markets, there are basically two ways to expand the subscriber base: (1) "steal" subscribers from competitors, and (2) expand to include more of the *potential market* of unbound consumers (see Figure 4.8).

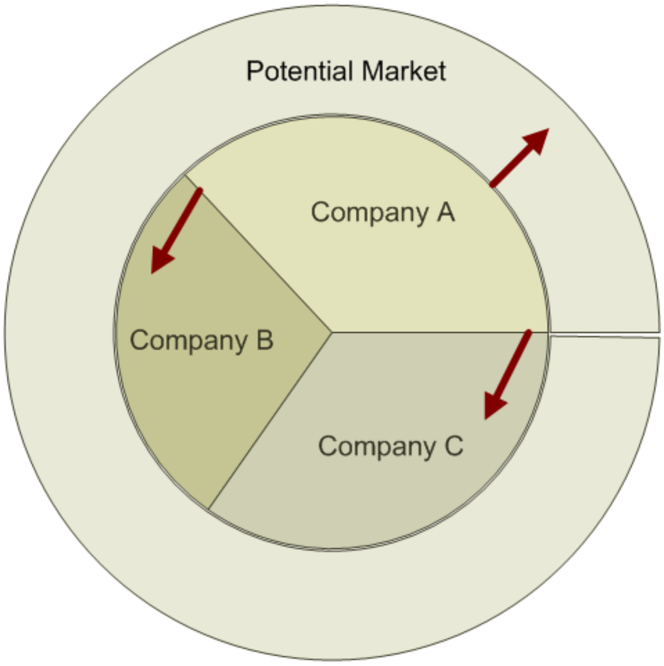

Figure 4.8: Expansion of subscription base

However, as mentioned in Chapter 1, the mobile market in Norway is near (if not past) its saturation point. Statistics from "Post- og teletilsynet" shows that in the third quarter of 2006 there were 106 subscriptions per 100 inhabitants of Norway[ot06], while the same number was 80 in 2001[8]. This means that the remaining potential market is very small, and that marketing toward this part of the population is not a remunerative investment. The implication of this is that the main competition is not about *new* subscribers, but about subscribers already having a relationship to one of the other providers/companies in the market. It thereby becomes a two-frontier war over subscribers: on one hand you must maintain your current subscriber base, on the other hand you must try to access other companies subscribers.

4.5.1 Social Networks

When an innovation is launched in any market, we cannot assume that it immediately catches the attention of all consumers. In his book[Rog03], Rogers gives

[8] While 106 subscriptions per 100 inhabitants may seem strange, the reason is that 10 percent of the population have *more than one* subscription. The real spreading of mobile subscriptions is 93 percent of the population.

some characteristics of each adopter category. According to these characteristics, the consumers differ in (amongst others) *social participation*, *social network size*, *marketing influence* and *opinion leadership* (a complete list of the characteristics is given in Appendix D). Earlier adopters are more likely to discover innovations at early stages since they more actively seek information through social networks, have more contact with "change agents", and have greater exposure to mass media communication than do later adopters.

Such differences are also pointed out in Malcolm Gladwell's "*The Tipping Point*" [Gla01]. Gladwell describes how some people are natural *connectors* for others, having much larger social networks than the average and thereby connecting people who would else not know each other. These persons also have the tendency to have larger influence on other people than the average, and are thereby often an important factor in diffusion of information. Figure 4.9 shows a simple social network for 20 consumers. As seen, each node (representing a consumer) has a number of other nodes connected to it. The lighter colored nodes have less connections (Laggards), while the darker colored nodes have more (Innovators).

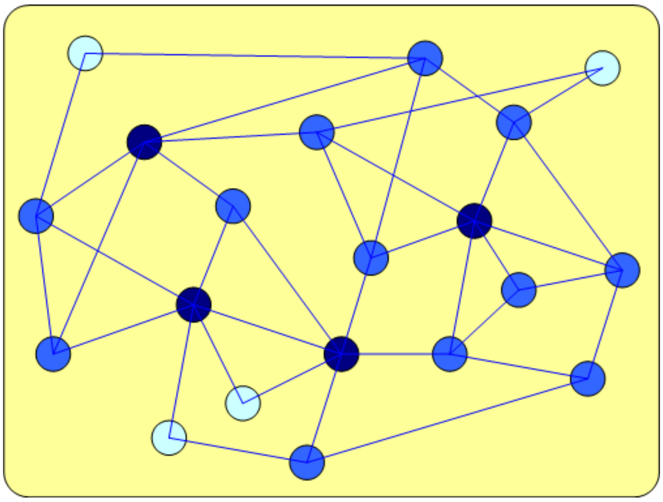

Figure 4.9: Social Networks

Some of the most recent (and recognized) work on the field of social networks is performed by Newman, Watts and Strogatz in their article *"Random graph models of social networks"*[NWS02]. In this article they describe three distinctive features of a real social networks structure:

Small world effect: people are connected by a very short chain of intermediate acquaintances

Clustering: in many real-world networks, the probability of a tie between two actors is much greater if the two actors in question have other mutual acquaintances.

Skewed degree distribution: the distribution of actors "degrees" [9] is highly skewed.

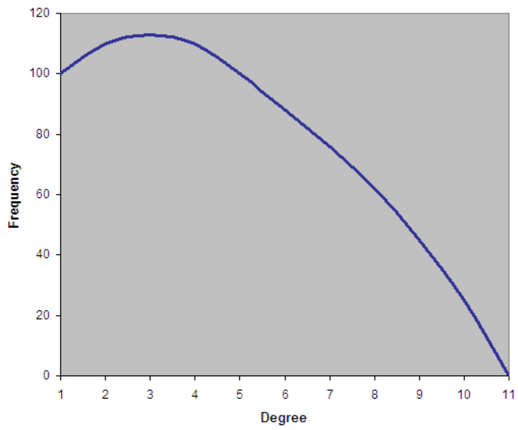

Figure 4.10: A skewed degree distribution

Figure 4.10 shows an example of a skewed degree distribution. A problem with the traditional *random graph*[ER59] approach to modeling of social networks, is that it creates a Poisson degree distribution instead of the skewed degree distribution. This makes it a poor approximation to real-world networks[NWS02], and an alternative approach should therefore be used to model the social networks that better reflects the three features mentioned in this section.

Opinion leadership is the extent to which one actor influence other agents it communicates with. According to Rogers[Rog03], earlier adopters have greater influence on their connections/friends than later adopters. As March [Mar78] explains: "*..human beings know that some people are better at rational argument than others*". Further, he notes that "*.. [human beings] exploit cleverness by asking others to construct reasons for actions they wish to take*". To encompass this, we can give each consumer an "influence factor" that reflects the influence he or she has on other. When the total influence on a consumer from its connections becomes large enough, the consumer may be affected by this influence.

4.6 Summary

The agents can be summarized by their properties in accordance to the guideline proposed early in this chapter. As described, an agent is defined by a set of **rules**, a **working memory** and a **rule interpreter**. Table 4.1 and 4.2 summarize these properties for the consumer- and service provider agents.

[9] The "degrees" reffer to the number of ties an actor has to other actors (i.e. the number of friends)

Table 4.1: Consumer Properties Summary

Part	Description
Rules	**Condition 1:** A known service provider provides a higher benefit than the current **Condition 2:** The known service provider is not recently used by this consumer **Condition 3:** The investment in the current service provider is payed back **Action:** If 1&2&3, change service provider
Memory	- Known service providers - Tried service providers - Known innovations - Social connections
Rule Interpreter	Performs according to Equation 4.7

Table 4.2: Service Provider Properties Summary

Part	Description
Rules	**Condition 1:** Another price strategy is thought to optimize performance measurements, and the service provider has the economic strength to allow for this strategy. **Action 1:** Change the prices **Condition 2:** The company is innovative and has recently developed a new innovation that can be launched in the market. **Action 2:**Launch the innovation.
Memory	- Number of subscribers - Current and past strategies - Revenue history
Rule Interpreter	The performance should be measured according to Equation 4.9 and 4.10, as well as other measurements where necessary.

Chapter 5

Requirements Specification

> "*There's no point in being exact about something if you don't even know what you're talking about.*"
> - John von Neumann

This chapter presents the Software Requirements Specification (SRS) for the simulation software. The SRS is loosely based on the IEEE standard 830-1998[1] to ensure that important aspects of a SRS are attended to. Since this chapter is a part of the report, as opposed to a "stand alone document", some parts of the standard is already covered by other chapters and are thereby omitted.

5.1 Overall Description

The simulation software to be developed, from this point called the *simulation engine*, is meant to be a self contained module usable in several settings. It will be developed as a Java ARchive (JAR) with an Extensible Markup Language (XML) interface for message exchange. This means that the engine may both be used as a simple library to wrap into a user interface, as well as it may be used in a more extensive architecture.

5.1.1 Product Perspective

The simulation engine is intended to be used as a component in larger software frameworks. The module itself does not have a user-friendly interface and will not provide data storage. Figure 5.1 serves as an example of a simulation framework the simulation engine may be applied to. As seen, several instances of the simulation engine may be used to serve possible users through a web interface.

[1] http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=15571 (Visited 2007/01/30)

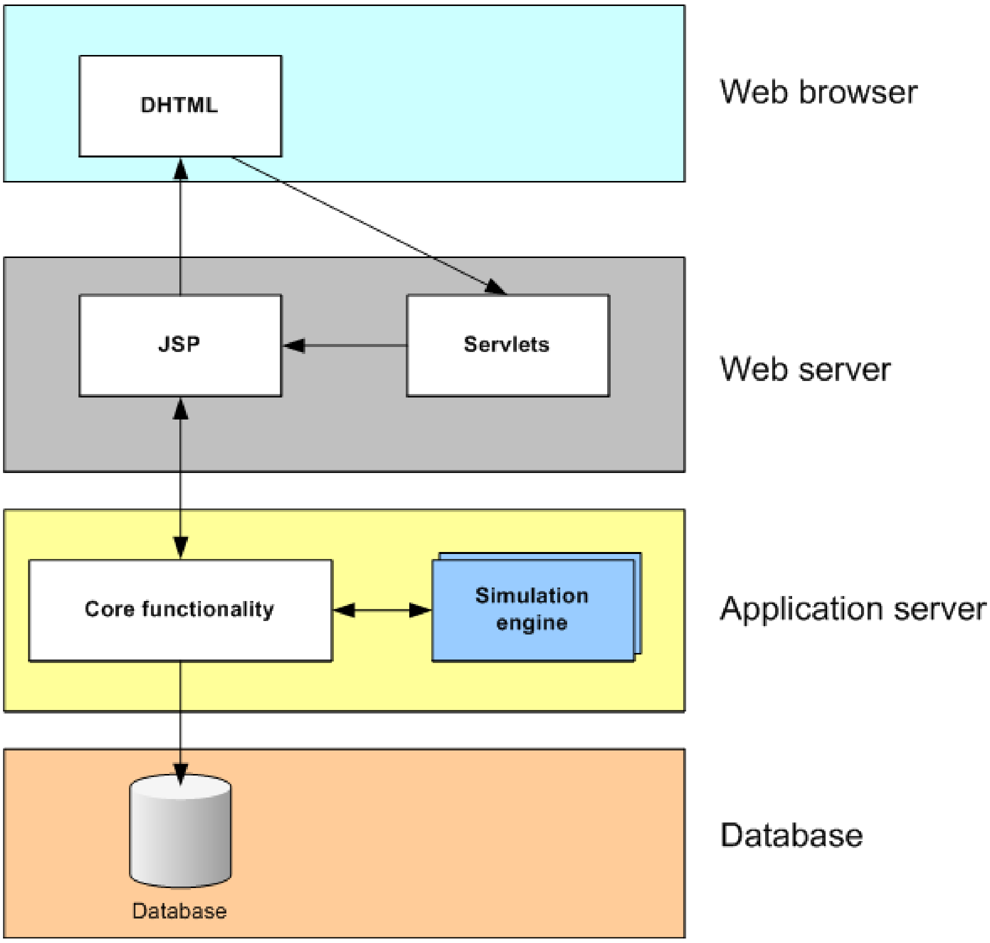

Figure 5.1: Blueprint of possible surrounding framework

The focus of this thesis will be on the design and implementation of the simulation engine as a *self contained* and *reusable* module. However, as an experiment, three bachelor students at NITH will implement the surrounding framework as their bachelor thesis. This will serve as a proof of the reusability of the simulation engine.

5.1.2 Product Functions

The main task of the software is to simulate competition between service providers in the mobile market based on a set of parameters provided by the user. The simulation should be transparent to the user, so that he or she can follow the process step by step. Finally, the results of the simulation must be returned to the user.

As shown in Figure 5.2, the main functionality as seen from outside the simulator is quite basic. The simulator is given a set of parameters as input, and returns the results of the simulation. The simulation shall imitate the competition between service providers providing competing innovations. The consumers are to choose between the providers based on the properties of the innovation,

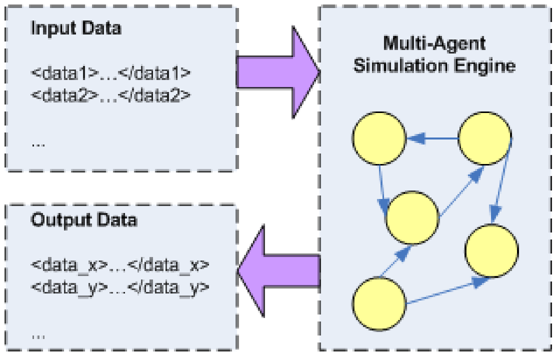

Figure 5.2: Functionality of the simulator

the service provider, the market, and externalities.

5.1.3 Operating Environment

The software will be implemented in Java 6.0, and is therefore independent of operating system and hardware platform. The only requirement is that the hardware platform and operating system are able to run Java and have the right version installed.

5.1.4 Design and Implementation Constraints

It is important that the engine is runnable both on ordinary computers as well as on more powerful servers. Simulations can be resource intensive, and the simulation engine should be scalable so that it can perform both smaller and larger simulations.

Being implemented as JARs, several simulation engines should not be run in the same Java Virtual Machine (JVM). By giving each simulation its own JVM, the simulations will not affect each other in any harmful way.

5.2 Functional Requirements

The functional requirements for the simulation engine are intended to guide the design of the architecture by describing the basic requirements of the simulator as well as the roles and attributes of the agents as completely as possible. The requirements are based on the theory and findings from previous chapters.

Since this is a research project and not a "contracted job", the requirements are neither exhaustive nor complete. They mainly serve as guidelines in design and implementation. Further, the implementation will be done in several iterations where the most important requirements are implemented first and new requirements are added if needed. It is therefore important to weight the requirements according to their importance, where the importance is amongst others seen according to how much the requirement influences the realism of the simulation.

The requirements are presented in tables and grouped by the functionality they are closest related to. They are tagged by IDs named FR-Yx, where Y is a letter separating the categories of requirements, and x is the number of the requirement (and FR is an abbreviation for Functional Requirement). This will improve the traceability of the requirements in prospective work. The requirements are also weighted by High (**H**), Medium (**M**) or Low (**L**) importance, to help prioritize if any requirements are in conflict or not reasonable[2] to solve. In addition, the tables present the anticipated Degree of Difficulty (abbreviated as DoD) to better understand the extent of implementing each requirement. This will be weighted the same way as importance, where high means "high difficulty" and low means "low difficulty".

5.2.1 General Requirements

The simulation engine is to be implemented as a self-contained piece of software. The requirements in Table 5.1 lists the general requirements for the software.

Table 5.1: Functional Requirements: General

ID	**Description**	**DoD**	**Priority**
FR-G1	The simulation shall be based on Agent-Based Computational Economics	H	H
FR-G2	The simulation engine must be accessible through exchange of XML messages	M	H
FR-G3	The simulation engine shall send temporary results to the client application	M	H
FR-G4	General configuration parameters common for all simulations should be configurable through a configuration file	M	H
FR-G5	The simulation engine shall log important event on several configurable levels	M	H
FR-G6	The simulation engine shall be implemented as a Java ARchive	L	H
FR-G7	The simulation shall use discrete time steps	L	M

[2] Some requirements may take to much time to solve, related to how important they are. If this is the case, we say that it is not reasonable to solve.

5.2.2 Consumer Properties

The consumer agents and their behavior are described in Chapter 4. These agents are the driving force of the simulation, since they make the decisions of where to place the money in the market. It is therefore extremely important that these are modeled as realistic as possible. The requirements for the consumer agents are given in Table 5.2

Table 5.2: Functional Requirements: Consumers

ID	Description	DoD	Priority
FR-C1	Consumer agents shall communicate through social networks	H	H
FR-C2	Consumer agents shall remember earlier decisions	M	H
FR-C3	The consumers should be affected by their social networks	M	H
FR-C4	Consumer agents shall look for the best possible alternative, even when connected to a service provider	L	H
FR-C5	The consumers shall base their choice of service provider on the perceived benefit according to Equation 4.7	L	H
FR-C6	Information should be affected by noise from the market	M	M

5.2.3 Service Provider Properties

As with the consumer agents, the service provider agents are also described in Chapter 4. The service providers are the producing part of the economic system, developing innovations for the consumers. The requirements are presented in Table 5.3.

Table 5.3: Functional Requirements: Service Providers

ID	Description	DoD	Priority
FR-P1	The service providers shall be able to develop several innovations	H	H
FR-P1	Innovations shall be allowed to arrive at subsequent times	H	H
FR-P2	Service providers can use pricing strategies to attract consumers	H	H
FR-P3	Service providers shall optimize based on performance measurements	H	H
FR-P4	Service providers can have an installed consumer base at the start of the simulation	L	H

5.3 Quality Requirements

This section presents the quality requirements that must be fulfilled by the simulation engine. The requirements are illustrated by Quality Attribute Scenarios[3], as suggested by [BCK03]. The scenarios describe events where the quality attributes are important. This approach is used to make the requirements more explicit, and it will also be possible to test the requirements directly by comparison to the expected response of the scenarios. In addition, the scenarios describe the functionality of the system in a structured matter. The scenario tables contain the following parts:

ID The identification tag of the requirement

Source The source of the stimulus

Stimulus A condition that needs to be considered when it arrives at a system

Artifact The artifact that is stimulated

Environment The condition of the system that the stimulus occurs within

Response The expected response from the system to the stimulus

Response measurement A measurement that makes it possible to test the requirement after and/or during implementation

The following sections will describe requirements for the quality attributes *modifiability*, *testability*, *performance*, and *availability*. Attributes such as *security* and *usability* are left out since they do not directly apply to the piece of software that is to be developed. The scenarios will not be exhaustive, but represent important aspects of the quality attributes to ensure that these attributes receive attention throughout development.

5.3.1 Availability

Availability is the quality attribute concerned with system failure and its associated consequences. In our case, the simulator may crash for several reasons. First, it is well known that simulations may take up a lot of memory. If the simulator exceeds the memory allocated to the Java Virtual Machine, it may crash. Further, it is possible that internal errors lead to failures. A *failure* occurs when the system is no longer consistent with its specifications, and is always observable either by another system or an end user. What is important for availability is to mask the *faults* of the system so that they do not become failures.

Table 5.4 specifies the scenario “An unanticipated message is received by the document parsing process”.

[3]The scenarios will be presented as tables rather than figures

Table 5.4: Quality Requirements: Availability 1

Part	Description
ID	QR-A1
Source	External to the system
Stimulus	Unanticipated message received
Artifact	Document parsing process
Environment	Normal operation
Response	Try to mask the fault by changing the received value. Notify the sender about the incident.
Response measurement	No downtime

Table 5.5 specifies the scenario "The memory of the Java Virtual Machine is overloaded".

Table 5.5: Quality Requirements: Availability 2

Part	Description
ID	QR-A2
Source	Internal to the system
Stimulus	The Java Virtual Machine is overloaded
Artifact	Memory
Environment	Normal operation
Response	Notify the user of the simulator about the fault, and reset the simulator.
Response measurement	The simulator should be back in a stable state within one minute

5.3.2 Modifiability

The modifiability of the architecture is concerned with the cost of changes (in time or money). Since the simulation engine is meant to be reusable for several possible settings, it should be easy to change some of the basic functionality to tailor the engine for each use.

Table 5.6 specifies the scenario " A developer wishes to modify the consumer behavior in the simulations at design time."

Table 5.7 specifies the scenario "A developer wishes to change the required input to the system at design time".

Table 5.6: Quality Requirements: Modifiability 1

Part	**Description**
ID	QR-M1
Source	Developer
Stimulus	Wishes to modify the consumer behavior
Artifact	Code
Environment	At design time
Response	Modifications are made with no side effects in other parts of the simulator
Response measurement	Modification made within two hours

Table 5.7: Quality Requirements: Modifiability 2

Part	**Description**
ID	QR-M2
Source	Developer
Stimulus	Wishes to change the required input to the system
Artifact	Code
Environment	At design time
Response	The required modifications are located, changes are made and the new build is tested
Response measurement	The new build is ready within five hours

5.3.3 Performance

Performance is about *timing*, and how the system responds to events. An event may be an interruption, a message, a request from users, or simply the passage of time. Basically we can say that performance is measured according to how long time it takes until the system responds to an event.

In the simulation engine, events may be generated externally (by the client application), or internally (by internal processes and threads). Performance is important in both cases to assure that the simulation both responds to invocation and runs fast and smoothly. As in other agent-based software, the performance is affected by the number of agents.

Table 5.8 specifies the scenario “A client application is trying to abort the current simulation”.

Table 5.9 specifies the scenario “The consumer agents perform their actions when a new time step is initiated”.

Table 5.8: Quality Requirements: Performance 1

Part	**Description**
ID	QR-P1
Source	Client application
Stimulus	Initiates an abortion of the current simulation
Artifact	System
Environment	During simulation
Response	The current simulation is aborted and a summary is returned
Response measurement	Simulation aborted and response returned within one second

Table 5.9: Quality Requirements: Performance 2

Part	**Description**
ID	QR-P2
Source	Consumer agents
Stimulus	Next time step is initiated
Artifact	System
Environment	During simulation
Response	All agents performs their action
Response measurement	The average latency for each agent is below 0.1 ms

5.3.4 Testability

The testability of a piece of software refers to the ease with which it demonstrates its faults through testing. Even though the simulation software is to be developed by one person, the need for testability is still there. The software will be implemented iteratively, and each iteration should be properly tested before the next one starts.

Table 5.10 specifies the scenario "The developer has completed an iteration and needs to test the build[4] ".

[4] A *build* here refers to the piece of software at the exit state of an iteration

Table 5.10: Quality Requirements: Testability 1

Part	**Description**
ID	QR-T1
Source	Developer
Stimulus	Iteration completed
Artifact	Build
Environment	At development time
Response	The software provides computed values or possible error messages
Response measurement	All faults should be masked, and failures shall not occur

Chapter 6

Architecture

"Always design a thing by considering it in its next larger context a chair in a room, a room in a house, a house in an environment, an environment in a city plan."
- Eliel Saarinen

This chapter describes the architecture of the simulation software and the rationale for it. First, the architectural drivers and tactics to achieve these are presented. Then, the architecture is described in accordance to the *4+1 view model*, as proposed by Philippe Kruchten[Kru95], but with one additional view: the *agent view*. Finally, the rationale of the architecture is given.

6.1 Architectural Drivers

The architecture of a software system is shaped by some of the most important functional, quality and business requirements [BCK03]. These shaping requirements are known as the *architectural drivers*. The architectural design should reflect the drivers to ensure that the software complies with the requirements at the end of the design phase.

The architectural drivers for this architecture are **modifiability** and **availability**. Modifiability is highly important to ensure that the incremental development can be done without major changes for each round. Availability is important to ensure that the final product behaves as expected in all situations.

6.2 Quality Tactics

“Quality tactics” is the name used to describe certain fundamental design decisions, which can be used to achieve quality requirements. Every quality tactic has some impact on one or more of the quality attributes. The tactics described in this section are based on [BCK03], and have been selected to achieve the quality requirements from Chapter 5.3 with focus on the architectural drivers: **modifiability** and **availability**.

6.2.1 Modifiability Tactics

The tactics to improve modifiability have as their goal to controll the time and cost to implement, test and deploy changes. We can separate between three classes of modifiability tactics:

1. Reducing the number of modules affected by a change (“Localize modifications”)
2. Limit modifications to the localized modules (“Prevent ripple effects”)
3. Controll the deployment time and cost (“Defer binding time”)

6.2.1.1 Localize Modifications

Restricting modifications to a limited set of modules will generally reduce the cost of changes[BCK03]. The following tactic is chosen as the main tactic to achieve this goal:

Maintain semantic coherence “Semantic coherence” is a measurement of the relationship among responsibilities in a module. The main goal of this tactic is to assure that each module is as self contained as possible, without excessive reliance on other modules. The goal is achieved by defining responsibilities for each module so that they are semantically coherent. One possible subtactic is **abstracting common services** to specialized modules.

6.2.1.2 Prevent Ripple Effects

A “ripple effect” is the necessity to make changes to modules not directly affected by the modification. The following tactics are chosen to prevent such effects:

Restrict communication paths As far as it’s possible, the modules with which a given module share data should be restricted. This is achieved by limiting the number of modules *consuming* data *produced* by the given module and vice versa.

Hide information By making as much information as possible private to the modules, changes are to a larger degree isolated within one module and thereby prevented from propagating to other modules.

6.2.1.3 Defer Binding Time

This class of tactics is closely related to late binding of parameters. In our case, the parameters of the simulator should naturally be provided through the interface for each simulation to make the simulator as dynamic as possible.

6.2.2 Availability tactics

As mentioned in Chapter 5.3, a *failure* occurs when the system no longer behaves as specified. A *fault* can become a failure if it is not properly detected and masked. Availability tactics are meant to prevent faults from becoming failures.

As with modifiability tactics, we can put the tactics to improve availability into three classes:

1. Fault *detection*
2. Fault *recovery*
3. Fault *prevention*

6.2.2.1 Fault Detection

To detect faults in the simulator, *exceptions* will be used. An exception is raised when a fault is recognized, and each exception should be handled with care so that the failure is avoided.

6.2.2.2 Fault Recovery

Fault recovery is not directly applicable to the simulator. Recovery consists of preparing for recovery and making the system repair. Since there is no consistent information stored in the simulator, such tactics seems superfluous.

6.2.2.3 Fault Prevention

To prevent faults in the first place, transactions should be used. This means that several sequential steps should be bundled such that all steps must be completed, or none should be completed. This ensures that the simulator is in a stable state at all times.

6.3 Architectural Description

As mentioned introductorily, the "4+1 view model"[Kru95] is chosen to model the architecture. This model allows us to describe the architecture through multiple, concurrent views. The four main views of the "4+1 view model" are:

- Logical view: The object model of the design
- Development view: Describes the static organization of the software in its development environment
- Process view: Captures concurrency and synchronization aspects of the design
- Physical view: Describes the mapping(s) of software onto hardware, and reflects its distributed aspects

The "+1" stands for a set of use cases, or *scenarios*, which tie the views together.

In addition, an *agent view* is added to better present the agent architecture with its properties. The intent is to tie the functionality of the agents to the other views of the architecture. To present this in an easily understandable way, a simple modeling language has been created. The notation of the language is presented along with the blueprint.

6.3.1 Logical View

The logical view of the architecture visualizes how the functional requirements are to be met; what the system must provide in terms of services to its users. This is modeled in the form of *object classes*, or *packages*, and how these cooperate. This will not only serve as a tool for functional analysis, but also to identify common mechanisms and design elements across the system.

Figure 6.1 shows the main functionality to be included in the simulator. The following are descriptions of the object classes/packages of the blueprint, and their functionality:

External Interfaces Gateway This is the "entrance point" to the simulator. All functionality that is to be visible from outside the simulator must be accessed through this interface (for instance *start*, *stop*, etc.). The interface shall receive setup documents in XML format to ensure flexibility. The module is also responsible for returning results to the external application.

Simulator Control Mechanisms The main functionality of this container is to manage the simulation process. This includes setting up the simulation, starting and stopping the simulator, and resetting the necessary parameters when needed.

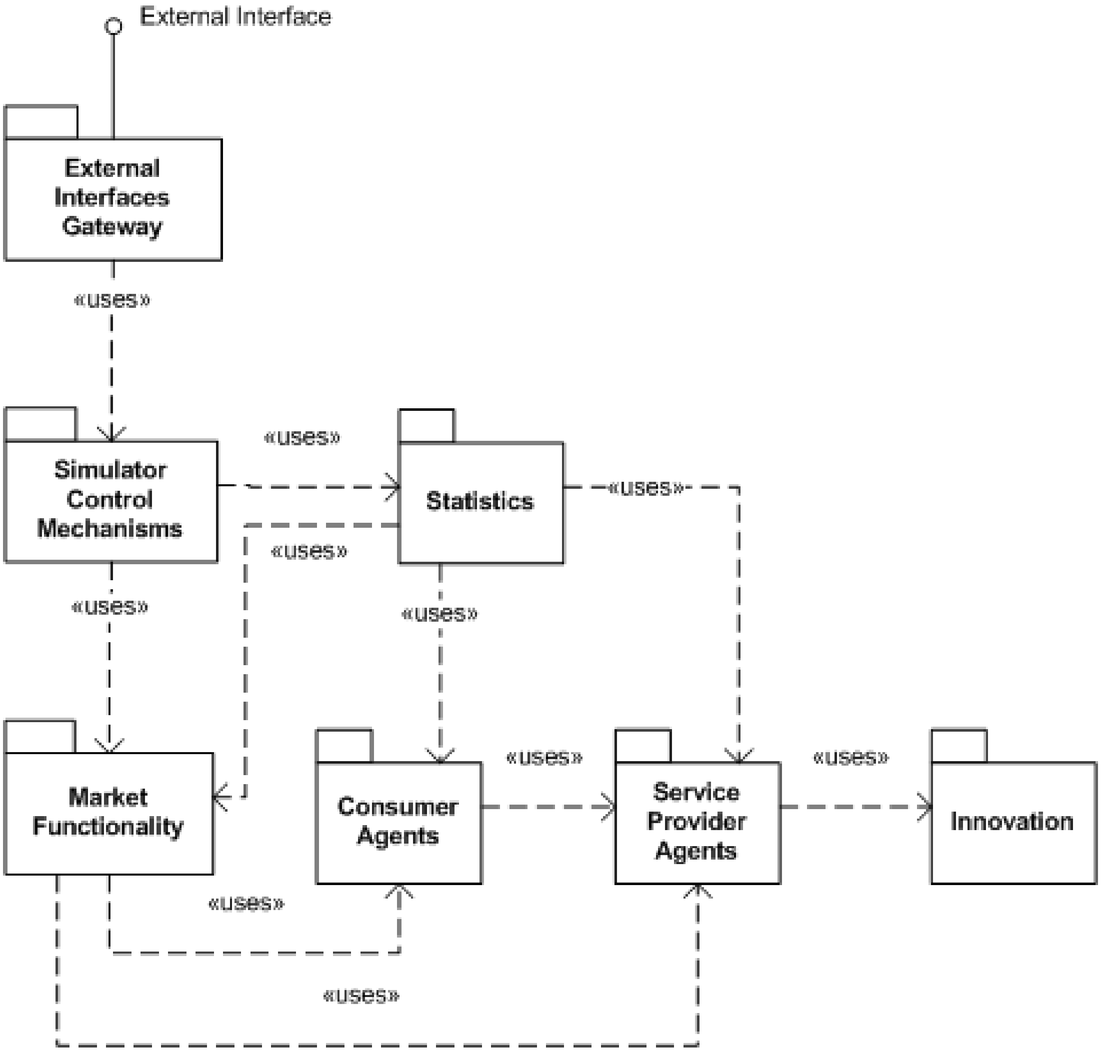

Figure 6.1: Logical blueprint

Statistics The statistics package is responsible of collecting and refining information about the activities in the simulator. Low-level information must be analyzed and made available as high-level abstractions. While the agents themselves are responsible of knowledge about their own activities, the statistics package should collect this information to provide reliable and understandable information to the user of the simulator.

Market Functionality All agents are deployed in the *market* of the simulator. The market is an abstraction of all externalities affecting the agents, such as *time*, *noise*, *communication channels* and so on.

Consumer Agents The consumer agents are described in several chapters of this report. They are the agents responsible of taking the main decisions in the system: which service provider to subscribe to.

Service Provider Agents The service provider agents are responsible of providing services to the consumers. They should act according to performance measurements to improve their own position in the market.

Innovation Innovations are the economic goods that are traded in the economy.

6.3.2 Development View

The development view of the architecture focuses on the *actual* module organization of the software. The software is separated into subsystems which are organized hierarchically in layers, where each layer provides a well defined interface to the layers above it.

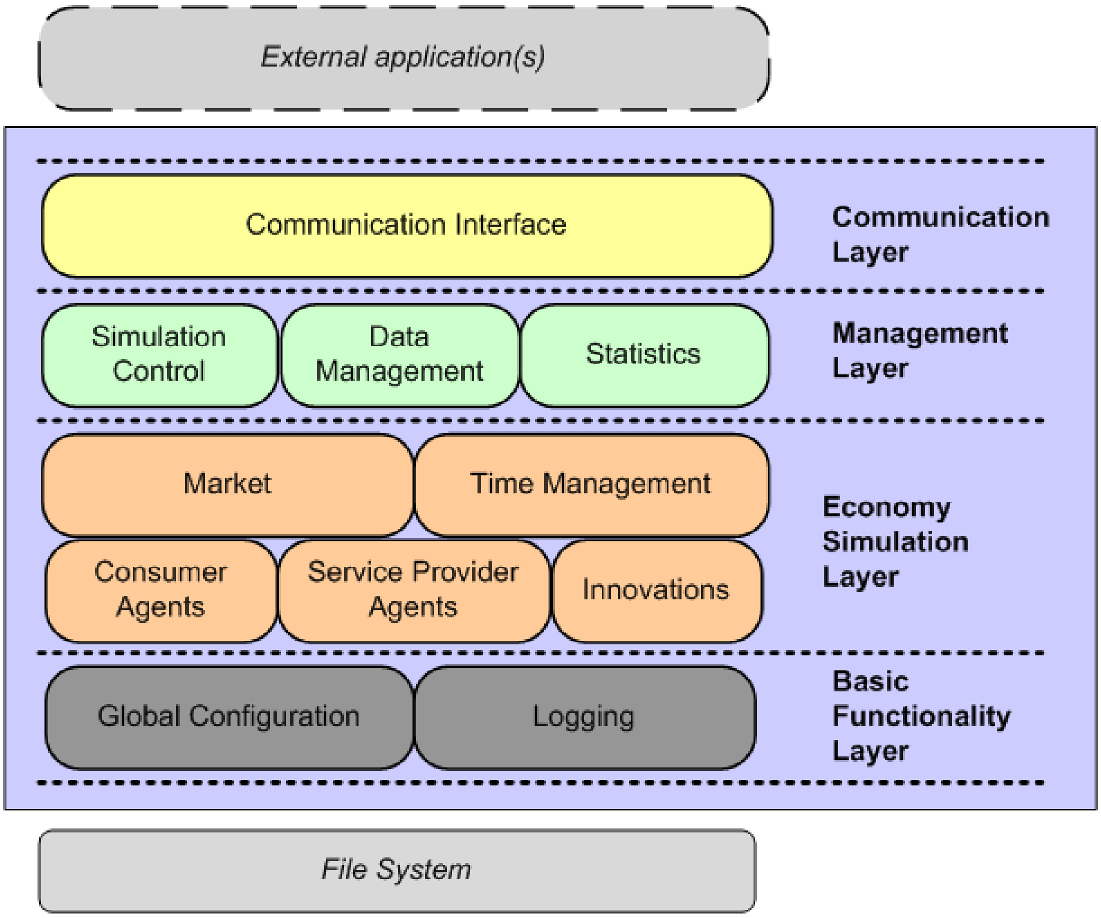

Figure 6.2: Development blueprint

Figure 6.2 shows the development blueprint of the simulator. The design rule of the figure is that a subsystem in a certain layer can only depend on a subsystem in the same layer or in layers below. This means that the functionality of the *Global Configuration* and *Logging* subsystems are accessible to all modules above them (hence the name *"Basic Functionality Layer"*), but do not have access to any other subsystems themselves. The following are descriptions of the subsystems:

External Application(s): These are the clients of the simulations. The external applications may be everything from simple user interfaces to complex web applications. The communication is initiated by accessing the communication interface, and all messages sent from and to the simulator shall be as XML documents.

Communication Layer:

Communication Interface: This is the interface the applications must communicate through. Each simulation must create a new instance of the communication interface, and all communication goes through this interface. This makes it easier to change the communication protocols as well. For instance, the simulator may easily be reorganized to a Web Service by changing this layer.

Management Layer:

Simulation Control: This subsystem is responsible of setting up, starting and stopping the simulations correctly. It is also responsible for sending feedback to the client application at the right times.

Data Management: This subsystem contains the parser(s) of the XML documents sent back and forth to the external applications. At the time of simulation initialization, the data management submodule is responsible of parsing the setup document and using the information correctly.

Statistics: The statistics subsystem is responsible of gathering information from the economy simulation layer and presenting this information in the form of useful key measurements. The information created in this subsystem is sent back to the external applications.

Economy Simulation Layer:

Market: The market is the baseline of the economy. Through this submodule consumers get information of service providers, and service providers can push information to consumers. All consumers and service providers are registered in this submodule.

Time Management: This submodule drives the simulation forward. The simulation shall use discrete time steps, and the speed at which these time steps are performed shall be dynamically configurable.

Consumer Agents: The consumer agents are thoroughly described throughout this report. The main responsibility of this submodule is defining the consumer agents behavior.

Service Provider Agents: As the consumer agents, the service provider agents are described already. The main responsibility of this submodule is defining the behavior of the service provider agents.

Innovations: This submodule shall handle the properties of innovations in the economy.

Basic Functionality Layer:

Global Configuration: This submodule is responsible for the configurations which are common for *all* simulations. These configurations should be specified in a configuration file on the file system. In this way, the parameters can be read in globally for all simulations.

Logging: As with the global configuration submodule, logging is also common for all simulations. To avoid a large amount of log files, all simulations should write to the same file through *one* shared instance of this module.

6.3.3 Process View

The process view of the architecture is presented by UML sequence diagrams. This view takes into account non-functional requirements such as performance and availability. A *process* is a grouping of tasks that form an executable unit[Kru95]. The main focus of this blueprint is to show the process flow in the simulator.

Figure 6.3 shows the process blueprint for the processes running when the simulator is started. The simulation is initiated by some external process. The controller process is then responsible for starting the market, which controlls the time thread. For each time step the market requests the agents to perform their actions. When they are finished, the controller process is asked to return results. The controller process then gathers the necessary information through statistics and other information sources, and returns this information to the external process.

Figure 6.4 shows the process blueprint for the processes performed by the consumer agents for each new time step in the simulator. The market initiates the processes by telling each consumer in the market that they must perform their actions. The selection of which consumers perform their action at which time should be stochastic, since similar patterns for each time step may provoke unrealistic results.

Figure 6.5 shows the process blueprint for the processes performed by the service provider agents for each new time step in the simulator. As with the consumer agents, the processes are initiated by the market. Since there is no communication with other agents here, we can assume that the timing of service providers is irrelevant.

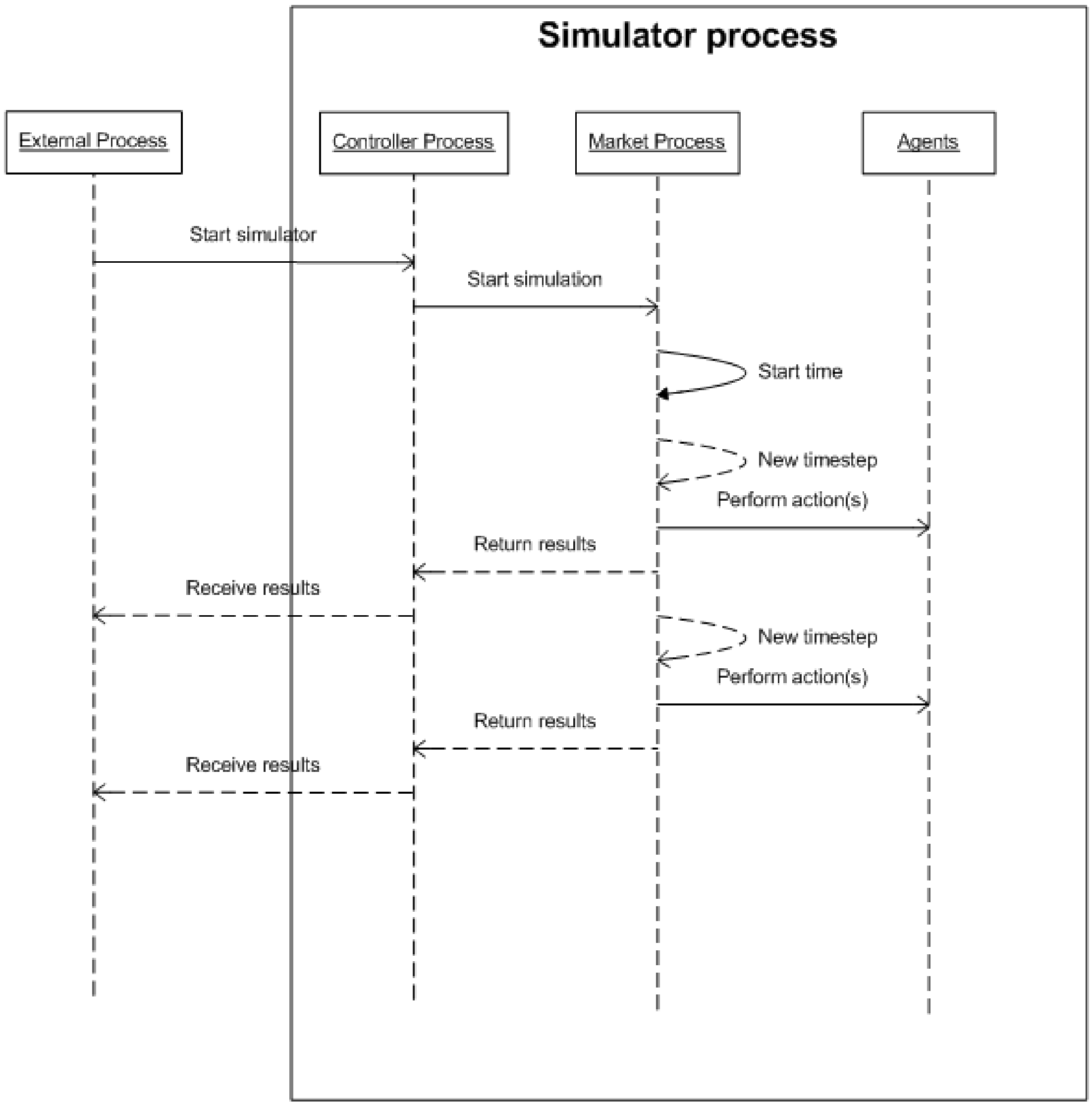

Figure 6.3: Process blueprint (starting the simulator)

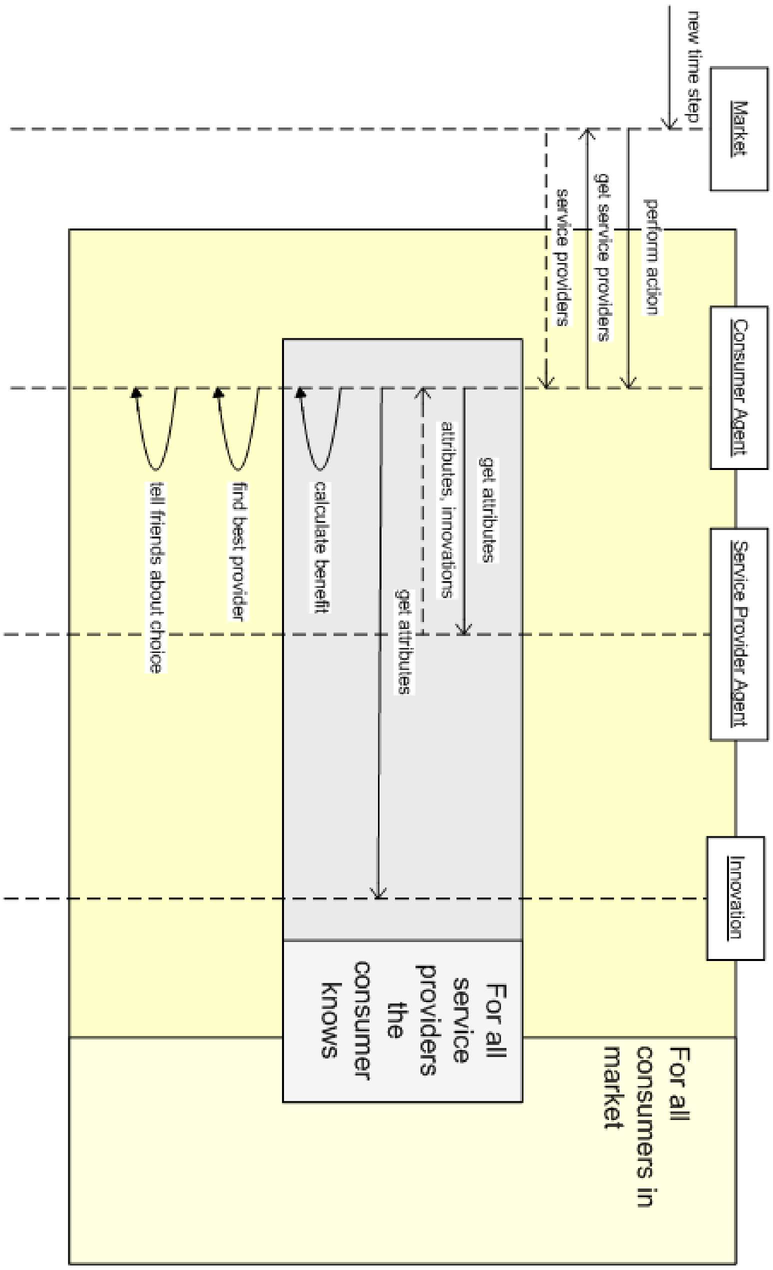

Figure 6.4: Process blueprint (consumer agent behavior)

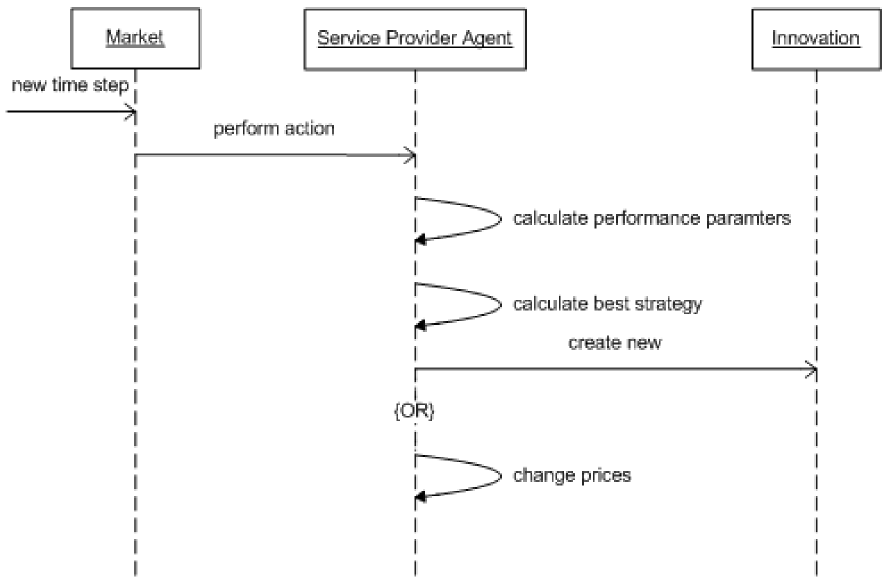

Figure 6.5: Process blueprint (service provider agent behavior)

6.3.4 Physical View

The physical view of the architecture is intended to map the software to the hardware. Since the software is designed for one computer, this view is excluded from the architectural description.

6.3.5 Agent View

In addition to the traditional architectural blueprints, the agent view is included in this architectural description. The intent is to show the basic relationships between the agents, their properties, actions, and externalities in a way easily understandable to a broad audience. It will thereby be a supplement to both the *logical view* and the *development view*.

Figure 6.6 shows the agent architecture blueprint. The notation used is created for this specific purpose, and does not follow any other design guidelines. What was important when designing this modeling language was that the relations between *rules*, *memory* and *rule interpreter* could be represented in an easily understandable way.

The associations show the connections between externalities, agents and their properties. The [X..Y] notation is intended to be read as "From X to Y". For instance, the `SUBSCRIBES TO` association between consumer and service provider should be read "A consumer SUBSCRIBES TO *from 0 to 1* service provider(s)". The association between service provider and marketing channel should be read "A service provider USES anything *from 0 to N* marketing channels", where N is intended to mean the maximum possible number.

The model contains *one* externality (as seen from the agents), namely the *market*. The market is included to show how noise is added to the information in the agent system. Both the market itself and marketing creates noise, which affects both the consumers decisions and the service providers strategies.

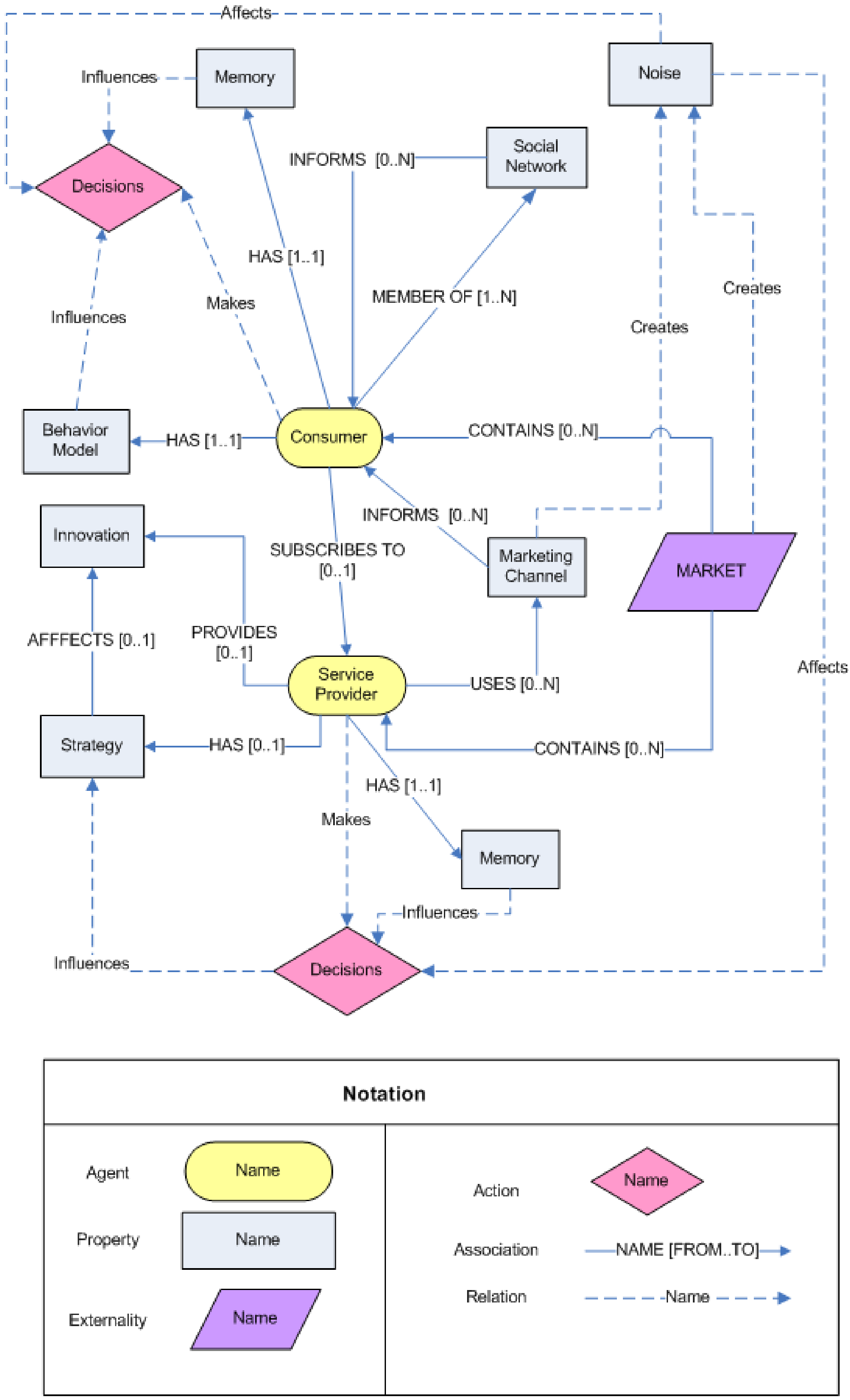

Figure 6.6: Agent architecture blueprint

6.3.6 Scenarios

The scenarios are meant to bind the views together, and are in some sense redundant. They serve as validation of the architecture, and as a driver for further architectural refinement. The scenarios are described in an informal way to ease the readability.

6.3.6.1 Scenario: Initializing the simulator

Precondition: None

Postcondition: The simulator is set up correctly, or the external application is warned about what went wrong

Action sequence:

1. The *external application* creates a setup document following the standard
2. The simulation engine receives the setup document through the *communication interface*
3. The *communication interface* passes the document to the *simulation control* module, which assures that the *data management* module parses the document correctly
4. The *global configuration* module reads the configurations that are not passed in the setup document
5. If the setup of the simulator was performed without errors, a confirmation is returned to the external application. Else, an error message is returned

6.3.6.2 Scenario: Starting the simulator

Precondition: The simulator is set up correctly

Postcondition: The simulation is started

Action sequence:

1. The *external application* starts the simulation through the *communication interface*
2. The *communication interface* tells the *simulation control* module to start the simulation
3. The *simulation control* starts the *market*, which starts the time thread

6.3.6.3 Scenario: An unexpected event occurs

Precondition: The simulation is running

Postcondition: The unexpected event is logged and, if causing an exception or error, the external application is warned

Action sequence:

1. An unexpected event occurs in the *Economy Simulation Layer*
2. The event (Error or Exception) is caught and logged through the *Logging* module
3. If the event causes erroneous results in the simulation, the simulation is stopped and an error message is returned to the *Simulation Control* module

   a) The *simulation control* module creates an XML response through the *Data Management* module

   b) The response is returned to the external application through the communication interface

6.3.6.4 Scenario: Stopping the simulator

Precondition: The simulation is running, but is wanted to stop

Postcondition: The simulation is stopped

Action sequence:

1. The *external application* stops the simulation through the *communication interface*
2. The *communication interface* gives the responsibility to the *simulation control* module
3. The *simulation control* module tells the *market* to stop the time thread, and tells the *data management* module to create a final results document
4. The *data management* module uses the *statistics* module to gather the required information and returns an XML response document
5. The *simulation control* module sends the final document to the external application through the communication interface

6.4 Rationale

In this chapter the architecture of the simulator has been described using the 4+1 view model with one additional view. I believe that the architecture clarifies the findings made in this report, and that it solves important challenges discussed throughout this report.

The quality tactics for *modifiability* were followed in the design by making the architecture modular and layered. I also added two configuration interfaces; one low level (global configuration) and one high level (XML through the communication interface). The remaining tactics were followed during implementation. The quality tactics for *availability* are not directly reflected by the architecture since they are more implementation specific.

Most functional requirements of Chapter 5.2 are also integrated in the design. The ones not directly reflected are: **FR-C4, FR-C5** and **FR-P4**. These were therefore given attention during implementation.

Chapter 7

Implementation

> "*Creativity requires the courage to let go of certainties.*"
> - Erich Fromm

As mentioned in Chapter 1, the software has been implemented iteratively. The following sections shortly describe what was implemented in the iterations, and how each iteration affected the behavior of the simulator.

7.1 Iteration 1: Interface and Skeleton

The first iteration was used to set up an interface towards external applications, and to implement a skeleton for the simulator. The architecture of Chapter 6 was implemented with empty functionality, and the communication paths were tested. Next, simple XML functionality for parsing and creation of the communication documents was implemented (see Appendix J for further descriptions of communication interface).

The last step in this iteration was to implement the *Basic Functionality Layer*, since this layer would be used by the succeeding iterations. The *Global Configuration* module was created as a cached XML reader for fast access to variables. The *Logging* module was created as a light-weight log with five levels: debug, info, warn, error and fatal.

7.2 Iteration 2: Simple Consumer Behavior

In the second iteration, the basic functionality of the Economy Simulation Layer was implemented. First, simple market functionality such as *adding agents* and *time management* was implemented.

A simple decision model based on Equation 4.4 was implemented in the consumer agents, and the service provider agents were set as "dummies" with fixed prices and *one* innovation (with network effects) each throughout the simulation. All information was global and perfect, and consumer agents had unlimited resources.

The behavior of this iteration was quite predictable: the "winner-takes-all" equilibrium appeared already at the first time step. This was logical, since it was what the model was intended to do. All consumers behaved as rational actors, immediately selecting the best alternative.

7.3 Iteration 3: Refinement of Consumers - Adopter Categories

While iteration 2 behaved as expected, this was not what the software was intended to simulate. As explained in Chapter 4, we cannot expect all agents to behave equally. It was therefore natural to distribute the consumers in adopter categories. The characteristics of the categories were implemented in the *behavior model* of each consumer (see Figure 6.6), and the categories were distributed using a stochastic method[1].

The implemented adopter characteristics were: *noise*, *influence on other consumers* and *size of social network*. The noise was made configurable[2], as was the level of influence each category has. The *size* of the social network is also made configurable by defining the maximum and minimum number of friends allowed (innovators will then have close to the maximum, while laggards will have close to the minimum).

However, since the social networks themselves were not implemented in this iteration, the only thing tested was the noise factor. What was found in this iteration was: *the noise creates a foundation for smaller providers to exist*. In difference from the second iteration, some consumers were now sticking to the less beneficial provider for a longer time. When the number of time steps simulated was large, all consumers ultimately chose the best alternative, but for a long while the noise was a "stickiness-factor".

Simple statistical measurements were also implemented in this iteration. The number of transitions between providers, customer churn, ARPU, and total revenue were collected by the *Statistics* module for each time step. What was recognized was a tremendous customer churn throughout the simulation, which seemed unnatural. Consumers seemed to change too frequently, at a rate of above two changes each year.

[1] A stochastic draw was made from the interval [0,100]. If the number was below 2,5 the consumer became an *Innovator*, if between 2,5 and 16 the consumer became an *Early Adopter* and so on.

[2] Noise to the value can be set to different levels, whereas noise to number of subscribers can be turned on or of.

7.4 Iteration 4: Refinement of Consumers - Payback Time and Memory

It is not natural for consumers to change as rapidly as they did in iteration three (see for instance [ot06]). The main problem was that the consumers had unlimited economic resources. To put some limits on this behavior, *Payback Time* and *Memory* were introduced.

The *payback time* was set as a limit for the economic strength of agents. The basic requirement is that *consumers are not allowed to make new investments until the last investment is payed back*. *Payed back* here means that the total perceived benefits must be higher than the total costs.

The *memory* was implemented to keep track of which providers the consumer had tried out. It is natural to assume that a consumer will put a provider in "quarantine" for a certain time after he or she decides to change from this provider to another one. The time consumers shall keep the providers in such a quarantine is configurable through the low-level configuration (Global Configuration).

When testing this new model, the customer churn had decreased to more stable levels, and consumers changed their subscriptions less than twice per year in average.

7.5 Iteration 5: Social Networks

By now, most of the foundation for adding functionality related to social networks was implemented. In this iteration, the code needed for generation of such networks was added. However, as described in Chapter 4.5, it is not sufficient to randomize the process. The networks should be created in accordance to the three features presented in the mentioned chapter (small world effect, clustering, and skewed degree distribution).

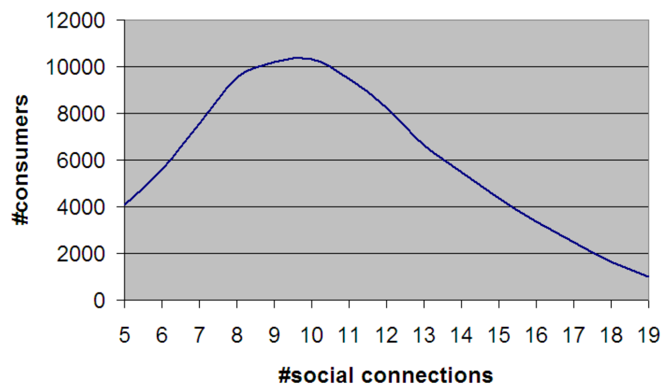

Figure 7.1: Skewed degree distribution in the implementation

Figure 7.1 shows an actual degree distribution of a social network created by the simulator for 100.000 consumers with between 5 and 19 connections (configurable). The horizontal axis represents the number of connections (the degree), while the horizontal represents the number of consumers having the specified degree. In addition, the later adopters are located to the left (fewer connections) while the earlier adopters are located to the right. It is thereby confirmed that the social network generator of the simulator is in accordance with the *"skewed degree distribution"* requirement.

The second feature that was tested was the "small world effect"; that all consumers can reach all others in a limited number of steps. A well known article in this field is "The small world problem" by Milgram[Mil67]. Through an experiment in the United States, he found that the average path length of acquaintances between two randomly chosen persons fell around 5.5. The article has inspired much of the well known work in the field, amongst other the "Six Degrees of Separation" theory[3] and the more recent, highly popular, book "The tipping point" by Malcolm Gladwell[Gla01].

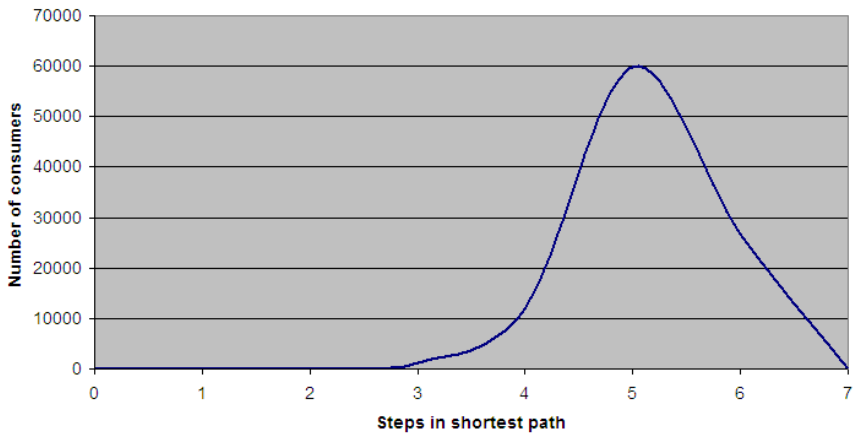

Figure 7.2: Shortest path between randomly chosen consumer and all other consumers

For testing purposes, a modified Dijkstras algorithm was used to find the shortest paths between a randomly chosen consumer and all other consumers in the network. The networks consisted of 100.000 consumers, each connected to between 5 and 20 other consumers. Figure 7.2 shows the result from these tests. From 10 runs of 100.000 consumers, the average path length became 5.12, which is fairly close to the results by Milgram (5.5). The results from the testing are given in Table G.1 in Appendix G.

As mentioned in Chapter 4, the social networks should also be *clustered*. A cluster is a collection of nodes where all nodes have connections with all other nodes within the collection. In our case, these are the "close friend groups", where all the consumers are good friends with all others.

To find such clusters was not a straight forward task in a node network consisting

[3] For more information about "Six Degrees of Separation", see `http://en.wikipedia.org/wiki/Six_Degrees_of_Separation` (Last visited 2007/05/03)

of hundreds of thousands of nodes. However, an effective solution was found for analyzing these networks. To provoke clustering, the network creation algorithm uses a parameter from the configuration file governing the probability that two consumers with a common friend are friends themselves. This kind of "clustering coefficient" was also used by Watts and Strogatz in their 1998 article[WS98], where they showed for a variety of real-world networks that such a coefficient took values between 40 and 50%. This coefficient allows us to create different kinds of social networks, from those consisting of only "loose connections" to networks with few but large clusters. However, when setting this coefficient to between 40 and 50% (0.4–0.5), the simulation better reflects a real-world scenario of consumers.

The social networks of the model then seem to satisfy all three features found in real social networks (as described by [NWS02]). Testing of the model was also quite astonishing: When testing with *one* initial provider with a few consumers subscribing to it and a lot of consumers unbound (not subscribing to any provider), the typical S-shaped growth of innovation diffusion appeared. Since this was the only provider, the intrinsic value of a subscription was equal to the subscription rate so the binding fee had to be set as zero to make consumers consider the choice. However, when changing the *Network Effect Factor*, speed of diffusion changed as well. This is in accordance with the assumption of network effects as a diffusion trigger.

The algorithms used for the creation of social networks are given in Appendix H.

7.6 Iteration 6: Service Provider Behavior

Until this iteration, the service providers had been kept as dummies without any intelligence or memory. It was now time to make these agents act autonomously.

The first step was to implement the strategies. The cost-related strategies were implemented through quite simple algorithms . This gave the service providers different degrees of economic flexibility and allowed them to actually *compete* for the consumers. Next, the innovation strategy was implemented. This was done by creating an *Innovation Factory*. The innovation factory acts according to *chaos theory*, as proposed in [Cm96], where the first phase is chaotic and the last is more governed and sequential (exploration vs. exploitation). The main difference between a *chaotic* and a *stochastic* process, is that the chaotic process actually follows a specific pattern. This means that the patterns of the chaotic process is possible to analyze, and we can give real estimates of how many steps it takes to get from one value to the next. The companies are given different degrees of *innovativeness*, and a higher degree of innovativeness gives higher frequency of innovation launches. This is reflected both in the *chaotic*(exploration) and the *sequential*(exploitation) part of the innovation development. The algorithms used by the innovation factory are given in Appendix H.

The memory of the service providers was also implemented. This allows the service providers to remember key measurements for several time steps, and to act according to these measurements.

The entire architectural blueprint of Figure 6.6 was now implemented with one exception: *marketing*. This feature was not implemented, mainly for two reasons: 1) Uncertainty of how marketing affects the consumers, and 2) the need for simplicity in the model to maintain understanding of processes and results.

To “make up” for this exclusion of marketing, the consumer agents were allowed to search for new service providers. The *Innovators* were given the ability to search for new providers in average every N’th time they evaluate possible providers, where N is configurable through the Global Configuration.

This iteration completed the planned implementation of the *Economy Simulation Layer*. However, the results were difficult to understand, and the data needed to be analyzed to see trends.

7.7 Iteration 7: Graphical User Interface

As mentioned earlier in this report, students from NITH in Bergen have implemented a web interface using the simulation engine. Because of the short time period (approximately 20 weeks), the two projects were not synchronized at all times. It could take up to one week from changes were made in the simulator engine before these changes were reflected in the user interface. This led to the implementation of a simple user interface wrapped around the simulation engine. This user interface needed to be very dynamic to enable rapid changes as the simulation engine developed.

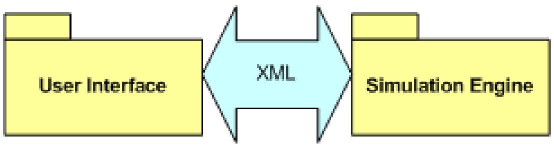

Figure 7.3: Graphical User Interface

Figure 7.3 shows how the user interface is connected to the simulation engine through the same interface as all other external application. In addition to serving as a “rapid-change” interface, it also served as a mean for testing the communication interface of the simulator. The interface is packed into the JAR file along with the simulation engine to make it directly runnable (see Appendix J).

One of the features of the interface package is the graph module. While implementing the interface, the need for a simple graph-line painting utility in Java was recognized. A light weighted utility that is used for all the line graphs in the interface was therefore developed. This made it easier to make changes and include new visualizations.

The user interface consists of five tabs: Configuration of service providers, configuration of the simulation environment, response from the simulator and two tabs for visualization of statistics.

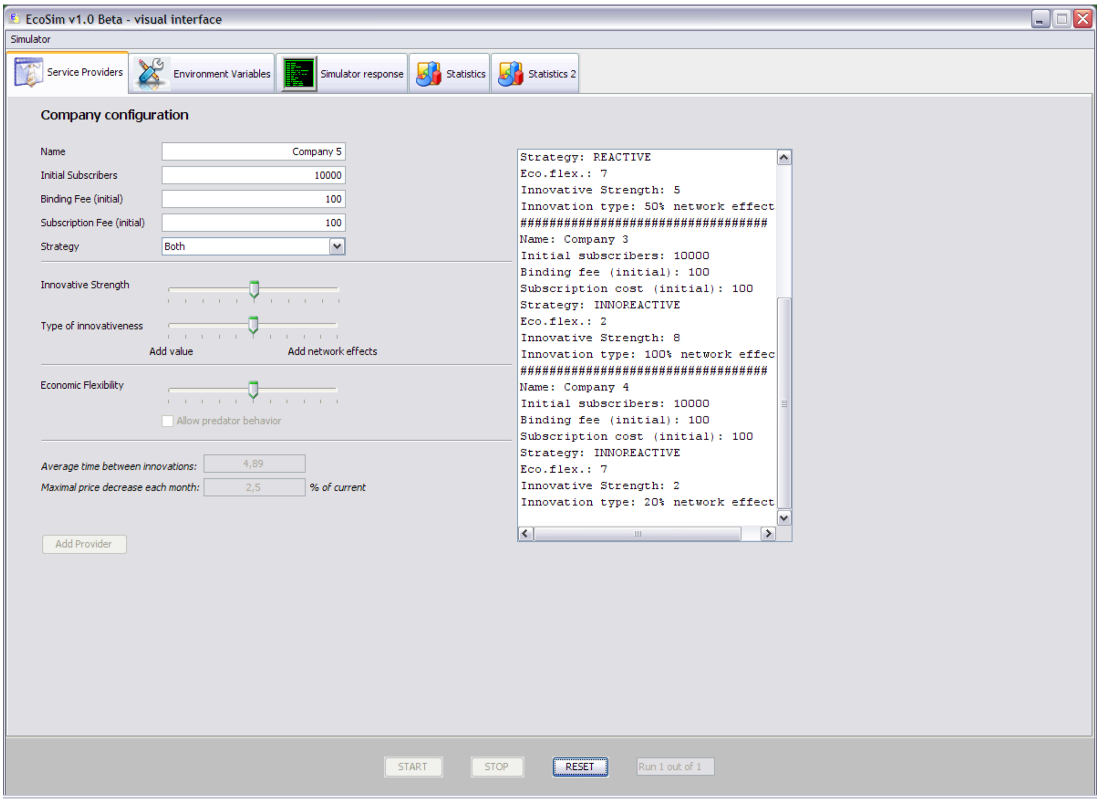

Figure 7.4: GUI: Configuration of service providers

Figure 7.4 shows the tab for configuration of service providers. Through the interface you can configure:

- The name of the provider
- The initial subscribers of the provider
- The initial binding fee of the provider
- The initial subscription cost of the provider
- The strategy for this provider

The supported strategies are *innovating*, *reacting*, *none* and *both*. By selecting one of these strategies, more options specific for the selected strategy appears in the user interface.

Figure 7.5 shows the tab for configuration of the environment variables. Through this interface you can configure:

- The initial (unbound) population
- The length of the simulation (number of months to be simulated)
- The number of runs (for several runs of the same scenario)
- The speed of the simulation

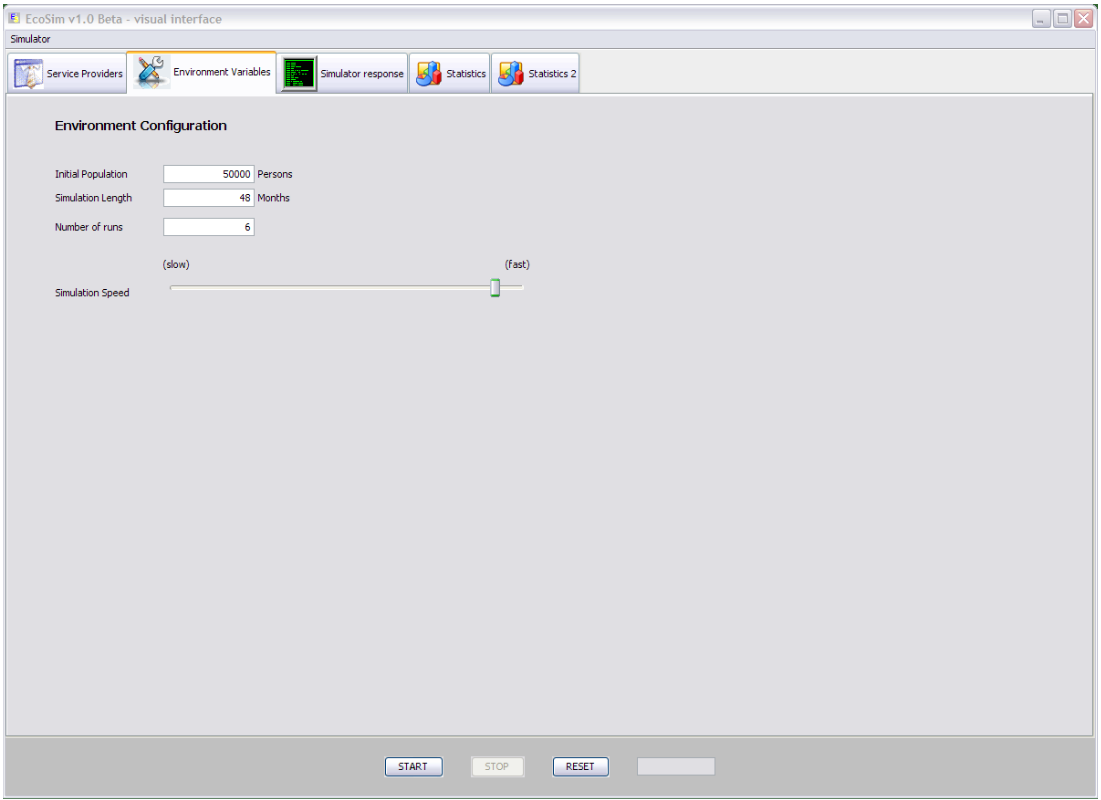

Figure 7.5: GUI: Configuration of simulation environment

The user interface module will automatically create images of the statistics tabs for each simulation. This way, several simulations for a specific scenario can be run subsequently without supervision, and the results can be inspected when all runs are finished. Another option would be to write the results to a database, which could enable more thorough statistical analysis. However, since this was a lightweight user interface, such features were not included[4].

Figure 7.6 shows the tab for simulator response. This was mainly created for debugging purposes, but is now used to display specific information of what has happened on all time steps. It is easy to change what information is to be displayed by editing the source code of the class parsing information received from the simulation engine (`edu.ntnu.ecosim.visual.XMLManager.java`).

Figure 7.7 shows the first statistics tab. In this tab, the following graphs are displayed:

- The number of subscribers for each provider
- The customer churn of each provider
- The ARPU for each provider
- The total revenue for each provider

[4] The web application developed by the students in Bergen will include a database connection, and hence the possibility for such features.

Figure 7.6: GUI: Response from simulation engine

This is also the main statistics tab, since it displays all the performance measurements of the providers as explained in Chapter 4.4.

The last statistics tab, shown in Figure 7.8, displays some additional information that may be of value for the user. The *average cost pr. consumer* is included for macro economic analysis. This kind of information can be used to see the general trends in the market of how much each consumer is paying for the mobile services. Further, a pie chart is included to give dynamic feedback of the market share of each provider. Finally, the number of innovations launched by each provider is displayed for informative purposes.

The user interface does not include all possible information, but was made to enable rapid testing and debugging of the simulation engine. More information from the simulations can be retrieved in the log, or by making simple changes in the source code of the interface.

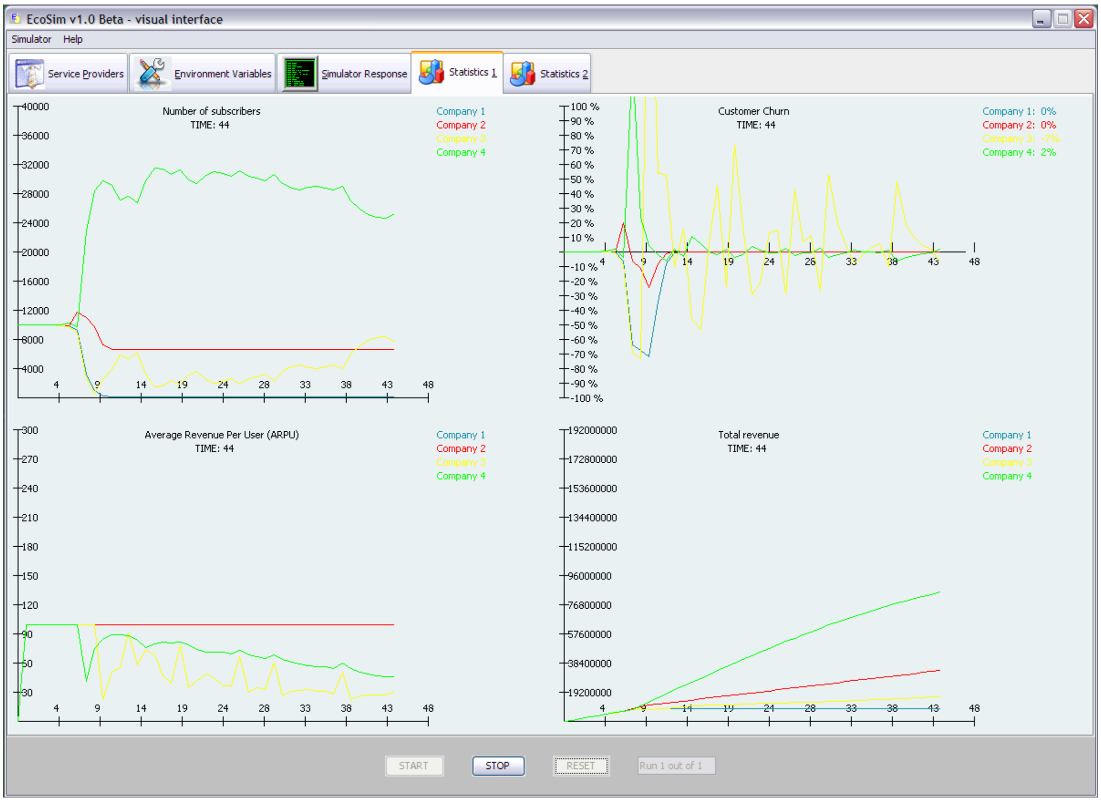

Figure 7.7: GUI: Statistics 1

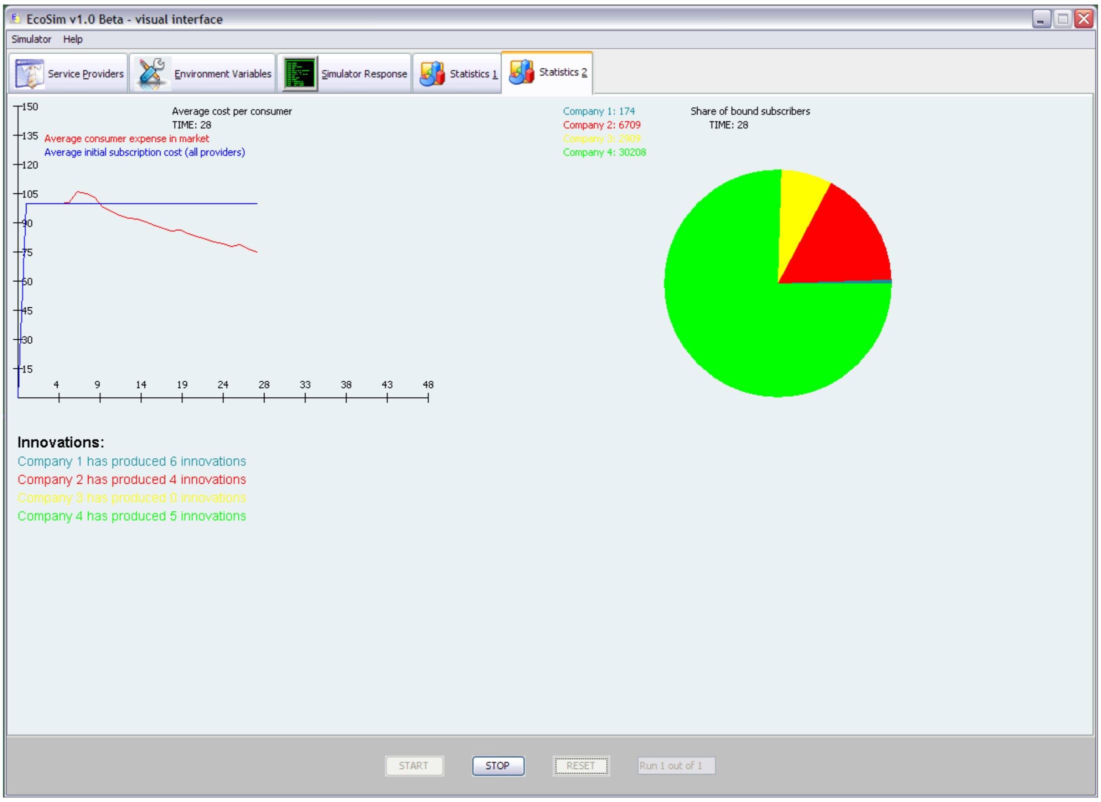

Figure 7.8: GUI: Statistics 2

7.8 Iteration 8: Memory Management

One of the final concerns in the development was the massive memory usage of the simulator. The JVM used all of the available memory, and sometimes ended in an `OutOfMemory error`. It was therefore high priority to 1) mask such faults before they become failures, and 2) make the code more memory efficient.

7.8.1 Masking the Faults

There were two main sources of massive memory consumption: creation of social networks and activation of consumer agents. Both these sources included work on huge `ArrayLists`, containing as many objects as there are consumers in the simulation (often up to 100.000). When the memory consumption of these operations became too massive, the JVM threw an `OutOfMemory error`. The solution to this was to catch this error, canceling the ongoing operations and send an error message through the communication interface.

7.8.2 Redesigning the Code

When it became obvious that memory consumption was a problem, much of the code was already implemented without considering such problems. The code therefore needed a "walk through" to see where memory could be saved.

First of all, unnecessary `ArrayLists` were removed. For the social networks, each consumers list of friends were implemented as an alternative list class (called `FriendList`), which can be trimmed to save memory. Further, explicit garbage collection was used in the code where it was obvious that this should be done (for instance where temporary lists were no longer used). Finally, unnecessary code was removed to make each object as memory efficient as possible. The most important sources of improvement were the classes related to the consumers, since these take up most of the memory.

7.9 Iteration 9: Differentiating Innovation Types

The final step of the implementation was to differentiate the kinds of innovations in the simulations. Until now, all innovations were producing network effects. While this was the main focus of the thesis, it was interesting to see how markets would react if some innovations were producing value on their own. Examples of such services are *games*, *calendars*, *calculators*, etc. These services give value to the user independent of the number of other users.

Including both kinds of service innovations allows for even more simulation scenarios. For instance, one can simulate a market of companies focusing on different kinds of innovation development.

7.10 Summary

The ninth iteration concluded the implementation, and the simulator now includes all the features mentioned in Chapter 2.5:

1. Companies produce innovations with **subsequent arrival**
2. There are **more than two competing innovations** in the simulated market
3. The simulator uses **advanced modeling of social networks**, which is proved to reflect real-world networks
4. The simulator makes **explicit use of prices**

In addition to the simulation engine, a simple user interface is implemented and tested. Stable and reusable functionality for configuration management and logging are also implemented as part of the `basic functionality` layer.

Appendix F contains detailed class diagrams for the implementation. The diagrams are divided into sections according to which layer (in the development view of the architecture) they belong to. In addition, the attached CD contains the source code of the implementation and the Javadoc API documentation. The contents of the CD are described in Appendix I.

Chapter 8

Testing and Results

"Don't be afraid of opposition.
Remember, a kite rises against; not with; the wind."
- Hamilton Mabie

This chapter describes the results from testing of the simulator. First, the verification and validation of the simulator are described. Then the results from testing the hypothesis of Chapter 2 are given. Further, some results from other testing of the simulator are discussed. A summary of possible sources of errors is presented to ease future evaluation of the simulator. Finally, the chapter concludes with a discussion of the results, and the results are related to recent work done in the field.

8.1 Verification and Validation

As described in Chapter 2, verification refers to determining that the simulation software performs as intended, whereas validation is concerned with whether or not the conceptual simulation model is an accurate representation of the system simulated. Before the results from simulation could be generalized, both these aspects had to be covered.

8.1.1 Code Verification

The first step of the *verification* process was verifying the code. This meant that the code had to be analyzed for obvious errors. Each iteration ended with a walk-through of the code consisting of adding documentation and analyzing the internal algorithms. This ensured that the code was both *necessary* and *correct*.

In addition, intermediate results from each iteration were analyzed to see if they reflected general assumptions for the implemented functionality. For instance, when implementing the algorithm for social network creation, separate network analysis code was added to ensure these networks reflected real-world networks. The social networks were compared with recognized theory on real-world networks, and were shown to be in accordance to these theories. In the final version, analysis of the constructed networks can be turned on or off through the configuration file.

After the last iteration, a program called *FindBugs*[1] was run on the code. This program searches for bugs in Java code to find correctness bugs, bad practice, dodgy code, and other possible errors. After correcting most of the bugs found, the only ones left were some comments on the explicit garbage collection and the methods for getting singleton instances. The explicit garbage collection was kept since this had been shown to be efficient for memory management. The singleton methods were also kept, as they were safe enough for usage in the simulator.

The results of simulation were animated to ease the verification. According to Kleijnen [Kle95], this is a good way find to both programming errors and conceptual errors. Both these kinds of errors were found and corrected during testing just by evaluating the graphical output.

8.1.2 Simulator Validation

In this thesis, no real-world data has been obtained to compare the results with. Instead, validation is performed through comparison to theory and testing of hypothesis. The following sections of this chapter present the results from the testing of the hypothesis, and also a comparison to recognized theory.

[1] `http://findbugs.sourceforge.net/` (Visited 2007/05/08)

8.2 Testing of Hypothesis

An important part of the testing was the validation (or falsification) of the hypothesis. The following sections present the results from testing of the hypothesis. The parameters of the service providers are configurable through the user interface, whereas the ones for the consumers are configured through the configuration file. Table 8.1 and 8.2 give short descriptions of the parameters, and they are further explained in Appendix J. The *initial (unbound) population* is the number of consumers that do not have any subscription when the simulation starts.

Table 8.1: Service provider parameters

Parameter	Legal values	Description
Name	Any textual string	The name of the service provider
Initial subscribers	Integer [0..N]	The initial installed base of this service provider
Binding fee (initial)	Integer [0..N]	The initial binding fee of the service provider (may change according to strategy)
Subscription fee (initial)	Integer [0..N]	The initial subscription fee for subscribing to this provider (may change according to strategy)
Strategy	NONE, REACTIVE, INNOVATIVE, or INNOREACTIVE	The selected strategy of the service provider (see Chapter 4.4.2). The *REACTIVE* strategy operates on the binding fees and subscription fees, while *INNOREACTIVE* is a combination of *REACTIVE* and *INNOVATIVE.*
Innovative strength	Integer [0..10]	The innovative strength of the service provider governs how often the company is able to launch new innovations
Innovation type	0–100%	The percentage of innovations that have network effects, as opposed to the ones having intrinsic value.
Economic flexibility	Integer [0..10]	The economic flexibility of the service provider governs how much the prices can be reduced to attract new subscribers
Predator behavior	ON or OFF	If the company has predator behavior, it will not settle with any other result than a 80% market share. If it does not gain the required share, it will continue to lower its prices.

Table 8.2: Consumer parameters

Parameter	Legal values	Description
MaxFriends	Integer [1..N]	The maximum number of friends a consumer can have
MinFriends	Integer [1..MaxFriends]	The minimum number of friends a consumer can have
ClusteringFactor	Real number [0..1]	The probability (percentage/100) that two consumers with a common friend are friends themselves. The parameter influences the social network creation algorithm (see Appendix H).
InnovatorInfluence	Integer [0..100]	The influence innovators have on their friends
EarlyAdopterInfluence	Integer [0..100]	The influence early adopters have on their friends
EarlyMajorityInfluence	Integer [0..100]	The influence early majority consumers have on their friends
LateMajorityInfluence	Integer [0..100]	The influence late majority consumers have on their friends
LaggardInfluence	Integer [0..100]	The influence laggards have on their friends
ValueNoiseFactor	Real number [0..1]	The noise related to valuation of basic subscriptions. The value will be higher for later adopters than earlier, and will at most (for the laggards) be between +400xValueNoiseFactor (%) and -800xValueNoiseFactor (%).
useSubscriberNoise	true or false	Whether or not noise should be added to the perceived number of other consumers using the service provider. (Only used when NetEffIsLocal is false)
PlanningHorizon	Integer [0..N]	The planning horizon of the consumers for future values and costs (months)
MemoryTime	Integer [0..N]	The “quarantine time” for providers that have been tried and later rejected
NetEffIsLocal	true or false	Whether the network effects should be calculated based on friends (true) or entire installed base of the provider (false)
ProviderSearchInterval	Integer [0..N]	How often innovators should search for new providers in the market (months between each search)
ProviderInfoGlobal	true or false	Whether information about service providers spread through social networks (false) or is globally known by all consumers (true)

8.2.1 Hypothesis 1

Hypothesis 1: Innovation diffusion will differ when the services are valued through social networks instead of globally.

This is the base-line hypothesis of this thesis. Much work was put into constructing social networks that better reflect real social networks. As explained by Abrahamson and Rosenkopf[AR97], these networks may have major effects on innovation diffusion. The results from testing of this hypothesis is therefore highly important for validation of the model.

8.2.1.1 Settings

Initial (unbound) population: 99980

Service providers: (1 company)

Table 8.3 shows the settings of the only service provider in this simulation.

Table 8.3: Hypothesis 1: Service provider settings

Parameter	**Value**	**Variations**
Name	Company 1	
Initial subscribers	20	
Binding fee (initial)	0	
Subscription fee (initial)	100	
Strategy	INNOVATIVE	
Innovative strength	10 (max)	
Innovation type	100% network effects	

Consumer parameters (from configuration file):

Table 8.4 shows the configuration of the consumer parameters from this simulation. The variables are set through the configuration file.

Variations:

In the first simulation, consumers valuate the subscriptions based on the number of close friends are using the services of the subscription. Changing **NetEffIsLocal** to **false** in the second simulation made the consumers valuate the subscriptions based on the entire installed base of the service provider.

8.2.1.2 Results

Both the simulated scenarios should produce the expected S-shaped curve, and so they did. The results were as expected: when the consumers valuated the services according to how many of their *friends* were using them, the diffusion

Table 8.4: Hypothesis 1: Consumer settings

Parameter	**Value**	**Variations**
MaxFriends	20	
MinFriends	5	
ClusteringFactor	0.45	
InnovatorInfluence	50	
EarlyAdopterInfluence	40	
EarlyMajorityInfluence	30	
LateMajorityInfluence	20	
LaggardInfluence	10	
ValueNoiseFactor	0.00	
useSubscriberNoise	false	
PlanningHorizon	12	
MemoryTime	6	
NetEffIsLocal	true	changed to false in second simulation
ProviderSearchInterval	3	
ProviderInfoGlobal	false	

peaked (reached 99% of the market) at 14 months. This result was stable for all five runs of this scenario.

On the other hand, when the total usage of the services in the market (the installed base) was considered instead, the diffusion peaked at 9 months - five months earlier. This result was also stable for all five runs.

These results verified the hypothesis, and the simulations confirm that research on social networks may indeed have strong implications on theories of innovation diffusion. Innovations may actually diffuse slower than assumed, since the consumers do not consider the technology until they have received enough information/influence from their closest friends. Most research in the field has been based on consideration of the entire installed base for valuation of innovations with network effects. This may lead to false conclusions, since network effects *depend* on existing networks and their structure.

8.2.2 Hypothesis 2

Hypothesis 2: The "winner-takes-all" equilibrium is not the only equilibrium for companies with competing innovations.

The last decade, research on network effects of innovations may have lead many practitioners to overemphasize the installed base, implementing "get-big-fast" strategies to trigger these effects. This thesis is intended to follow up the 2006 article by Lee, Lee, and Lee[LLL06]. They argue that competition between incompatible technologies may lead to *other* equilibria than "winner-takes-all", since the customers choose technologies based on the choices of his or her *acquaintances* rather than the size of the installed base. They found that whether or not the competition results in a winner-takes-all situation depends on the structural characteristics of the social networks. They also found that *highly clustered networks* are more likely to preserve the local bias of "loosing" technologies.

To follow up on the mentioned work, the simulated scenario contained 3 companies producing innovations launched sequentially. Such simulations were requested by Lee, Lee, and Lee in the conclusion of their article, as their simulation contained only two competing innovations launched simultaneously.

8.2.2.1 Settings

Initial (unbound) population: 90000

Service providers: (3 companies) Table 8.5 shows the settings for the service providers of this simulation. All three service providers had the same settings, only differing in their names.

Table 8.5: Hypothesis 2: Service provider settings

Parameter	**Value**	**Variations**
Name	Company 1, Company 2, and Company 3	
Initial subscribers	20	
Binding fee (initial)	0	
Subscription fee (initial)	100	
Strategy	INNOVATIVE	
Innovative strength	10 (max)	
Innovation type	100% network effects	

Consumer parameters (from configuration file):

Table 8.6 shows the configuration of the consumers for this simulation. The variations used in the sensitivity analysis (explained later) are also pointed out.

Table 8.6: Hypothesis 2: Consumer settings

Parameter	**Value**	**Variations**
MaxFriends	20	Changed in sensitivity analysis “connections”
MinFriends	5	Changed in sensitivity analysis “connections”
ClusteringFactor	0.45	Changed in sensitivity analysis “cluster size”
InnovatorInfluence	50	Changed in sensitivity analysis “influence”
EarlyAdopterInfluence	40	Changed in sensitivity analysis “influence”
EarlyMajorityInfluence	30	Changed in sensitivity analysis “influence”
LateMajorityInfluence	20	Changed in sensitivity analysis “influence”
LaggardInfluence	10	Changed in sensitivity analysis “influence”
ValueNoiseFactor	0.00	
useSubscriberNoise	false	
PlanningHorizon	12	
MemoryTime	6	
NetEffIsLocal	true	
ProviderSearchInterval	3	
ProviderInfoGlobal	false	

8.2.2.2 Results

The results from this simulation differed from the results that Lee, Lee, and Lee achieved in their simulations. The graphical output of the testing is shown in Appendix E. In all five test cases, the first provider to launch an innovation captured a critical base of the first consumers and thereby got a head start in the “race”. Because of network effects *and* diffusion through social networks, this head start made the leading service provider to the tipping point of diffusion before the other providers innovations had diffused enough to reach the same point. At that time it was already too late for the other providers to catch up.

In addition, the model contains consumer differentiation. Not all consumers have the same influence on other consumers, and in addition the consumers with the most friends and the largest influence (the innovators) are most likely to be the first to adopt the innovations. This gives the head start of the first launched innovation an even bigger impact on the outcome, making it harder for the succeeding innovations to achieve the critical consumer base. This leads to the following proposition:

Proposition 1: Because of differentiated characteristics of consumers, the “winner-takes-all” equilibrium is the most likely outcome when competing innovations are launched sequentially.

This proposition is however exposed to both idealization errors **and** errors in data. Boundaries for influence and the number of friends for each category are supplied through the configuration file, and the behavior of the consumers are idealized. Further, the work by Lee, Lee, and Lee mentions a *high degree* of clustering as the trigger for market sharing. In the basic simulation, the recommendations by Watts and Strogatz[WS98] for a clustering factor of 40-50% were followed. To see how the settings in the configuration file influenced the results, a sensitivity analysis was performed. The results from this analysis are given in Appendix E. The main findings were:

- A very high clustering factor (100%) gave a faster diffusion rate, and lead to more innovations diffusing in paralell. However, the tendency is still that one company captures the entire market.
- When the influence was set equal for all consumers, the “tipping point” of diffusion happened at a later time, and the non-winning companies seemed to have better conditions for creating an initial base and retaining it.
- When the allowed number of friends was changed from [5..20] to [5..40], the diffusion was accelerated since the earlier adopters had even more connections/friends.
- When all consumers had the same amount of friends (10), the diffusion was slowed down and several service provider gained a larger consumer base before one took the lead.

Also in the basic simulation, variations were found. When a competing providers innovation had diffused to a certain level of market share before the tipping point of the “winning” provider, this providers subscriber base seemed remarkably resistant. The consumers staying with this “loosing” provider in one case perceived a value/benefit **three times** higher than the ones staying with the winner. This can be explained through the structure of the social networks. Probably, the consumers staying with the “loosing” provider were connected in a cluster of the social network where most (or all) of the consumers used this provider. The ties within this cluster were strong, but the innovations did not manage to diffuse out of this subgroup.

The phenomena is further discussed by Abrahamson and Rosenkopf[AR97] as “boundary pressure points” and “boundary weaknesses” in the social network. In Figure 8.1, we have the following setting: two nodes that are not in the main clusters have a high influence on their connections (i.e. early adopters or innovators). They have both chosen the subscription that is common in Cluster B. The two nodes and a third node within Cluster B (the lighter node) then creates a *pressure point* to Cluster A (at the darker node of Cluster A). If their influence on this node is strong enough, and the pressure point node of cluster A has enough influence on its connections, this could create a change in Cluster A. Further, since the darker node of Cluster A has much influence on its connections, it also influences the node in Cluster B. This creates a *boundary weakness* in cluster B.

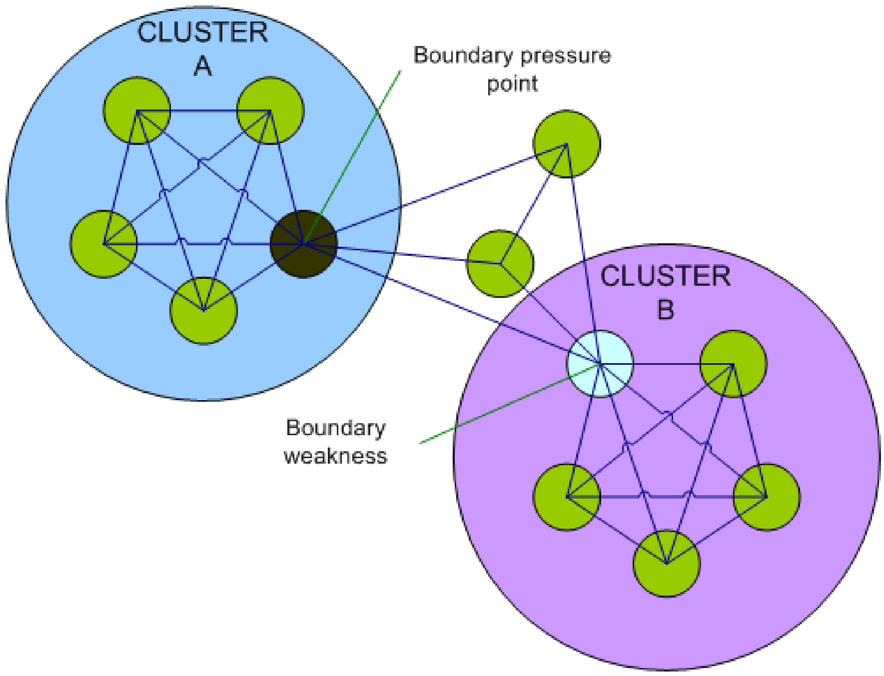

Figure 8.1: Boundary pressure points and weaknesses

This leads to a tense situation between the two clusters. Considering a much larger network with hundreds of larger clusters and even more smaller clusters, this can create highly dynamic situations where small changes make tremendous differences. However, if the pressure on such pressure points is not high enough, smaller groups may settle with another provider than the majority because of the strong ties within the subgroup.

8.2.3 Hypothesis 3

Hypothesis 3: If the companies compete equally through prices, and no innovation efforts are present, no company will stick out as the natural leader in the market.

The main reason for this hypothesis is to test how the "react" strategy of the model works. Through this strategy, companies can compete by lowering their prices. The binding fee can be decreased and increased, but it is not allowed to increase the subscription fee (it can only be decreased). By allowing for "predator behavior", the companies will not settle with a shared market, but will compete until one of them gains more than 80% of the market.

8.2.3.1 Settings

Initial (unbound) population: 0

Service providers: (3 companies)

Table 8.7 shows the settings for the three service providers of this simulation. All three providers had the same configuration, except their names.

Table 8.7: Hypothesis 3: Service provider settings

Parameter	**Value**	**Variations**
Name	Company 1, Company 2, and Company 3	
Initial subscribers	10000	
Binding fee (initial)	100	
Subscription fee (initial)	100	
Strategy	REACTIVE	
Economic flexibility	10 (max)	
Predator behavior	ON	

Consumer parameters (from configuration file):

Table 8.8 shows the configuration of the consumer parameters for this simulation.

Table 8.8: Hypothesis 3: Consumer settings

Parameter	Value	Variations
MaxFriends	20	
MinFriends	5	
ClusteringFactor	0.45	
InnovatorInfluence	50	
EarlyAdopterInfluence	40	
EarlyMajorityInfluence	30	
LateMajorityInfluence	20	
LaggardInfluence	10	
ValueNoiseFactor	0.00	
useSubscriberNoise	false	
PlanningHorizon	12	
MemoryTime	6	
NetEffIsLocal	true	
ProviderSearchInterval	3	
ProviderInfoGlobal	false	

8.2.3.2 Results

The results from this test were as expected: the prices went down, but no consumers changed their service provider. The reason is that all service providers react the same way, lowering their prices equally to try to gain a larger market share. This results in lower ARPU for all service providers without any effects on the market shares. Since the consumers use the providers prices to estimate the benefit of a subscription, this perceived benefit also decreases.

8.3 Other Results from Testing of the Simulator

Since the hypothesis were now tested, and the results were explainable, further testing of the simulator could be done. Through several scenarios the simulators behavior was tested with different settings. Since the model was created to simulate mobile markets, the main attention was on scenarios with similarities to a real mobile market.

8.3.1 Importance of Noise

The first result found was that *noise* (see Chapter 4.3.3) was necessary to create other equilibria than "winner-takes-all". By adding noise to the valuation of subscriptions (see Appendix J for configuration), the later adopters (late adopters and laggards) are not as capable of evaluating the value of subscriptions as the earlier adopters. This is in accordance with Rogers[Rog03]. The noise creates uncertainty, and allows for the critical consumer bases to be created before one service provider becomes superior. This gave favourable conditions for the growth of clusters where "smaller" service providers were valued as high, or higher, than the "larger" ones. Figure 8.2 shows the simulation of a scenario with four innovating companies with:
a) between +8 and -16% noise (configuration setting 0.02)
b) between +12 and -24% noise (configuration setting 0.03)
c) between +16 and -32% noise (configuration setting 0.04)
d) between +20 and -40% noise (configuration setting 0.05)

As seen from these results, the noise seems to relax the market competition to some degree. The noise differs between the categories, and makes the later adopters even more reserved against changing providers. The noise curve is also made skew, with higher possibility of under-valuation than over-valuation. However, a side effect of noise was that the consumers became less rational. Figure 8.3 shows the results of simulations where the noise factor was:
a) between +28 and -56% (configuration setting 0.07)
b) between +36 and -72% (configuration setting 0.09)
c) between +44 and -88% (configuration setting 0.11)

As seen from the figure, this amount of noise makes the selection of providers a process of "guessing" rather than well-considered choice. The innovations also become more likely to be important for earlier providers than later, since the noise can even the innovative differences between the companies for later adopters.

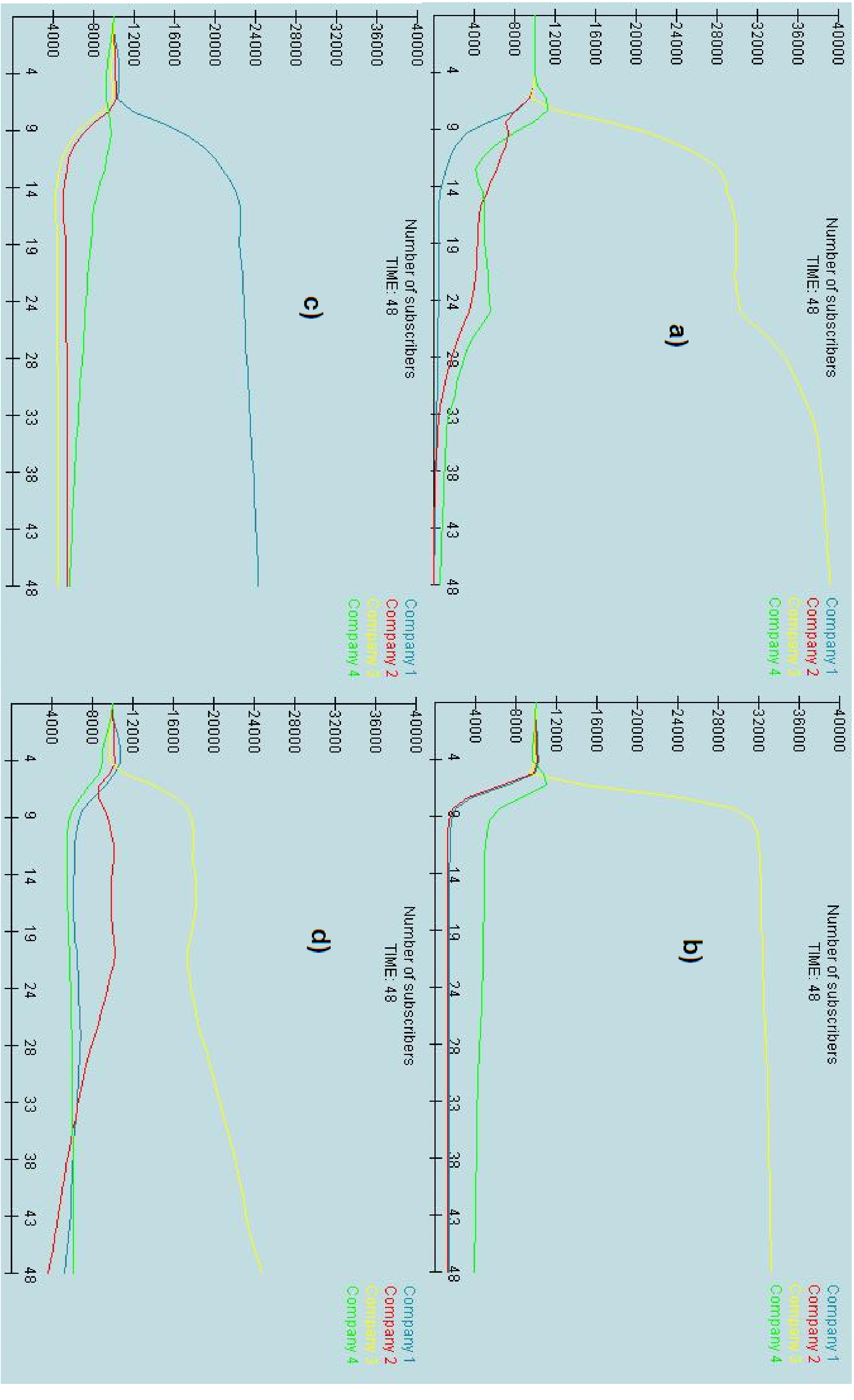

Figure 8.2: Results of Innovation with Noise

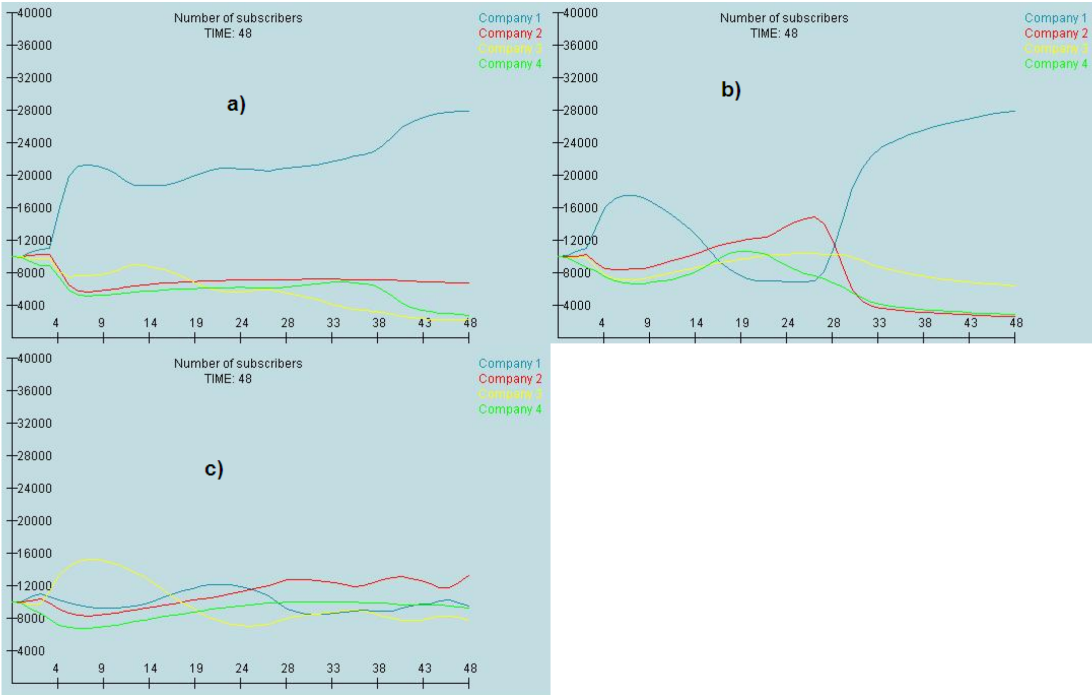

Figure 8.3: Results of Innovation with Noise 2

The four service providers in this simulation had the following configuration:

Initial subscribers: 10000
Binding fee (initial): 100
Subscription fee (initial): 100
Strategy: INNOVATIVE
Innovative strength: 10 (max)
Innovation type: 100% network effects

8.3.2 Highly Competitive Markets

Another interesting test was to see how a market of *highly competitive companies* perform when there is some noise in the market. The noise is also more natural in market with high competition, since the prices and innovations in such markets change more often than in calm markets. The noise factor was therefore set to 0.03 (giving between -24% and +12% noise to the valuation).

In the first test case, the competition between four service providers was simulated. They were each given an initial subscriber base of 20.000 subscribers, the highest degree of innovativeness, the highest degree of economic flexibility, and an equal probability of innovating services with network effects and services with intrinsic value. The results of this test are shown in Figure 8.4. As seen, the customer churn is extremely high until one company takes the lead after about 24 months. The average number of provider changes per consumer was 8.8109625 after 48 months (approximately 2 per year). The ARPU of the companies have sunk to about one third of the initial ARPU. An interesting finding from this simulation was that the consumers staying with company 3 perceived a much higher benefit/advantage than the ones staying with company 4 (the "winning" company). Again, the social networks play their role in the simulation: because of the clustered groups of consumers, the information about new innovations by Company 3 does not diffuse through the networks.

What would then happen in a similar scenario[2] where information is more easily diffused? To analyze this, a scenario with the same service providers was simulated, but this time with a lower clustering factor (0.1 instead of 0.45) and a higher number of allowed friends for the consumers ([5,40] instead of [5,20]). This gives networks where consumers are looser connected in several smaller groups, and the boundary pressure points are thereby more exposed to influence. The results are documented in Figure 8.5. In this case, the average number of provider changes per consumer had increased to 10.8827625. Once again, the ARPU decreased due to price competition, and the customer churn rates showed similar tendencies (i.e. they calmed after one provider took the lead). The main difference between this scenario and the original one was that the peak of the customer churn rate was more compact (between month 9 and 24), but was not as damped in the succeeding months.

[2] The scenario must be said to be similar, not equal, because of the chaotic and stochastic processes in the simulations.

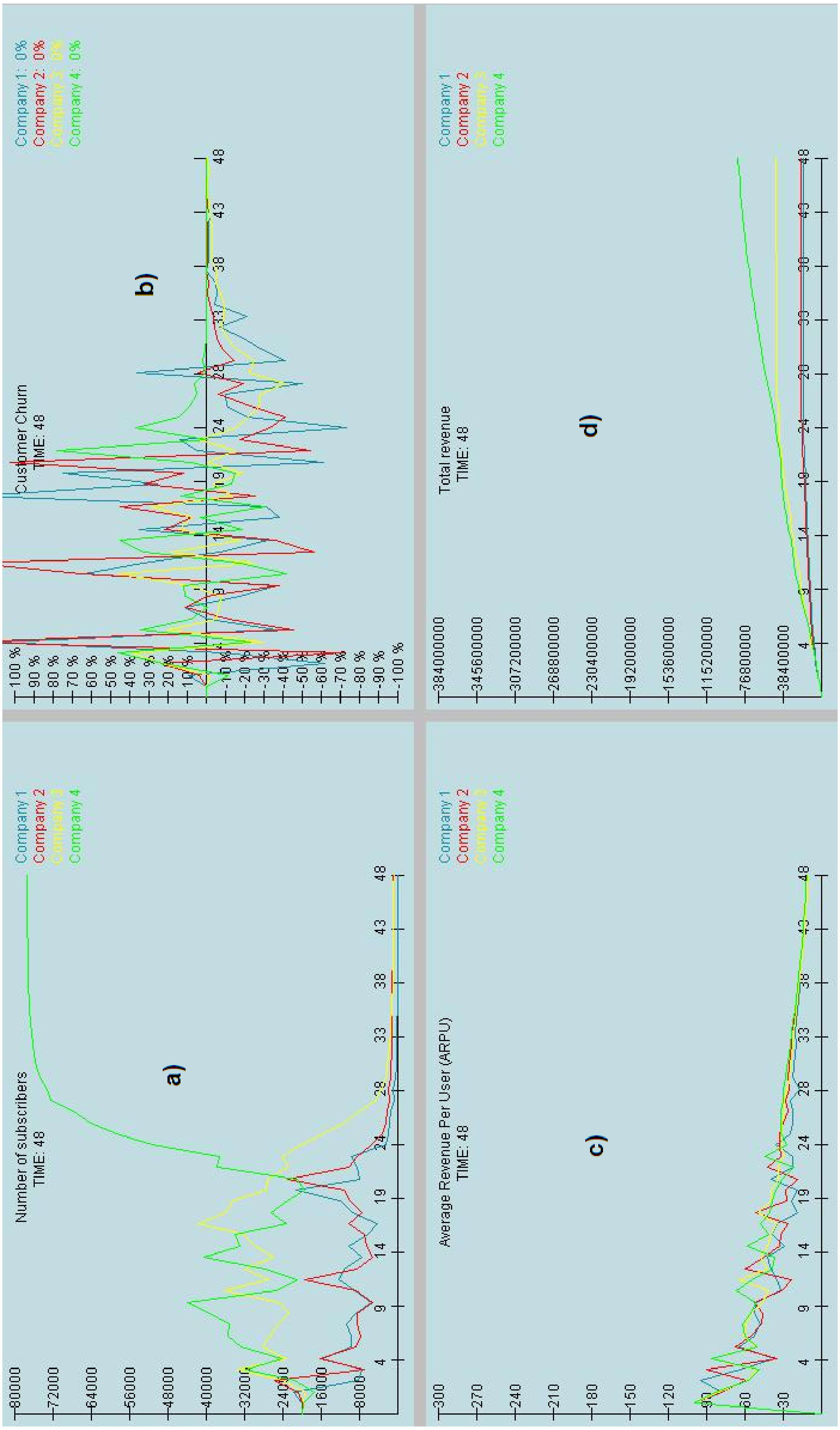

Figure 8.4: Highly Competitive Market 1: a)subscriber share, b) customer churn rate, c) ARPU, and d) Total revenue

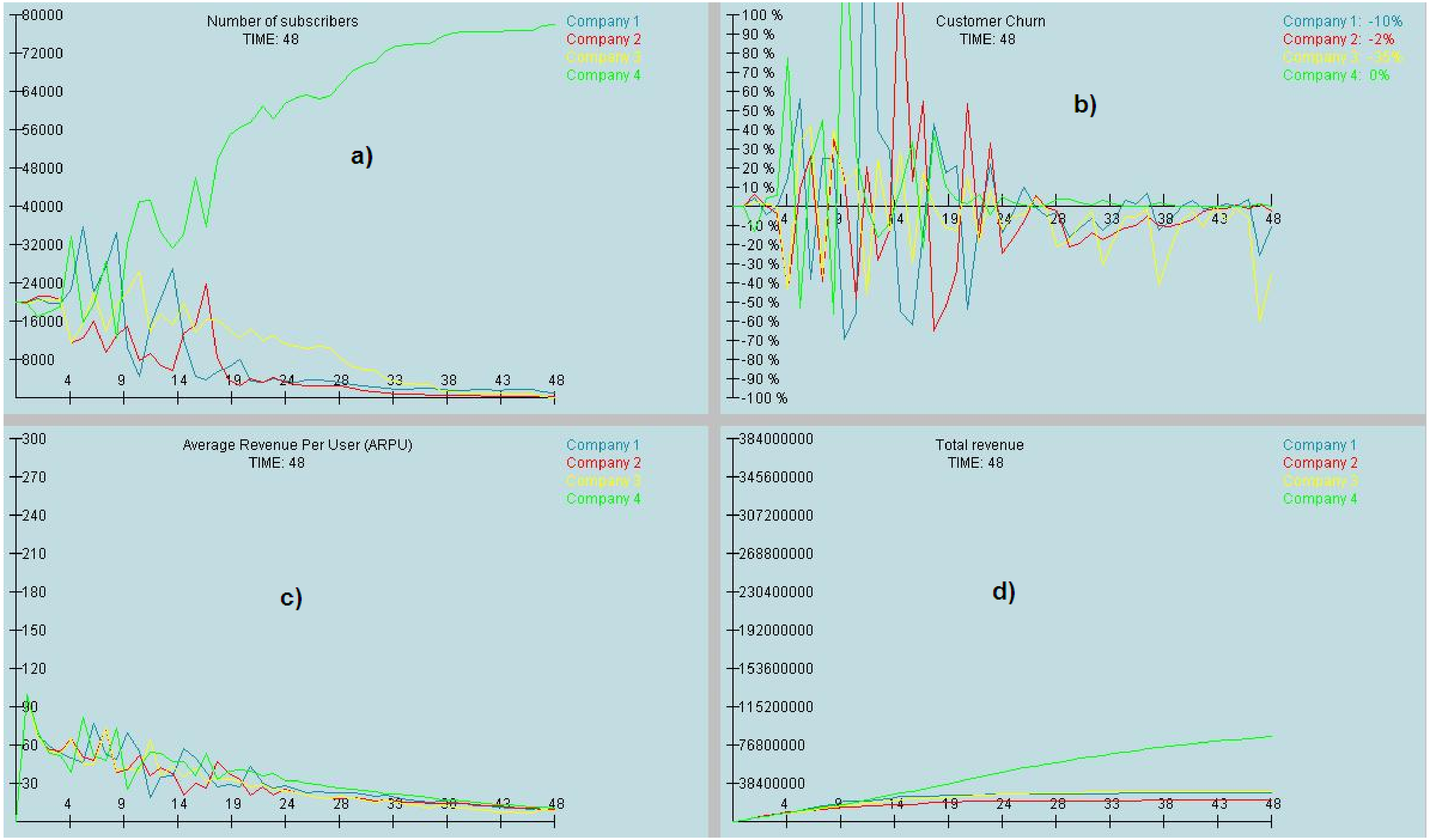

Figure 8.5: Highly Competitive Market 2: a)subscriber share, b) customer churn rate, c) ARPU, and d) Total revenue

8.4 Possible Sources of Errors

Since a computer simulation of any phenomena includes an *abstraction* of the phenomena itself and *subjective selection* of what to include and what to exclude in the simulation model, there will always be sources of errors in simulations. It is not always necessary to remove all such errors, but it is important to recognize them if analysis of the results are to be of any value. The categories given by Robert Piché[3] are used to diagnose possible problems in the model.

8.4.1 Idealisation

Idealisation errors come from discrepancy between the physical situation and the computational or mathematical model. In this thesis, recognized theories have been combined to create the model. However, since the simulation is of human behavior it is impossible to create a model that reflects this phenomena 100%. Both the behavior of the companies and the consumers are idealised, and may thereby be sources of error in the simulation.

8.4.2 Errors in Data

Sometimes the mathematical model contains real coefficients and constants that have to be supplied based on measurements. When these measurements are not correct, the model is not correct either. In the simulation program, several parameters have been used to make the model modifiable. These parameters are extracted to the configuration file, but no "real data" have been used to verify the values.

One example of where this kind of errors is likely, is in the influence the consumers have on each other. Influence is measured as a number between 1 and 100, where 100 is the highest influence. The "innovators" have the value of 50, meaning that they influence their friends by a value of 50. The laggards on the other hand have an influence of 10. It is necessary for a consumer to receive 100 in "influence" from its friends before a service provider is considered at all in the model. These levels are therefore a likely source to errors if the model does not reflect the real world.

8.4.3 Truncation Error

A truncation error may appear when a function of a continuous variable is approximated by a function of a discrete variable. In the simulation model, time (a continuous variable) is divided into discrete time steps. The program compensates to some degree for this error by randomizing the times at which the agents

[3] `http://virtual.cvut.cz:8080/dynlabmodules/ihtml/odl/partners/tut/unit1/` (Visited 2007/05/08)

make their decisions, but the simulation will still be just an approximation of real time decisions.

8.4.4 Roundoff Error

When calculating mathematical expressions, the computer rounds of the numbers. The results are therefore approximations of the real results. However, in the simulation model of this thesis, such small differences are not considered to be of a magnitude large enough to make relevant differences in the results.

8.4.5 Bugs and Blunders

There are of course possibilities of human errors by mathematical modellers, programmers, and users. In all stages, from design through usage, it is possible to make mistakes. As explained previously in this chapter, the code is inspected for bugs throughout development. Blunders, on the other hand, are not as easy to find. Since this thesis is written by one person, the selection of algorithms and relevant theory is highly subjective and hence exposed to blunders.

8.5 Discussion of Results

First of all: the simulation model *is* highly simplified in relation to a real market. For instance, innovations are usually developed by one service provider and then adopted by other service providers either through buying the technology or development of similar products. In the model of this thesis, all innovations are unique and competing.

Secondly, my model contains no compatibility between the innovations of different service providers. In the real world, mobile innovations with network effects often communicate across service providers.

Third, the innovation process is highly idealized. Even though it follows recognized theory (chaotic exploration phase, followed by sequential exploration phase), it assigns stochastic values to the innovations. This means that the results will differ for each simulation, and it is the *trends* shown by several simulations of the same scenarios that must be considered.

A *strength* of this thesis, compared to previous work, is the size of the agent base. The work by Abrahamson and Rosenkopf [AR97] contained 21 agents, and the work by Lee, Lee, and Lee [LLL06] contained 1.000 agents. As comparison, the simulations performed in this thesis contained over 90.000 agents. By involving so many agents in the simulation, the stochastic processes are represented by a larger spectrum of the possible values, which increases the probability that two subsequent simulations will show the same trends. However, each simulation took more time (between 15 and 30 minutes for the larger simulations), which made it impossible to perform as many simulations of each case as previous work (the article by Lee, Lee, and Lee builds its conclusion on 10.000 runs of each simulation scenario).

All this taken into consideration, there are several valuable findings made in this thesis through the simulations. The following sections contain discussions of these findings and the implications they may have on existing theory.

8.5.1 The Consequences of Sequential Innovation Launch

This thesis is intended to follow up the work by Lee, Lee, and Lee[LLL06] and Abrahamson and Rosenkopf[AR97]. A conclusion from the work by the former authors was that the "winner-take-all" hypothesis depends on how customers interact with one another, and that a high degree of clustering increases the probability of other outcomes. Their conclusion was based on simulation of *simultaneous launch* of *two* competing technologies/innovations. The intention was to follow up, and possibly verify, that this conclusion also holds when innovations are launched subsequently and where several technologies (more than two) compete.

Through simulations it was shown that the winner-take-all hypothesis *is* the most likely outcome in a *perfect market of rational actors*. A high degree of clustering gave subsequent innovations a *higher* probability of gaining a subscriber base as long as they were launched soon after the first one. However, the diffusion rate of the first launched innovation made it hard for the subsequently launched innovations to maintain this subscriber base.

The following argumentation can explain these results:

The first consumers to adopt a new technology are the innovators, followed by the early adopters [Rog03]. These consumer categories are more likely to have more connections than the later categories [Rog03, Gla01]. They are also likely to have a higher influence on their connections than do later categories, because of their reputation as rational desicion makers [Rog03, Mar78]. Putting this together, we see that the first service provider to launch an innovation will most likely gain an initial consumer base consisting of consumers with 1) many connections and 2) high influence.

This demonstrates the importance of being the first to launch an innovation when the market contains several companies competing for the same consumers. However, as the simulations also showed, later entrants may still be competitive if their innovations are considerably better than the one first launched (see Figure E.5d)). Since the earlier adopters have a better ability to search for new innovations and see their real value, they may act as change agents for *other* technologies if the one first chosen becomes inferior.

Lee, Lee, and Lee [LLL06] further explained how *shortcuts* that bridge the gaps between remotely located subnetworks can attenuate the local bias over time. The simulation model of this thesis contains no parameter governing the number of shortcuts, but is instead intended to create social networks imitating the characteristics of real social networks. As shown in Chapter 7, the "shortest paths" between randomly chosen consumers satisfied the well known assumption of six degrees of separation.

8.5.2 The Significance of Consumer Categories

If all consumers were to behave equally, the simulations would contain less stochastic processes. This would make it easier to draw conclusions from the results, since the results would be easier to analyze. However, by ignoring the differences in consumer behavior, false conclusions may easily be made.

In this work, the categorization by Rogers [Rog03] has been used to define the behavior of the consumers. By allowing the consumers within each category to behave autonomously rather than following static constraints makes the model more stochastic. On the other hand, it may increase the credibility of the results.

The consumer categorization showed its importance throughout testing of the simulator. Both the differentiation in number of friends and the influence factor were shown to influence the results of simulations. The results indicated that the simulator behaved according to the theory by Rogers[Rog03], producing the well known S-shaped curve. This result was promising, since the simulator only uses the statistically documented results on *consumer behavior*, and not the proposed theory on innovation diffusion itself.

However, since the consumer behavior is parameterized, the values of the parameters are also important. The values are mostly configurable through the configuration file, and the values recommended in Appendix J are based on both theory and personal suggestions. To increase the value of the simulator, further research in parameterization of user behavior should be performed.

8.5.3 The Importance of Noise

One of the first assumptions made in this thesis was:

Assumption 1: The consumers valuation of a basic subscription is a function of the prices for available subscriptions in the market that the consumer is aware of.

This assumption lays as a foundation for the consumers valuation, and thereby choice, of service providers. It is not an unreasonable assumption: consumers tend to assume value based on price. However, the assumption is based on rational consumers that make well considered choices.

There are two kinds of theories on human behavior[Mar78]:

Descriptive/behavioral: These theories purport to describe *actual* behavior of individuals or social institutions

Prescriptive/normative: These theories purport to prescribe optimal behavior

The initial behavior model of the consumers in the simulation model was normative. It assumed that all consumers think the same way, and that they value the subscriptions equally. Further, it assumed that consumers based their valuation on a *perfect discounting* (see Equation 4.7) of future benefits. March [Mar78] explains that rational choice involves two guesses: "... a guess about uncertain future consequences and a guess about uncertain future preferences". In the simulation model, the future consequences are the assumed return on investment, and the future preferences are assumptions that the preferences of other consumers will be static. However, if the model is to reflect a real market we need to put some slack on these assumptions. Consumers have differences in tastes, intellectual capability, perspective, and so on.

By bringing *noise* into the model, we abstract these kinds of differences. This makes the behavior based on "bounded rationality" instead of absolute rationality: the consumers still make rational choices, but the fundamental assumptions they base their choices on and the goals for the choices themselves differ. As March[Mar78] puts it: "*Human beings have unstable, inconsistent, incompletely evoked, and imprecise goals at least in part because human abilities limit preference orderliness*". Further, the model encompassed what is known as "management of preferences". This means that the choices made at a certain time will most likely affect future preferences for the same choice in a positive way. To include this in the model, a smaller noise variation was allowed for the valuation of the current service provider than for the alternatives. This will increase the probability of a higher valuation of the current service provider.

What could be seen from the simulations was that the introduction of noise made the market less competitive. The structure of the social networks did not by themselves create other equilibria than "winner-takes-all". In a market of perfect information and perfectly rational actors, the winner-takes-all hypothesis is still valid. However, when combining social network topologies **and** bounded rationality, other equilibria may occur. It is not noise itself that invalidates the winner-takes-all hypothesis, as this would only create chaotic markets of non-perfect choices. The noise has most impact on the choices of later adopters, making it more possible that subsequently launched innovations are considered by these, since the effect of earlier launched innovations do not get the same impact.

The implications of these findings are important: not only should innovation theory be affected by theory on social networks, it should also be affected by theories of choice. Both these subjects belong to the field of sociology, and it shows the importance of combinations of the two fields of economy and sociology when innovation is to be studied. Simplifications may easily lead to false assumptions (like the winner-takes-all hypothesis has lead to focus on getting-big-fast strategies in innovative companies).

Current *management research* also show that "get big fast" strategies are not always reasonable. Most recently, Sterman et. al. [SHBN07] highlighted the risks of ignoring the role of disequilibrium dynamics and bounded rationality in shaping competitive outcomes. Their work focused on how the bounded rationality of *firms* (rather than consumers) can lead to outcomes differing from those predicted by standard neoclassical models.

8.5.4 The Influence of Network Topology

Early theories on network effects focused on the importance of such effects on the diffusion rate of innovations. However, they ignored the fact that network effects depend on the *actual shape* of the networks between the users of the services. A common example for network effects is how the utility of having a fax machine depends on how many other users there are in the world [EH94, KS94]. Reconsidering this example, we see that the reality is more complicated. The value for *one specific consumer* lies not in the total number of other consumers using the innovations, but how many of *this consumers connections* are using it. If a consumer (person or company) mainly communicates with a limited group of other consumers, then it is the usage of the innovation *within this group* that is important to the consumer.

Current work on innovation diffusion and network effects has just recently started to focus on the importance of network topology. Although this is one step in the right directions, there is still a need to bring a third dimension into the innovation theories. Both network effects *and* network topology depend on sociological theories of consumer categorization and behavior.

The differences between the economic and sociological theories still create a barrier for research. The mathematical models of economy cannot easily incorporate the theoretical, often verbal, theories of sociology, and vice versa. This thesis suggests that computer simulation is one way of combining these fields, and thereby increase the understanding of innovation diffusion.

Chapter 9

Evaluation and Further Work

"*The best way to have a good idea is to have lots of ideas.*"
- Linus Pauling

This chapter concludes the report by presenting and evaluating the work that is done. It presents the contributions, and what should be done of further work to succeed these contributions.

9.1 Contributions

Through this master's thesis, a simulator for innovation in the Norwegian mobile market has been developed. Four highly relevant simulation methods were evaluated, and the choice fell on Agent-Based Computational Economics. This was found to be the most dynamic simulation method, and it allowed inclusion of more aspects of innovation theory than the other methods would have done.

The design is based on a thorough analysis of sociological and economic aspects of innovation and mobile services. In Chapter 4 the theoretical background for the simulation model was described, and this is by itself a contribution. Since the work is based on economic and sociological theories, much work was spent on understanding of these fields. The goal of Chapter 4 was to explain the underlying theory in a way easily understandable for a broad audience of future researchers.

Another aspect of the thesis was to make the simulator modular and reusable in several settings. The architecture reflects these goals of the thesis. The simulator itself has been wrapped into an architecture that includes aspects such as logging, configuration management, communication interface and simulation controll. This architecture is also reusable for other similar simulators, by changing the content of the "Economy Simulation Layer". Also during the implementation, the layers and modules of the simulator were made as independent of each other as possible, to allow for reuse of individual parts of the simulator.

It is only the recent years that it has been possible to do extensive research on complex networks, or as Strogatz[Str01] put it in 2001: "*Researchers are only now beginning to unravel the structure and dynamics of complex networks*". Computer simulation allows for research on extensive social networks, only limited by the capacity of the computer the simulation is running on. However, even the most recent work has been limited in the size of their social networks [AR97, WS98, LLL06]. The simulator developed in this thesis has been tested for up to 150.000 agents, and is shown to produce social networks reflecting the characteristics of *real* social networks.

The findings made in Chapter 8 opens for much future research. The consumers behavior was differentiated, and the parameters used for differentiation were analyzed to see how they affected the diffusion of innovations. This seems to be novel, and is an *extension* of the most recent work on social networks and innovation diffusion (such as [AR97, WS98, LLL06]). Until now, research has focused on consumers as actors with either equal or stochastically distributed preferences and qualifications. As I have emphasized in this report, consumer behavior is an area where the sociological and economic perspectives of innovation diffusion should be combined as this can increase our understanding of innovation diffusion dramatically.

I have suggested that the *characteristics* of consumers have a strong influence on diffusion when several innovations are launched sequentially, making a "winner-take-all" outcome more likely. The characteristics of early adopters give them a higher impact on the final results than what the later adopters have. On the other hand, these adopters are also more aware of what happens in the market and may easily change to another provider if the initial choice is found to be inferior. This means that much of the most recent theory on innovation diffusion through social networks may need to be reconsidered, as the effect of early launch can be more powerful than previously assumed. I believe more research on the characteristics of early adopters can enrich our understanding of innovation success and failure. Especially, their function in social networks should be further studied to see how much influence they actually have on their connections.

Further, I found that *noise* in the market allows for several service providers to co-exist, even under conditions of high innovative competition. The combination of imperfect information and bounded rationality enables the consumers to make poor choices. Even though these choices are poor initially, staying with the selected service provider may give the consumers a higher benefit than they could receive by changing. An interesting question is then in which directions the current developments in the "information society" will affect the mobile market in the future. New "objective" information channels such as *newsgroups* and *blogs*, and social networking sites such as Facebook (http://www.facebook.com) and LinkedIn (http://www.linkedin.com), may enable information to float faster between the clusters in social networks. More credible information provided through non-profit organizations, such as the Norwegian "Post- og Teletilsynet"'s *Telepriser* (http://www.telepriser.no) may reduce the noise created by marketing. The new generations interest in technological innovation may change the traditional distribution of consumer categories. After all; the theory by Rogers originate from the adoption of hybrid seed corn by farmers in the 1950s.

Finally, the bibliography of this report contains references to both classic and modern innovation research. The references are selected based on their recognition and novelty, and should be considered by future researchers if this thesis is to be followed by further work.

9.2 Evaluation

The *main goal* of this thesis was to develop the simulation software, to enable simulation of innovation in the mobile market. The main focus of the simulator has been to see how innovative efforts will affect markets under different circumstances. This goal has been fulfilled by performing a thorough analysis of the most recent research on innovation. The bibliography of this report contains references to much of the most recent and recognized research on innovation, and specifically innovations with network effects.

The simulation software has been thoroughly tested. An earlier version[1] of the software is also used in a bachelor thesis by three students at NITH, while the newest version has been integrated with a user interface accompanying this thesis. This shows that the communication interface is enabled for usage by a range of applications.

The simulator is influenced by theories from different fields of economics, sociology, and mathematics. The reason is that no single source of information could be found that covered the entire field that was to be simulated. The selection of the theories is based on **1)** their novelty, and **2)** their obtained recognition.

In Chapter 2, I stated a set of research questions. The following is a summary of how these are answered through this thesis:

1. Which simulation method is best suited for simulating innovation in the mobile market?

This was answered in Chapter 3 through a presentation of four possible simulation methods, from which ACE was chosen as the most suitable for use.

2. Which economic theories should be included in the simulation model?

The economic theories chosen are documented in Chapter 4. As explained, these theories were selected by their novelty and recognition. It would of course be possible to include more theory in the model. However, it was important to keep the number of different theories at a level where the results would be possible to analyze.

[1] The delivery of the bachelor thesis was at an earlier time than the delivery of this thesis. An earlier version of the software was therefore tested and assured stable for usage with their application.

3. Which sociological theories should be included in the simulation model?

As with the economic theories, the sociological theories are documented in Chapter 4. The main influence of this thesis has been the work by Rogers[Rog03], which is recognized as the pioneer of diffusion of innovations theory.

4. How should the consumers be modeled to best reflect a real market?

This research question is answered partly through research questions 2 and 3. Further, it is throughout this report explained which assumptions that are made for consumer behavior. Some of this behavior, such as valuation of subscriptions, is based on the authors own experiences and discussions with friends and family. The architecture also contains a model of the agents, which shows how the agents are simplified to consist of certain properties and actions.

5. How should the service providers be modeled to best reflect a real market?

This question is basically answered as with research question 4. However, the behavior of the service providers is probably even more simplified than the consumer behavior. The reason for this is that complicated service provider behavior could easily make the results more ambiguous. Their behavior is therefore kept as simple as possible, while still allowing for some "intelligence".

6. How can the actors be parameterized in a way that 1)makes the simulator as realistic as possible, and 2)minimizes the configuration of each simulation?

Much of the agent behavior depends on input values. For instance, as mentioned in Chapter 8, the influence each consumer has on its friends is parameterized. There are two sides of parameterization:

- It allows us to perform sensitivity analysis of the parameters to see how they affect the results
- It makes it more difficult to simulate "real" markets

The parameters of the simulator are provided through two interfaces. First, the most frequently changed parameters are provided through the communication interface, allowing them to be easily changed for each simulation. Secondly, the less frequently changed parameters are read from the configuration file. The balance between too much and too little configuration is hard to define, and the chosen approach was what seemed most reasonable from the authors point of view. By allowing for configuration through two interfaces, more parameters could be specified, and the simulator could more easily be tuned to perform more realistic.

9.3 Further Work

The simulator allows for testing of other scenarios than the ones described in this report. The findings made in this thesis are just scratching the surface, proposing fields open for more research. The focus of the testing in this thesis was to gain a broad understanding of how different parameters affect the diffusion of innovations through social networks.

One of the first things that should be done in further work is an evaluation of the current simulation model and the implemented software. Since this thesis was written by one person, it is likely to be affected by subjective selection of theory. It is also exposed to blunders both in source code and assumptions. Although I have tried to be objective on all fields, and the code is thoroughly investigated, such errors may be present.

An interesting extension of the simulator would be to tailor it for marketing research. Recent work by Maienhofer and Finholdt [MF02] had some of the same focus, as they used computer simulation to find the best targets for change agents. As I have suggested, the characteristics of consumers have a large impact on the diffusion. Simulation could show what would happen if marketing efforts were concentrated on specific consumer categories. Would it, for instance, be more valuable to focus marketing on earlier adopters or later ones? This could easily be simulated by simple modifications in the model.

As mentioned earlier in this report, many of the parameters used for consumer behavior are configurable. An interesting task for further work would be to analyze which values are most likely within real mobile markets. For instance, it would be possible through questionnaires to analyze how many communication partners are common for each category, and how many of these the consumers would actually discuss mobile innovations and subscriptions with and give recommendations to. Since the field of interest is specified (the mobile market), it is easier to collect such information than for a general case of innovations. As many of the parameters used in the simulator as possible should be collected in a such a way. If the "base case" of the market is known, it will be possible to experiment with what would happen if the market changes. For instance: what parameters are the most critical as seen from the base case?

Finally, this thesis suggests several factors affecting innovation diffusion (noise, influence, number of connections, etc.). Since the implemented simulator is highly complex, it would be interesting to define specific models for further research of these factors. By investigating them isolated from the rest of the environment, it will be easier to make specific conclusions about to which degree they affect diffusion. However, this may as well be considered as taking a step in the wrong direction if the conclusions of such research are seen in isolation.

Bibliography

[Alt94] Lee Altenberg. Evolving better representations through selective genome growth. Working Papers 94-02-008, Santa Fe Institute, February 1994. available at http://ideas.repec.org/p/wop/safiwp/94-02-008.html.

[AR97] Eric Abrahamson and Lori Rosenkopf. Social network effects on the extent of innovation diffusion: A computer simulation. *Organization Science*, 8(3):pp. 289–309, May 1997.

[BCK03] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice, 2nd edition*. Addison-Wesley Professional, April 2003.

[Bla86] Fischer Black. Noise. *Journal of Finance*, 41(3):pp. 529–543, 1986. Papers and Proceedings of the Forty-Fourth Annual Meeting of the America Finance Association, New York, New York, December 28-30, 1985.

[Bux01] Peter Buxmann. Network effects on standard software markets: a simulation model to examine pricing strategies. In *Proceedings of the 2nd IEEE Conference on Standardization and Innovation in Information Technology, SIIT 2001, Boulder, CO, USA, October 3-6, 2001*, pages pp. 229–240. IEEE, 2001.

[CDW07] Gino Cattani, Alex Dorsch, and Sidney G. Winter. The value of moderate obsession: Insights from a new model of organizational search. LEM Papers Series 2007/03, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy, January 2007. available at http://ideas.repec.org/p/ssa/lemwps/2007-03.html.

[CHRW05] Christer Carlsson, Kaarina Hyvonen, Petteri Repo, and Pirkko Walden. Asynchronous adoption patterns of mobile services. In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 7*, page 189.1, Washington, DC, USA, 2005. IEEE Computer Society.

[Cm96] Yu-Ting Cheng and Andrew H. Vand de Ven. Learning the innovation journey: Order out of chaos? *Organization Science*, 8(7):pp. 593–614, 1996.

[CMO72] Michael D. Cohen, James G. March, and Johan P. Olsen. A garbage can model of organizational choice. *Administrative Science Quarterly*, 17(1):pp. 1–25, March 1972.

[EH94] Nicholas Economides and Charles Himmelberg. Critical mass and network evolution in telecommunications. *In Proceedings of the 22nd Telecommunications Policy Research Conference*, October 1994.

[ER59] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:pp. 290–297, 1959.

[For61] Jay Forrester. *Industrial Dynamics*. MIT Press, 1961.

[Fre01] Koen Frenken. Modelling the organisation of innovative activity using the nk-model. June 2001. Paper prepared for the Nelson-and-Winter Conference, Aalborg, 12-16 June 2001.

[Fre06] Koen Frenken. Technological innovation and complexity theory. *Economics of Innovation and New Technology*, 15(2):pp. 137–155, March 2006. available at http://ideas.repec.org/a/taf/ecinnt/v15y2006i2p137-155.html.

[FS85] Joseph Farrell and Garth Saloner. Standardization, compatibility, and innovation. *The RAND Journal of Economics*, 48(16):pp. 70–83, 1985.

[Gla01] Malcolm Gladwell. *The Tipping Point*. Abacus, May 2001.

[GT00] Nigel Gilbert and Pietro Terna. How to build and use agent-based models in social science. *Mind&Society*, 1(1):pp. 52–72, March 2000.

[Kau93] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, May 1993.

[Kle95] Jack P.C. Kleijnen. Verification and validation of simulation models. *European Journal of Operations Research*, 82:pp. 145–162, 1995.

[Küp01] Claudia Küpper. Service innovation - a review of the state of the art. University of Munich, Institute for Innovation Research and Technology Management, 2001.

[Kru95] Philippe B. Kruchten. The 4+1 view model of architecture. *Software, IEEE*, 12(6):pp. 42–50, 1995.

[KS85] Michael L. Katz and Carl Shapiro. Network externalities, competition and compatibility. *The American Economic Review*, 75(3):pp. 424–440, June 1985.

[KS94] Michael L. Katz and Carl Shapiro. System competition and network effects. *Journal of Economic Perspectives*, 8(2):pp. 93–115, 1994.

[LH99] Christoph H. Loch and Bernando A. Huberman. A punctuated-equilibrium model of technology diffusion. *Management Science*, 45(2):pp. 160–177, February 1999.

[LLL06] Eocman Lee, Jeho Lee, and Jongseok Lee. Reconsideration of the winner-take-all hypothesis: Complex networks and local bias. *Management Science*, 52(12):pp. 1838–1848, 2006.

[Mar78] James G. March. Bounded rationality, ambiguity, and engineering of choice. *The Bell Journal of Economics*, 9(2):pp. 587–608, 1978.

[MF02] Dirk Maienhofer and Thomas Finholt. Finding optimal targets for change agents: A computer simulation of innovation diffusion. *Comput. Math. Organ. Theory*, 8(4):pp. 259–280, 2002.

[Mil67] Stanley Milgram. The small world problem. *Psychology Today*, 1:pp. 61–67, May 1967.

[Moo02] Geoffrey A. Moore. *Crossing the Chasm*. HarperBusiness, September 2002.

[MR99] Alwin Mahler and Everett M. Rogers. The diffusion of interactive communication innovations and the critical mass: the adoption of telecommunications services by german banks. *Telecommunications Policy*, 23(10-11):pp. 719–740, November 1999.

[MST01] David McFadzean, Deron Stewart, and Leigh Tesfatsion. A computational laboratory for evolutionary trade networks. *IEEE-EC*, 5:pp. 546–560, October 2001.

[NWS02] M.E.J. Newman, D.J. Watts, and S.H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences USA*, 99(Supplement 1):pp. 2566–2572, February 2002.

[ot06] Post og teletilsynet. Det norske telemarkedet - 3. kvartal 2006, December 2006.

[Rog03] Everett M. Rogers. *Diffusion of Innovations*. Free Press, fifth edition, 2003. (First edition published in 1962).

[SHBN07] John D. Sterman, Rebecca Henderson, Eric D. Beinhocker, and Lee I. Newman. Getting big too fast: Strategic dynamics with increasing returns and bounded rationality. *Management Science*, 53(4):pp. 683–696, April 2007.

[Shy01] Oz Shy. *The Economics of Network Industries*. Cambridge University Press, February 2001.

[Ste00] John D. Sterman. *Business Dynamics: Systems Thinking and Modelling for a Complex World*. Irwin McGraw-Hill, 2000.

[Str01] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.

[TA86] Michael Tushman and Philip Anderson. Technological discontinuities and organizational environments. *Administrative Science Quarterly*, 31(3):pp. 439–465, September 1986.

[Tes02] Leigh Tesfatsion. Agent-based computational economics: Growing economies from the bottom up. *Artificial Life*, 8(1):pp. 55–82, 2002.

[TGG05] Nobuo Tanaka, Michel Glaude, and Fred Gault. *Oslo Manual: Guidelines for Collecting and Interpreting Innovation Data (Third edition).* OECD Publishing, 2005. A joint publication of OECD and Eurostat.

[VFS05] Hal R. Varian, Joseph Farrell, and Carl Shapiro. *The Economics of Information Technology : An Introduction (Raffaele Mattioli Lectures).* Cambridge University Press, January 2005.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):pp. 440–442, June 1998.

Appendix A

Acronyms

ACE Agent-Based Computational Economics

ARPU Average Revenue Per User

JAR Java ARchive

JVM Java Virtual Machine

NITH Norwegian School of Information Technology

NTNU Norwegian University of Science and Technology

OECD Organisation for Economic Co-Operation and Development

ROI Return on Investment

SRS Software Requirements Specification

VAS Value Added Services

XML Extensible Markup Language

Appendix B

Problem Description

Many methods and algorithms for simulation of business processes exist today. In many cases it may be difficult to decide which model is best fitted for simulating a specific scenario. Knowledge of this requires knowledge of both economy/strategy and algorithms for modelling. Further it's needed to have the ability to combine these fields in such a way that the result of the simulation is perceived as a value increase.

This task will look into simulation models for business processes. It shall be argumented why one model is chosen before the others, and a solution shall be designed and implemented. Analysis of the results related to expected results will be a concluding part of the thesis. Possible methods to investigate are including, but not restricted to; Agent-based modelling, Non-linear modelling, Social network models, System dynamics, and NK-modelling.

It is expected that the student in cooperation with supervisors and other relevant persons decides upon the most adequate method to simulate a scenario for actors in the mobile industry. Value chains and value networks that constitute this business are complex, and there are many dependencies between actors. Simulation and visualization of these processes may contribute to an increased understanding of behavior in this business, and how the actors may respond to sudden events and organize themselves to optimize their own market position. Implementation will be done in cooperation with other students, and it is expected that the student (Martin) has a leading role in the progress related to which models to implement and how results are presented/visualized.

Assignment given 15th of January 2007
by Øyvind Strømme, Accenture

Appendix C

E-mail correspondence with Lori Rosenkopf

After reading the article of Abrahamson and Rosenkopf[AR97] on social network effects on the extent of innovation diffusion, I contacted Lori Rosenkopf to get information about the most recent work on this field. This lead to the article of Lee, Lee, and Lee [LLL06] which confirms the novelty of my work.

C.1 The mail sent to Lori Rosenkopf

From: martinb@stud.ntnu.no [mailto:martinb@stud.ntnu.no]
Sent: Wed 4/18/2007 3:52 AM
To: Rosenkopf, Lori
Subject: Request for information on social network effects

Dear Ms Rosenkopf

My name is Martin Børke, and I currently write my master thesis at the Norwegian University of Science and Technology (NTNU) on innovation diffusion in the norwegian mobile market.

I found your 1997 article "Social Network Effects on the Extent of Innovation Diffusion" highly relevant for my work, especially since my work is directly related to combining social network effects and Increasing Returns theories (by Katz&Shapiro and Farrell&Saloner).

My question is: do you know if any work is done on this field since the time you published the article? I have searched for citations to your article, but have not yet found any work directly related to this field.

Yours sincerely

Martin Børke
Department of Computer and Information Science
NTNU (Norwegian University of Science and Technology)

C.2 The mail received from Lori Rosenkopf

Date: Wed, 25 Apr 2007 08:25:55 -0400
From: "Rosenkopf, Lori" <rosenkopf@wharton.upenn.edu>
Reply-to: "Rosenkopf, Lori" <rosenkopf@wharton.upenn.edu>
Subject: RE: Request for information on social network effects
To: martinb@stud.ntnu.no

Martin, thanks for your note. Most of the more recent work on network structure makes use of the Watts/Strogatz small world formulation (with beta parameter). So for example there is a recent Lee/Lee/Lee article in Mgmt Science that address some similar issues, which I think would be the closest fit to your interests. Check it out.

Best, Lori

Appendix D

Characteristics of Adopter Categories

In the book "Diffusion of Innovations"[Rog03] by Everet M. Rogers, the categories mentioned in Chapter 4.2 are further characterized under the headings *socioeconomic status*, *personality values*, and *communication behavior*. The following sections will summarize the generalizations stated in the book on pages 287–292. The generalizations made are describing the relationship between earlier and later adopters, ranging from *Innovators*(very early adopters) to *Laggards*(very late adopters).

D.1 Socioeconomic Characteristics

1. *Earlier adopters are no different from later adopters in age*
2. *Earlier adopters have more years of formal education*
3. *Earlier adopters are more likely to be literate than are later adopters*
4. *Earlier adopters have higher social status than do later adopters*
5. *Earlier adopters have a greater degree of upward social mobility than do later adopters*
6. *Earlier adopters have larger-sized units (farms, schools, companies, and so on) than do later adopters*

D.2 Personality Variables

7. *Earlier adopters have greater empathy than do later adopters*
8. *Earlier adopters may be less dogmatic than are later adopters*

9. *Early adopters have a greater ability to deal with abstractions than do later adopters*

10. *Earlier adopters have greater rationality than do later adopters*

11. *Earlier adopters have more intelligence than do later adopters*

12. *Earlier adopters have a more favorable attitude toward change than do later adopters*

13. *Earlier adopters are better able to cope with uncertainty and risk than are later adopters*

14. *Earlier adopters have a more favorable attitude toward science than do later adopters*

15. *Earlier adopters are less fatalistic than are later adopters*

16. *Earlier adopters have higher aspirations (for formal education, higher status, occupations, and so on) than do later adopters*

D.3 Communication Behavior

17. *Earlier adopters have more social participation than do later adopters*

18. *Earlier adopters are more highly interconnected through interpersonal networks in their social system than are later adopters*

19. *Earlier adopters are more cosmopolite than are later adopters*

20. *Earlier adopters have more contact with "change agents"* [1] *than do later adopters*

21. *Earlier adopters have greater exposure to mass media communication than do later adopters*

22. *Earlier adopters have greater exposure to interpersonal communication channels than do later adopters*

23. *Earlier adopters seek information about innovations more actively than do later adopters*

24. *Earlier adopters have greater knowledge of innovations than do later adopters*

25. *Earlier adopters have a higher degree of opinion leadership than do later adopters*

[1] A "change agent" is an individual who influences clients' innovation-decisions in a direction deemed desirable by a change agency

Appendix E

Testing of Hypothesis 2

The following sections present results from testing hypothesis 2, and from the three scenarios used for sensitivity analysis related to Proposition 1: *Because of differentiated characteristics of consumers, the "winner-takes-all" equilibrium is the most likely outcome when competing innovations are launched sequentially.*

E.1 Basic simulation

The initial scenario, as presented in Chapter 8.2.2, resulted in Proposition 1. The simulation of the scenario showed that the first company to launch an innovation would most likely capture the entire market. Figure E.1 a)-e) show the distribution of subscribers through time for the five runs of this simulation.

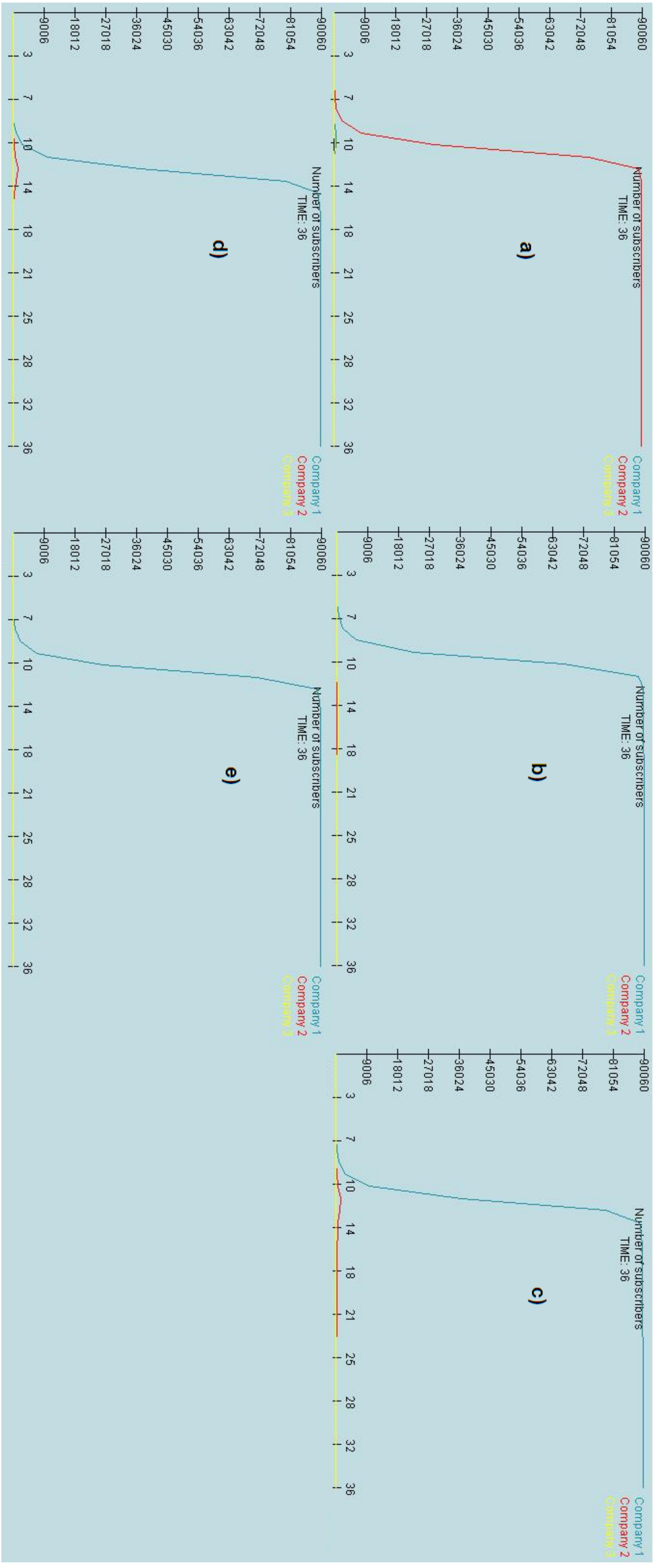

Figure E.1: Proposition 1: Basic Simulation

E.2 Sensitivity analysis: cluster size

The first factor to analyze was the size of the clusters. Lee, Lee, and Lee[LLL06] conclude that in networks with a high degree of clustering, the tendency for a lead product to drive out its rivals is lower than in networks with only small clusters. To see how the cluster sizes influenced the results, the `ClusteringFactor` was changed to 1.0. This means that it is initially 100% probability that two consumers with a common friend are also friends. This is of course not possible for all consumers in the network, but the networks were constructed with this as a main condition.

Figure E.2 shows the results from five runs of this simulation. The largest cluster sizes found in these networks consisted of 12 consumers, and the number of such sized networks for each of the simulations were **a)** 9, **b)** 89, **c)** 24, **d)** 28, and **e)** 4.

The results vary from the basic simulation in several ways. First, the diffusion happens at a faster rate: the "tipping point" occurs already at times between T=4 and T=5. Further, the clustering seems to give favourable conditions for the growth of initial consumer bases for the providers. However, the result is still one company capturing the majority of the consumers even with this degree of clustering (even if it is not the initial "leader").

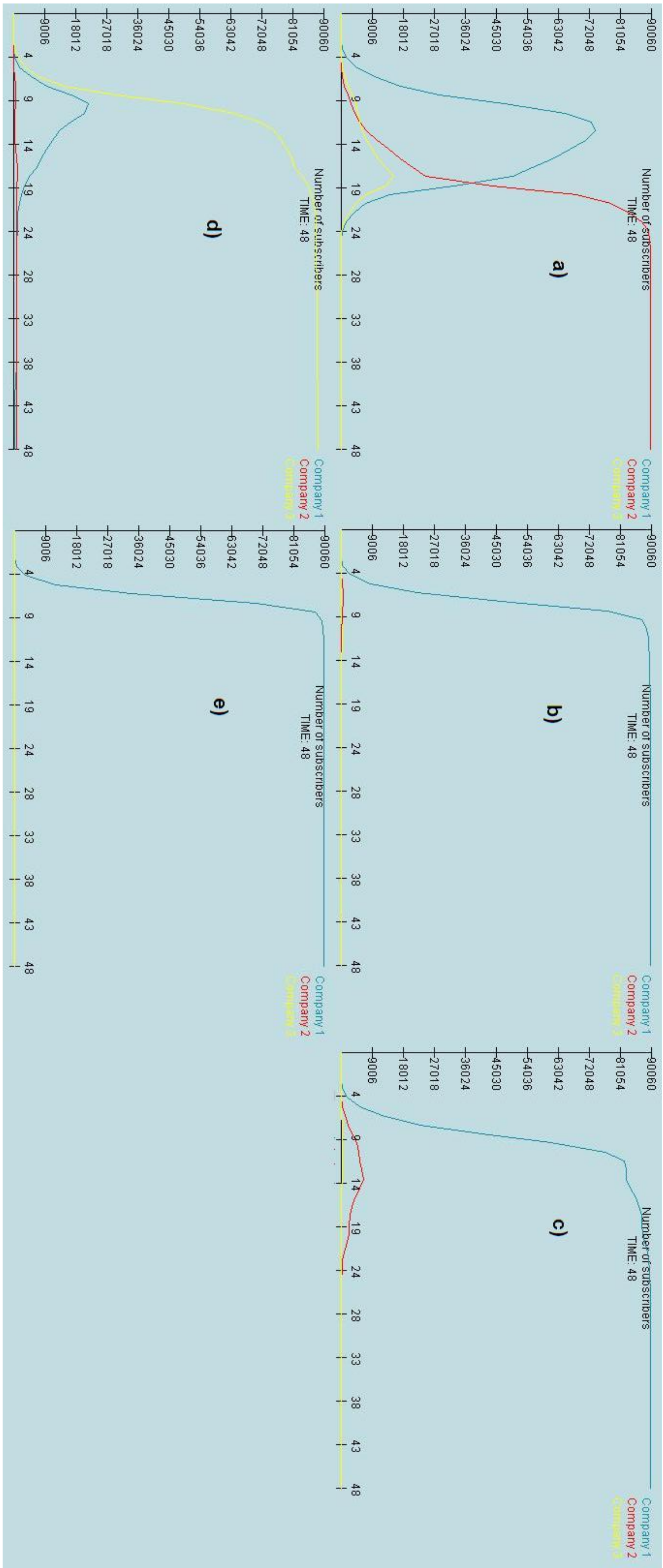

Figure E.2: Proposition 1: Sensitivity Analysis of Cluster Sizes

E.3 Sensitivity analysis: influence

The second of the three major factors thought to influence the results was the *influence* each consumer has on its friends/connections. Through the configuration file, it is possible to change the influence factor of each consumer category. As part of the sensitivity analysis, these factors were set to be similar (25) for all categories. The concept is shown in Figure E.3 where both the original *differentiated influence* graph and the *equal influence* graph are shown.

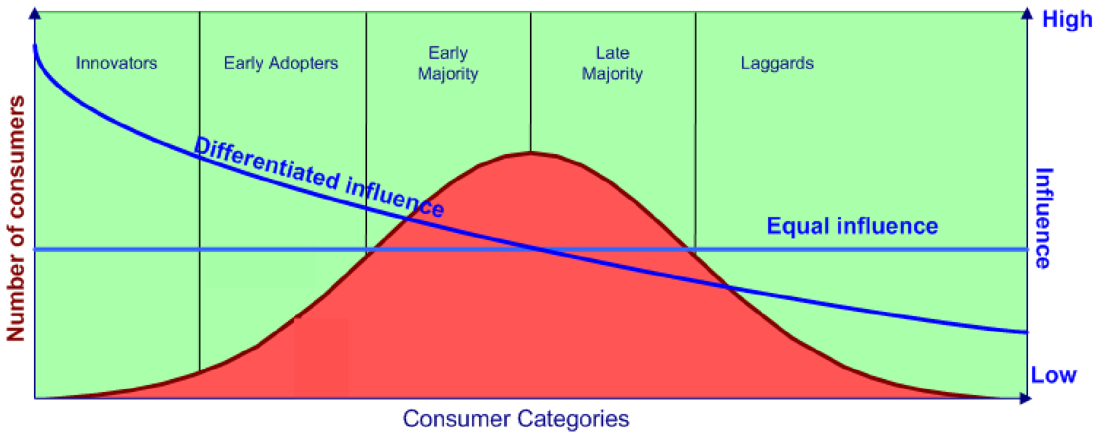

Figure E.3: Influence of the adopter categories

The results are shown in Figure E.4 a)-e). The first difference was that the "tipping point" of diffusion occurred at a much later time than in the basic scenario. In these simulations, the tipping point occurred at times between T=13 and T=14, as opposed to between T=8 and T=11 for the basic scenario. This confirms the assumption that *differentiation of influence leads to a bigger advantage for the first company to launch an innovation.*

Further, the companies seemed to have better conditions for creating an initial base and retaining it. This is can especially be seen in Figure E.4 c), where Company 2 is capable of keeping much of its consumer base for several time steps. The "peaks" of the non-winning companies are also higher than for the basic simulation, giving them better possibilities of reaching the "tipping point".

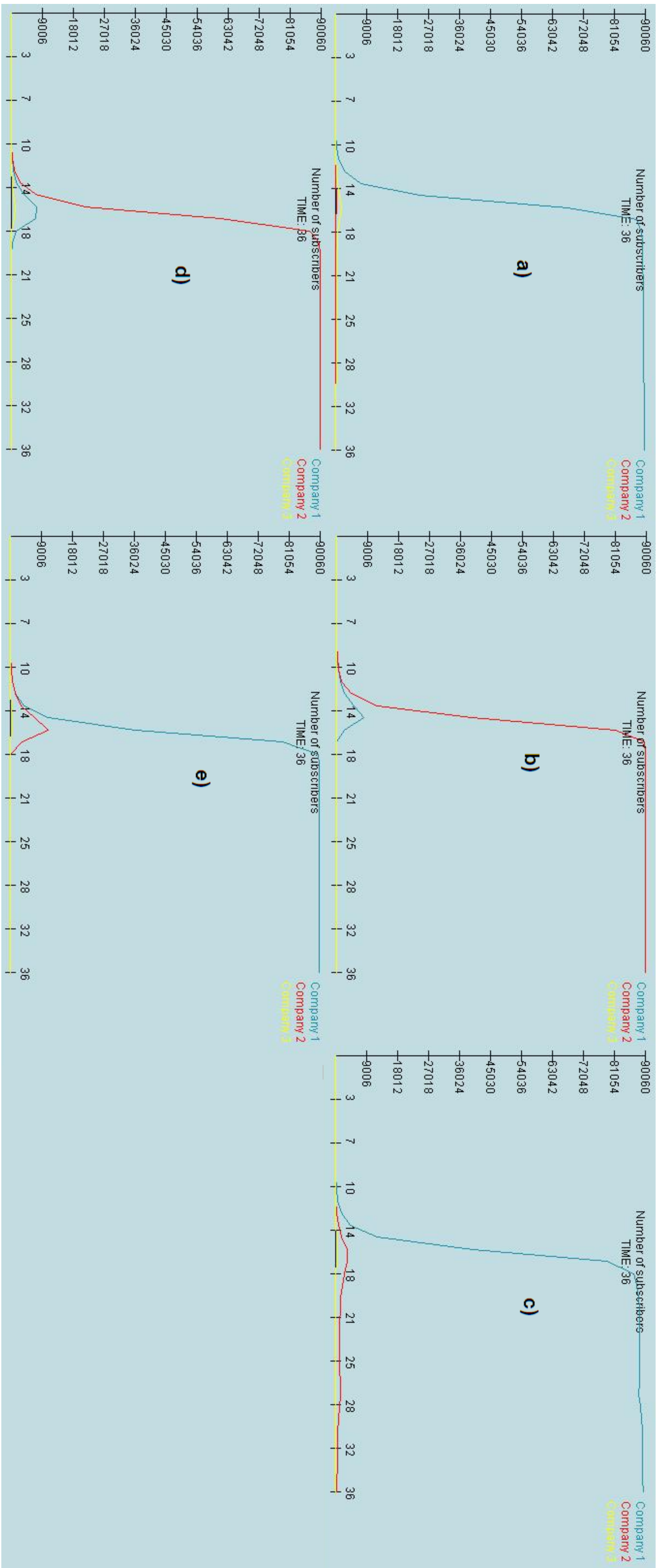

Figure E.4: Proposition 1: Sensitivity analysis of influence

E.4 Sensitivity analysis: connections

The second factor that is thought to influence the result of the basic scenario is the number of connections ("friends") each consumer has. This is also differentiated according to the theories of Rogers [Rog03]. To see how this influence the results, the range for possible number of connections allowed was changed from [5..20] to [5..40]. The results of five simulations with this setting can be seen in Figure E.5 a)-e).

The output from these simulations showed that the possibility of more connections lead to faster diffusion. This was expected, as the earlier adopters (innovators and early adopters) would now have even more connections to influence. The time when diffusion reached the "tipping point" was now between T=7 and T=8.

A remarkable effect also appeared in the simulation shown in Figure E.5 d). In this simulation, a service provider (Company 2) first achieves most of the consumer base, but is then overtaken by Company 3. The innovation history for these two companies before this event was:
T=4: Company 1 launched a new innovation with network effect 0.09684944
T=4: Company 2 launched a new innovation with network effect 0.07628754
T=6: Company 3 launched a new innovation with network effect 0.18955478
T=8: Company 2 launched a new innovation with network effect 0.00375937
T=10: Company 1 launched a new innovation with network effect 0.05455069
T=11: Company 3 launched a new innovation with network effect 0.34319194
T=15: Company 2 launched a new innovation with network effect 0.27997525
T=16: Company 1 launched a new innovation with network effect 0.01862138
T=16: Company 3 launched a new innovation with network effect 0.09466209

As seen from this history, Company 3 was very lucky in this simulation. The innovation (with high network effects) launched at time T=6 gave the service provider a portion of the consumer base. The consumers changing to this company at that time were probably innovators, since these tend to be the first to adopt new innovations. However, it was not until the innovation at T=11 that the service provider really took off. At this time it already had a (small) subscriber base, and this base helped spreading the news at a high rate (because of the number of allowed connections). The result seen from the graph can then be described as a combination of luck and network topology (number of connections).

Further, the allowed number of connections was set to be [10,10]. This means that all consumers have *exactly* 10 connections. The results from the simulation are shown in Figure E.6 a)-e). With this setting, the diffusion started at around T=10, but diffused slower than for other settings. This lead to several service providers gaining a large consumer base before one provider took the lead.

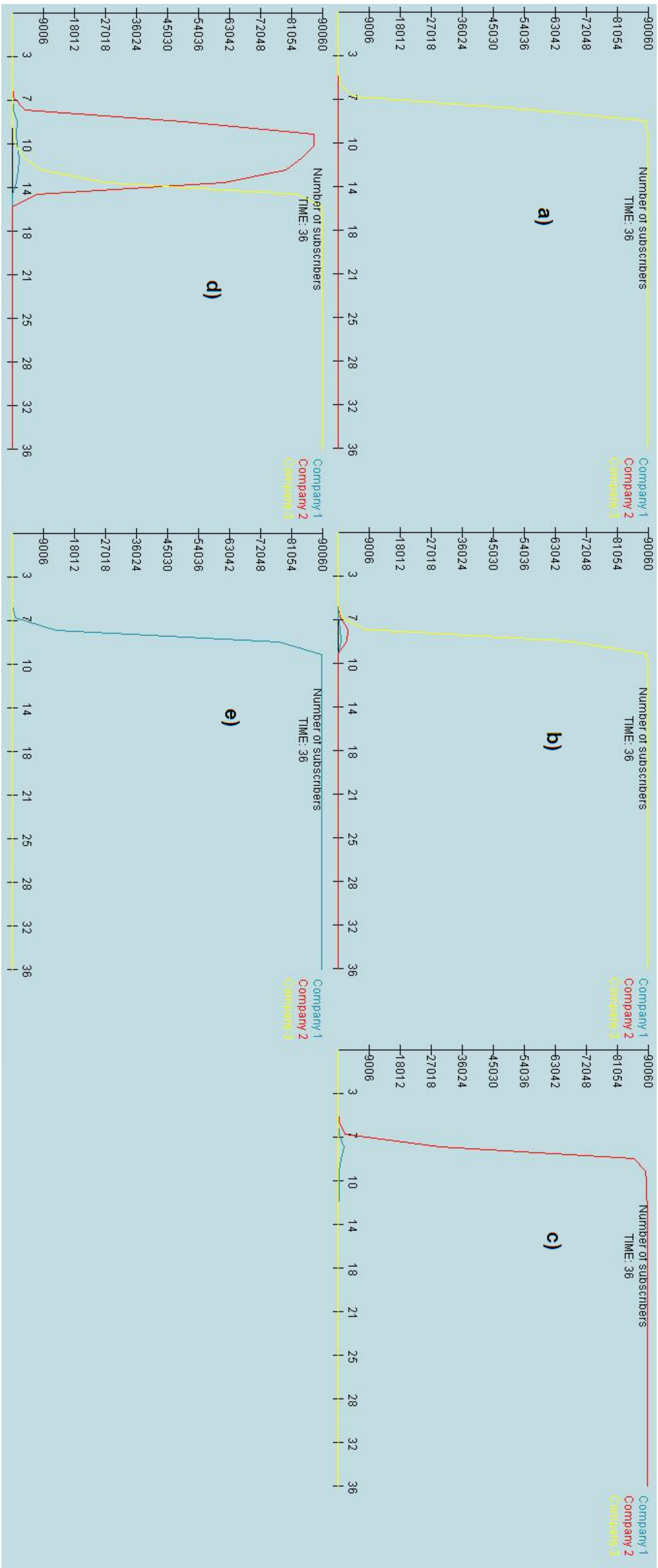

Figure E.5: Proposition 1: Sensitivity analysis of connections

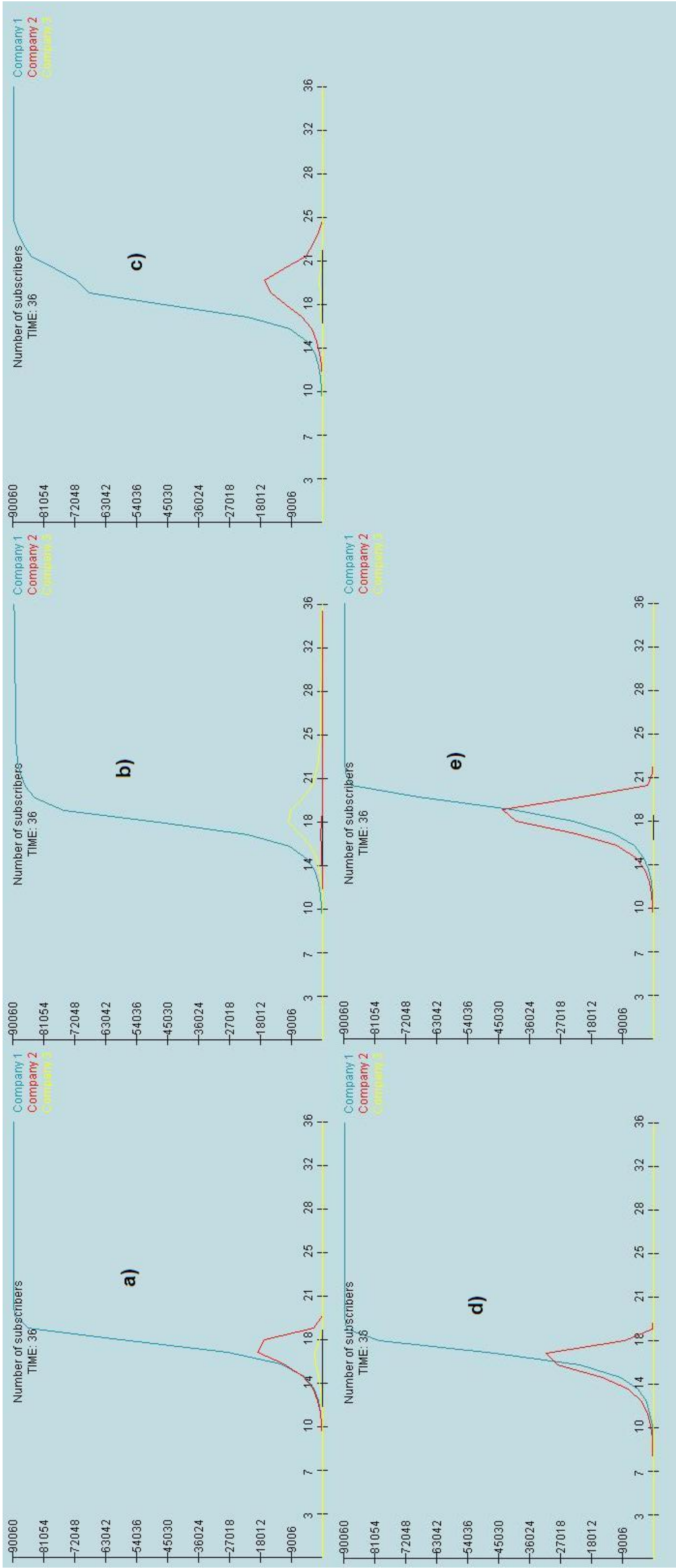

Figure E.6: Proposition 1: Sensitivity analysis of connections 2

Appendix F

Class Diagrams

This appendix contains the class diagrams generated from the final implementation. The class diagrams are divided into sections referring to the layers of the development view (see Figure 6.2).

F.1 Communication Layer

Figure F.1 shows the class diagram for the *communication layer*.

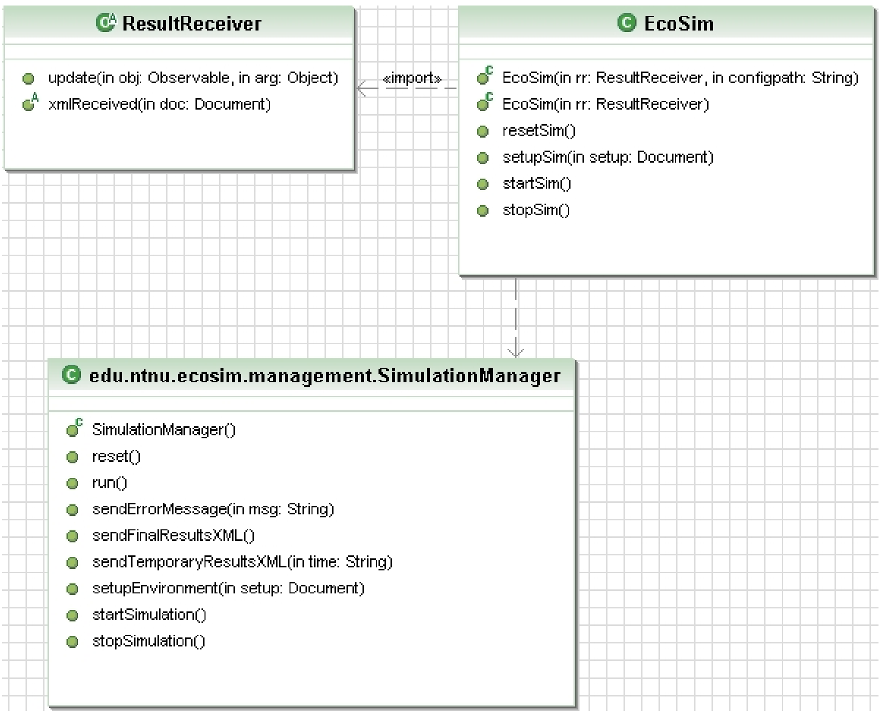

Figure F.1: Class Diagram: Communication Layer

F.2 Management Layer

Figure F.2 shows the class diagram for the *management layer*.

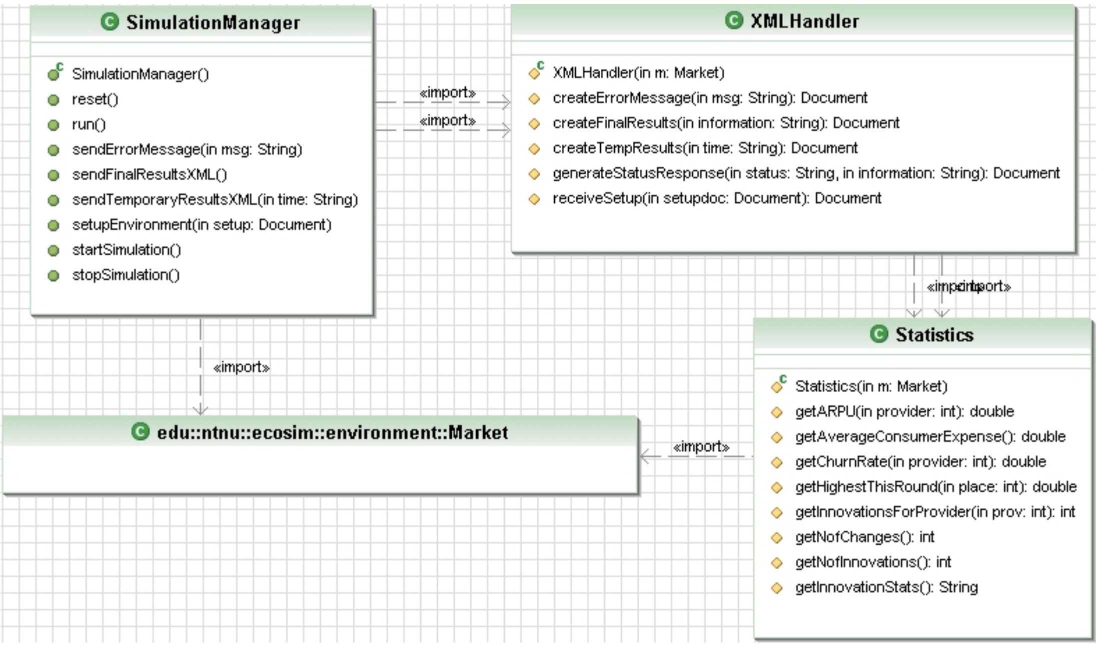

Figure F.2: Class Diagram: Management Layer

F.3 Economy Simulation Layer

The class diagrams of the *economy simulation layer* is divided into specific diagrams for the *consumer agents* (Figure F.3), the *service provider agents* (FigureF.4), and the remaining functionality (Figure F.5).

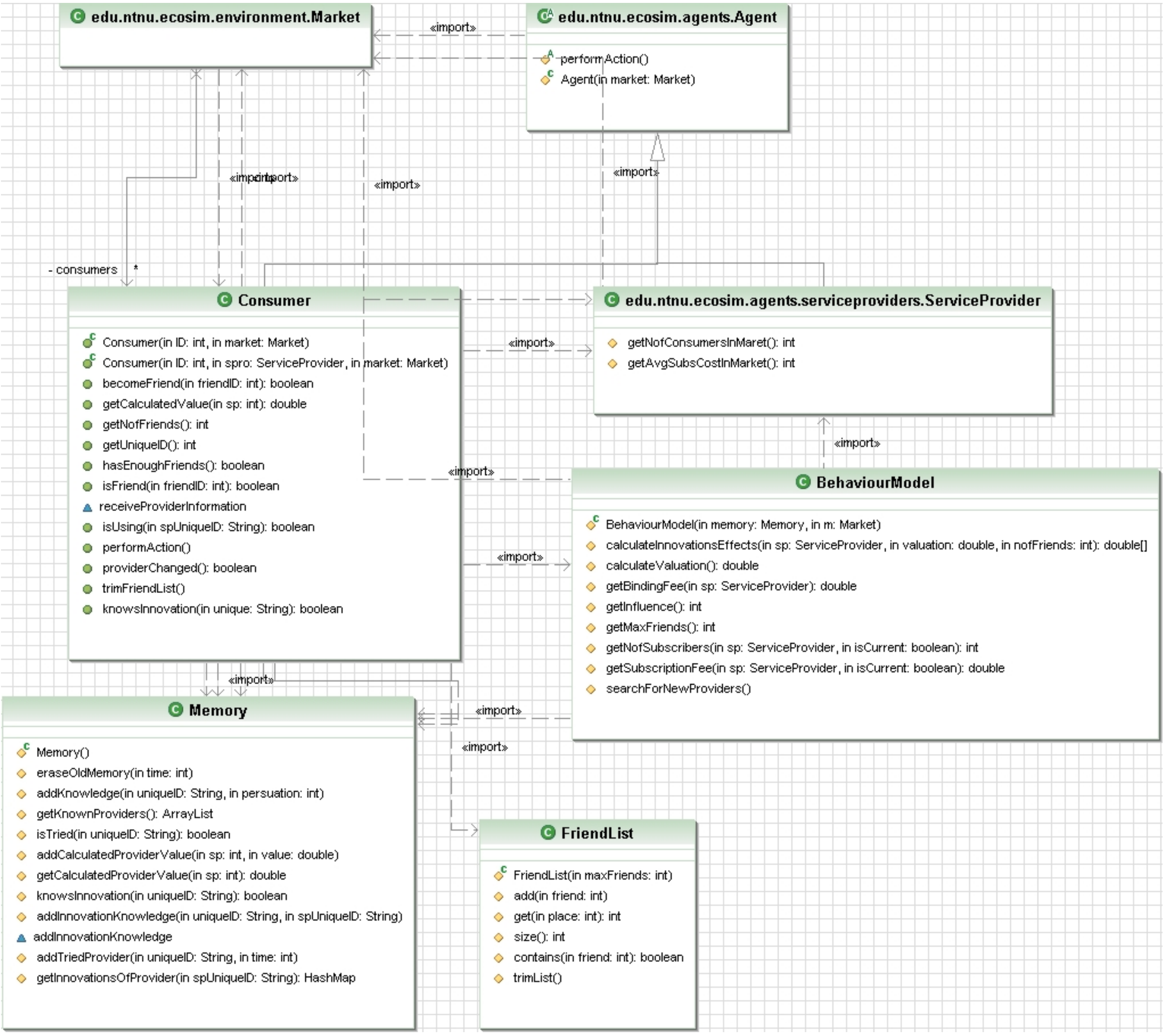

Figure F.3: Class Diagram: Consumer Agents

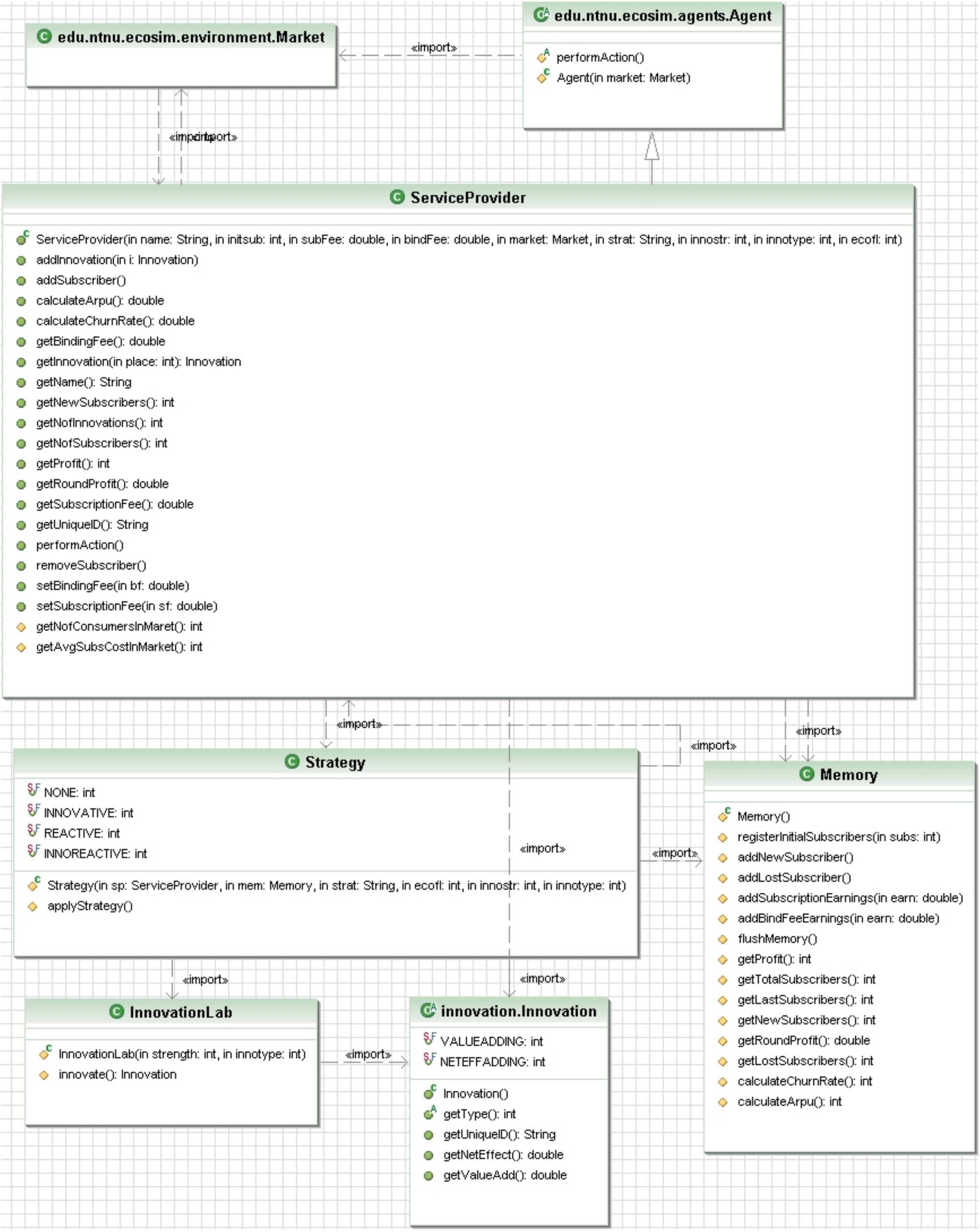

Figure F.4: Class Diagram: Service Provider Agents

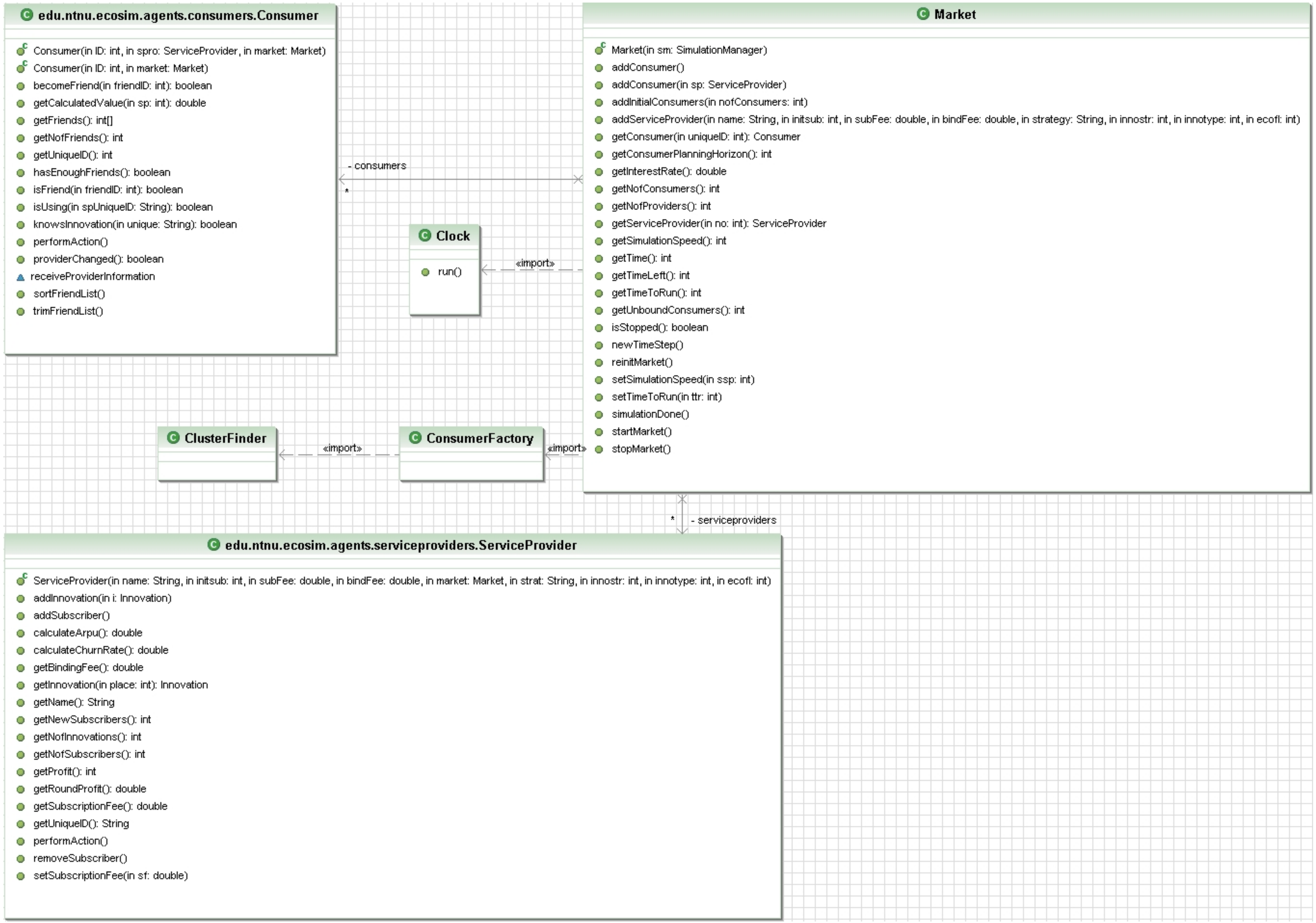

Figure F.5: Class Diagram: Market

F.4 Basic Functionality Layer

Figure F.6 shows the class diagram for the *basic functionality layer*.

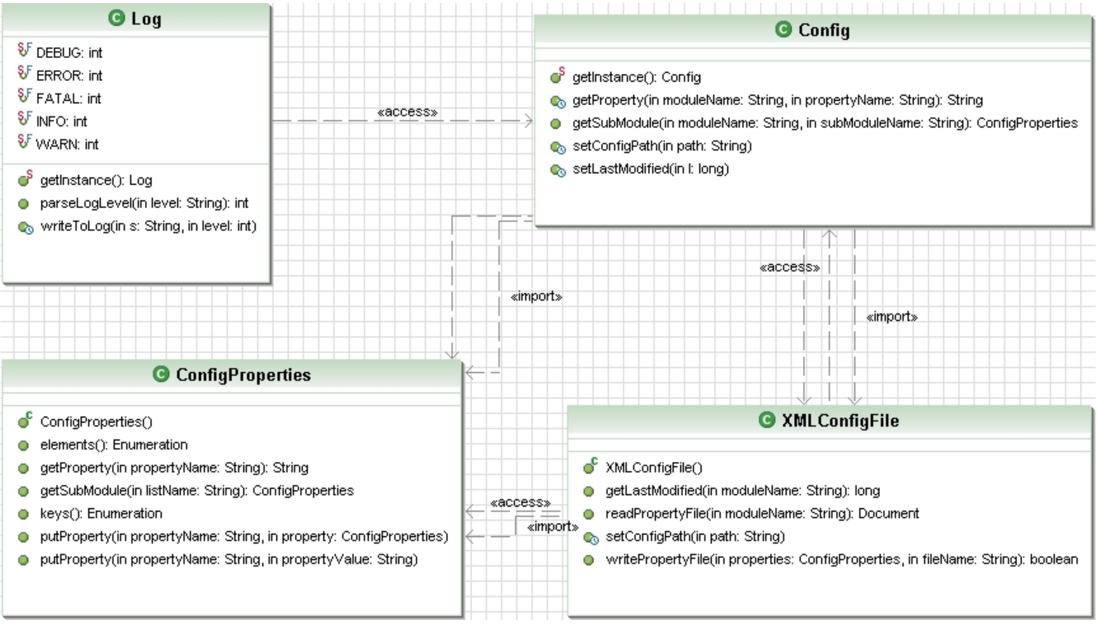

Figure F.6: Class Diagram: Basic Functionality

Appendix G

Statistical Results from Testing

Table G.1 shows the results from testing of the degrees of separation in the created social networks.

Table G.1: Results from testing of social networks: degrees of separation

Path length	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**	**Run 7**	**Run 8**	**Run 9**	**Run 10**	**Average**
0	1	1	1	1	1	1	1	1	1	1	**1**
1	17	11	10	5	15	8	12	6	9	7	**10**
2	211	104	129	53	164	87	125	48	108	101	**113**
3	2249	1070	1395	572	1697	903	1353	527	1129	1051	**1194.7**
4	21561	10769	13744	6112	16539	9476	13301	5402	11541	10933	**11937.8**
5	68956	60669	65926	45671	68455	57592	65700	41911	62480	61444	**59880.4**
6	7005	27367	18795	47361	13129	31911	19506	51658	24742	26456	**26791.2**
7	0	9	0	225	0	22	1	447	8	7	**71.9**

Appendix H

Algorithms

H.1 Social Network Creation

There are basically two ways to represent a social network: 1)through a "knowledge matrix", and 2)through node-links.

A knowledge matrix is a NxN matrix where N is the number of nodes (in this case consumers) that are represented. In the matrix two people knowing each other can be represented by marking the intersection in some way. However, when prototyping this solution it became obvious that the memory consumption of such a matrix would be to large for an ordinary computer to manage when the number of consumers exceeded about 50.000.

It therefore became clear that the consumers themselves needed to maintain a list containing relations to other consumers. However, a relation is a two-way property meaning that if X knows Y, then Y should also know X. To create such relationships for all consumers in the market, the algorithm of Figure H.1 was created.

H.2 Assignment of Friends

The number of friends each consumer is "allowed" to have is highly important. Real social networks have been shown to have a skewed degree distribution of friends/connections. As mentioned in this report, the social networks created by the simulator produce such distributions. Figure H.2 shows the algorithm for assignment of friends for a new consumer. The method `nextGaussian(Mean, Standard Deviation, Max, Min)` returns a stochastic draw from a gaussian function with the specified mean and standard deviation. It also assures that the returned value is not above Max or below Min.

```
ClusteringFactor cf;
createSocialConnections(){
 for(all consumers C in market){
  Friends F;
  while(!C.hasEnoughFriends){
   Consumer C2 = random consumer;
   if(!C2.hasEnoughFriends&&!C2.isFriend(C)){
    Friends.add(C2);
    C.becomeFriend(C2);
    C2.becomeFriend(C);
    for(all consumers CN in Friends){
     if(random number<cf&&!CN.hasEnoughFriends&&!CN.isFriend(C2)){
       CN.becomeFriend(C2);
       C2.becomeFriend(CN);
     }
    }
   }
  }
 }
}
```

Figure H.1: Social Networks Algorithm

```
calculateMaxFriends(Max, Min){
 steps = (Max-Min)/5;
 switch(User Type){
 case INNOVATOR:
  return nextGaussian(Min+(steps*4.5), (double)steps/2, Max, Min);
 case EARLY ADOPTER:
  return nextGaussian(Min+(steps*3.5), steps, Max, Min);
 case EARLY MAJORITY:
  return nextGaussian(Min+(steps*2.5), steps, Max, Min);
 case LATE MAJORITY:
  return nextGaussian(Min+(steps*1.5), steps, Max, Min);
 case LAGGARD:
  return nextGaussian(Min+((double)steps*1), steps, Max, Min);
 }
}
```

Figure H.2: Algorithm for assignment of friends

H.3 Innovation Development

As explained earlier in this report, modern research suggests that innovation development consists of a *chaotic* exploration phase, followed by a *sequential* exploitation phase. To be classified as chaotic, a system must:

- Be sensitive to initial conditions,
- Be topologically mixing, and
- Have periodic orbits that are dense

A common equation used in chaos theory is $X_{n+1} = a * X_n * (1 - X_n)$. This equation will behave almost like a stochastic process when the number a is above 3, but is actually fully deterministic.

Figure H.3 shows the algorithm that is used for the *exploration* phase of innovation in the simulations. This algorithm gives an advantage according to how innovative the companiy is.

```
innovationStrength; (between 0.1 and 0.5)
x; (initiated as random number between 0 and 1)

explore(){
 x = 3.95*x*(1-x);
 if(x<innovationStrength){
  finished exploring;
  }
}
```

Figure H.3: Algorithm for chaotic exploration

When the requirement for a finished exploration phase is fulfilled, the *exploitation* phase starts. This is described as sequential. Since the companies are unequally innovative, the time used for exploitation should also differ. Figure H.4 shows the algorithm used for this phase of the innovation development

```
exploitation; (initialized as 50)
innovationStrength; (between 1 and 10)

exploite(){
 if(exploitation<2){
  create innovation;
 }
 else{
  factor = innovationStrength + random number;
  exploitation = exploitation/factor;
 }
}
```

Figure H.4: Algorithm for sequential exploitation

Appendix I

Content of the Attached CD

A CD should be attached to this report, containing the parts of the thesis that were unnatural to include directly in the report. The CD contains the following folders:

javadoc This folder contains the Javadoc API of the simulator. The API is opened through the *index.html* file.

runnable This folder contains the runnable program. The folder must be copied to a directory on the local computer to run as supposed. For instructions on how to setup and start the simulator, see Appendix J.

source This folder contains the source code of the simulator (Java files).

report This folder contains this report (*report.pdf*), and the bibliography file used in the LaTeX document (*references.bib*).

Appendix J

Using the Simulator

This Appendix is intended as a users manual for the simulator. It describes how to configure and execute simulations through the user interface, as well as how the simulator can be accessed through the XML interface.

J.1 The Configuration File

Whether the simulator is accessed through the user interface or the XML interface, it will use the configuration file. This file is meant for settings that are likely to be the same for several scenarios. The existing settings are meant as a guideline for reasonable values. However, they may be changed according to the specifications given in this section.

The structure of the configuration file (named `ecosim.xml`) is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<module name="ecosim">
 <submodule name="Consumers">
  <property name="MaxFriends" value="20"/>
  <property name="MinFriends" value="5"/>
  <property name="ClusteringFactor" value="0.45"/>
  <property name="InnovatorInfluence" value="50"/>
  <property name="EarlyAdopterInfluence" value="40"/>
  <property name="EarlyMajorityInfluence" value="30"/>
  <property name="LateMajorityInfluence" value="20"/>
  <property name="LaggardInfluence" value="10"/>
  <property name="ValueNoiseFactor" value="0.03"/>
  <property name="useSubscriberNoise" value="false"/>
  <property name="PlanningHorizon" value="12"/>
  <property name="MemoryTime" value="6"/>
  <property name="NetEffIsLocal" value="true"/>
  <property name="ProviderSearchInterval" value="3"/>
```

```
  <property name="ProviderInfoGlobal" value="false"/>
 </submodule>
 <submodule name="Market">
  <property name="InterestRate" value="5"/>
 </submodule>
 <submodule name="Log">
  <property name="LogPath" value="current"/>
  <property name="LogLevel" value="DEBUG"/>
 </submodule>
 <submodule name="Innovation">
  <property name="HighestValueAddPct" value="10"/>
 </submodule>
 <submodule name="Debugging">
  <property name="AnalyzeSocialNetworks" value="false"/>
 </submodule>
</module>
```

The following are descriptions of the fields in the configuration file:

MaxFriends: the maximum number of friends allowed for the consumers (must be > 0)

MinFriends: the minimum number of friends allowed for the consumers (must be > 0 and $<$MaxFriends)

ClusteringFactor: the probability that two consumers having a friend in common are friends themselves (between 0 and 1, i.e. 0.45=45%)

InnovatorInfluence: the influence innovators have on their friends (between 0 and 100)

EarlyAdopterInfluence: the influence early adopters have on their friends (between 0 and 100)

EarlyMajorityInfluence: the influence early majority consumers have on their friends (between 0 and 100)

LateMajorityInfluence: the influence late majority consumers have on their friends (between 0 and 100)

LaggardInfluence: the influence laggards have on their friends (between 0 and 100)

ValueNoiseFactor: the noise to add to the consumers valuation of services (between 0 and 1)

useSubscriberNoise: whether or not to add noise to the consumers perception of how many subscribers each provider has (true/false)

PlanningHorizon: how many months the consumers should consider when evaluating service providers (must be > 0)

MemoryTime: how many months the consumers shall remember their decisions. In this period, service providers that the consumer has used before are ignored when considering new options.

NetEffIsLocal: whether or not network effects should be calculated based on the consumers social connections instead of the providers installed base (true/false)

ProviderSearchInterval: how often the consumers shall look for new service providers in the market (e.g. how many months there are between each time the consumer search for possible providers himself)

ProviderInfoGlobal: whether information on new providers is common information (global), or this information should spread through social networks (true/false)

InterestRate: the interest rate in the market (percentage, between 0 and 100)

LogPath: the path where the log should be written . Can be a file path (for instance "D://EcoSimFiles//logfiles"), or "current"

LogLevel: the lowest level the log should consider (DEBUG/INFO/WARN/ERROR/FATAL)

HighestValueAddPct: the highest allowed percentage a "value-adding" service can add to the subscriptions value (between 0 and 100)

AnalyzeSocialNetworks: whether or not the social networks should be analyzed after creation for debugging purposes (true/false)

J.2 The User Interface

A simple user interface was created to allow for simulations without further development. The attached CD contains a directory with the files necessary to run the simulator with the user interface. To run the simulator, the computer should follow these requirements:
1500 MHz Pentium class processor or better
1GB RAM
Java version 6 or newer
Screen resolution: 1200x870 or higher

The directory containing the files must be copied to a folder on the local hard drive. The user interface is started by double-clicking the `run.bat` file.

The user interface consist of five tabs, as explained in Chapter 7. This section describes the options in the "Service providers" and "Environment Variables" tabs.

J.2.1 Service Provider Configuration

The following options are enabled for service provider configuration (see also Figure 7.4):

Name: The name of the service provider

Initial Subscribers: The initial number of subscribers for this service provider

Binding Fee: The binding fee (the price for changing to this provider). This may change during time if a "reactive" strategy is chosen.

Subscription Fee: The monthly fee for subscribing to this service provider

Strategy: Here you will be presented to four choices:
- None: No strategy is selected, the service provider will neither innovate nor react to the market
- Innovative: The service provider will innovate with new services
- Reactive: The service provider will react to changes in the market through pricing strategies
- Both: The service provider will both innovate and react to market changes

Innovative Strength: The innovative strength will affect how often the service provider innovates. The "Average time between innovations" field will show an approximation of how often the innovations will be launched with the selected strength.

Type of innovativeness: As explained in the report, services may either have intrinsic value *or* produce network effects. The further to the right this slider is drawn, the larger percentage of the innovations have network effects.

Economic flexibility: The economic flexibility is related to the "Reactive" strategy. The higher economic flexibility, the more the company is capable of lowering its prices. The "maximal price decrease each month" field will show how many percentages the selected value will enable the service provider to decrease its prices with.

Allow predator behavior: This selection will only be enabled when the service provider has the highest degree of economic flexibility. When this option is selected, the company will not settle with any other option than 80% of the market share. If not selected, the companies will settle once the market is stable.

Add Provider: Add the configured provider to the list of providers.

J.2.2 Environment Configuration

The following options are enabled for environment configuration (see also Figure 7.5):

Initial population: The number of unbound consumers that exist in the market (consumers without subscriptions)

Simulation length: The number of months to be simulated

Number of runs: The number of simulations that should be done for this specific scenario (if more than one, all simulations will be run sequentially)

Simulation speed: How fast the simulation should run. If there are more than 10.000 consumers in the simulation in total, this variable will not have much affection on the speed.

J.2.3 Starting, Stopping, and Resetting the Simulator

Once the simulator is configured as wanted, it may be started by pressing the "Start" button. It will then take some time before the simulator responds, since all consumers and social networks need to be constructed. Once it starts, the output will be shown in the statistics tabs and the "Stop" button will be enabled.

If the user wants to stop the simulation, he or she simply press the "Stop" button and the simulation will be canceled.

The "Reset" button will be enabled when the simulation is stopped or finished. This will prepare the simulator for new simulations with different configurations.

J.2.4 Output

The output of each simulation will be shown in the three remaining tabs. However, the output will also be written as images **and** textual output in the "output" folder that will be created in the same directory the simulator is run from. This folder will contain sub folders for each simulation (when several simulations are run for the same scenario (Number of runs>1) all the outputs from these simulations will be written to the same folder).

Further, the log file may contain valuable information for each simulation. Amongst others, the social network statistics and innovation details will be written to this file as `INFO` messages.

J.3 The XML Interface

As mentioned earlier in this report, the simulator is accessible through exchange of XML messages. This will enable several applications to use the simulator and interpret the output in different ways. For instance, an application may be created that writes the results to a database. This will ease analysis of the results through, for instance, regression analysis.

To communicate with the simulator, the following steps need to be followed:

1. Create a class that extends `edu.ntnu.ecosim.communication.ResultReceiver` and implement the necessary methods
2. Instantiate an instance of the `edu.ntnu.ecosim.communication.EcoSim` class. The constructor to use is `EcoSim(ResultReceiver rr, String configpath)` where `rr` is the class you created in step 1. The "configpath" is the **absolute file path** to the location of the configuration file (`ecosim.xml`). Even if the file is located in the same directory as EcoSim.jar, the complete file path must be included.

This will give you access to the communication interface of EcoSim (the working name of the simulation engine). To communicate with the simulator, the following methods may be used:

setupSim(Document setup) This method is used to send the setup document (explained later) to the simulator

startSim() This method will start the simulation (after a confirmation about successful setup is received)

stopSim() This method will stop the current simulation and create the final results

resetSim() This method will reset the simulator, preparing it for a new simulation

Figure J.1 shows how these methods can be used to controll the simulator. The setup document must be sent to the simulator to prepare for simulation. The `ResultReceiver` will receive a confirmation when the setup is finished, and the simulator may then be started. This may take a while, since all agents and social networks are created during this time.

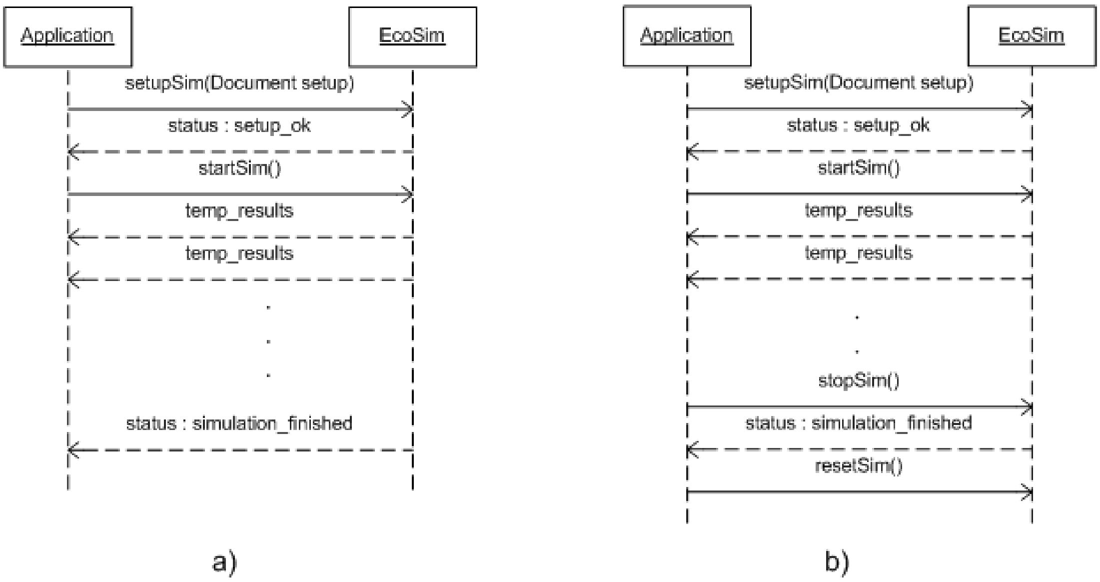

Figure J.1: Usage of the communication interface: **a)** a simulation that is finished naturally, and **b)** a simulation that is stopped during execution.

J.3.1 The setup document

The setup document should look as follows:

```
<setup>
 <environment>
  <unboundpopulation/>
  <simulationspeed/>
  <simulationlength/>
 </environment>
 <provider>
  <name/>
  <subscribers/>
  <bindfee/>
  <subscost/>
  <strategy/>
  <innostr/>
  <ecoflex/>
 </provider>
 .
 .
</setup>
```

The document must contain **one** environment, and **one or more** providers. The fields of the document should be interpreted as follows:

unboundpopulation This is the number of *unbound* consumers in the market (without subscription). Must be an integer between 0 and N.

simulationspeed An integer between 1 and 100, defining the speed of the simulation (100 is the fastest).

simulationlength The number of months to be simulated. Must be an integer higher than 1.

name The name of the service provider.

subscribers The number of initial subscribers this service provider has. Must be integer between 0 and N.

bindfee The initial binding fee of this service provider. Must be integer between 0 and N.

subscost The initial monthly subscription cost of this service provider. Must be integer between 0 and N.

strategy The strategy of this service provider. Must be either "NONE", "INNOVATIVE","REACTIVE" or "INNOREACTIVE".

innostr The innovative strength of this service provider. The parameter is only used when strategy is *INNOVATIVE* or *INNOREACTIVE*. Must be integer between 0 and 10, where 10 is the highest innovative strength.

ecoflex The economic flexibility of the service provider. The parameter is only used when strategy is *REACTIVE* or *INNOREACTIVE*. Must be integer between 0 and 10, where 10 is the highest economic flexibility. However, if the parameter is set above 10, the behavior will be *predatory*, meaning that the provider will try to capture more than 80% of the market.

J.3.2 Response Messages

There are two kinds of messages that are sent back to the `ResultReceiver`: status messages and temporary results.

J.3.2.1 Status Messages

The status messages have the following content:

```
<response>
 <type>status</type>
 <status>....</status>
 <information>.....</information>
</response>
```

The `<status>` field may contain either "`info`","`setup_ok`", "`setup_failed`", "`error`", or "`simulation_finished`".

The info message is used for general purposes when the simulation engine needs to report some information to the using application. For instance, during initialization of the simulator the info messages are used to give feedback on the progress.

The setup_ok message is an indication that the setup document was parsed correctly and that the simulator is ready to be started. This is the expected response after the setupSim(Document setup) method is used.

If a setup_failed message is received, the simulator should not be started as there is most likely some errors in the setup document. The `<information>` field will give information about the cause of failure.

If an error message is received, the information field should be further analyzed. This may be an exception thrown within the simulator that may affect the result, or it can be a more important error.

The simulation_finished message will be sent when the simulation is finished. This message will contain some textual information about the simulation in the information field.

J.3.2.2 Temporary Results

The temporary results messages are sent for each finished time step the simulator has performed. The messages have the following content:

```
<response>
 <type>temp_results</type>
 <time>...</time>
 <totalconsumers>...</totalconsumers>
 <totaltransitions>...</totaltransitions>
 <avgconsumerexpense>...</avgconsumerexpense>
 <unboundconsumers>...</unboundconsumers>
 <provider>
  <name>...</name>
  <customers>...</customers>
  <cost>...</cost>
  <subfee>...</subf>
  <highestvalue>...</highestvalue>
  <totalrevenue>...</totalrevenue>
  <arpu>...</arpu>
  <churnrate>...</churnrate>
  <innovations>...</innovations>
 </provider>
 .
 .
</response>
```

The fields of the document can be interpreted as follows:

time: The time for these results

totalconsumers: The total number of consumers in this market

totaltransitions: The total number of transitions made by the consumers since the simulation started

avgconsumerexpense: The average expense of each consumer at this time

unboundconsumers: The number of consumers without a subscription

name: The name of the service provider

customers: The number of subscribers for this service provider

cost: The binding fee at this time for the service provider

subfee: The subscription fee at this time for the service provider

highestvalue: The highest value/benefit any consumer has valuated this service provider to have

totalrevenue: The total revenue of the service provider

arpu: The ARPU after this time step for the service provider

churnrate: The customer churn rate at this time for the service provider

innovations: The total number of innovations launched by this service provider