Value-based Requirements Engineering

Exploring Innovative e-Commerce Ideas

Jaap Gordijn

VRIJE UNIVERSITEIT

Value-based Requirements Engineering: Exploring Innovative e-Commerce Ideas

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op dinsdag 25 juni 2002 om 15.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Jacob Gordijn

geboren te Soest

promotor: prof.dr. J.M. Akkermans

SIKS Dissertation Series No. 2002-8.

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Graduate School for Information and Knowledge Systems.

Promotiecommissie:
prof.dr. J. M. Akkermans (promotor)
dr. N. M. Sadeh (Carnegie Mellon University)
J. Sifonis (Managing Director Internet Business Solution Group, Cisco Systems)
prof.dr. R. J. Wieringa (Universiteit Twente)
prof.dr. M. R. Creemers (Vrije Universiteit Amsterdam / Universiteit Nyenrode)
prof.dr. Y. H. Tan (Vrije Universiteit Amsterdam)
prof.dr. J. C. van Vliet (Vrije Universiteit Amsterdam)

"Pappa, nou begrijp ik wat je doet: je tekent doolhoven." [1]
(Anne Gordijn, 7 years old)

[1] *"Daddy, now I understand what you're doing: drawing labyrinths."*

Preface

The construction of an information system is often compared to the construction of a building. Although this analogy is not in all cases appropriate, doing research is somewhat similar to building construction also. As such, I have some empirical evidence, because over the past few years, the two main projects for me have been a substantial house reconstruction project and a research track.

During my home reconstruction I have been advised by many people. For my research track, my promotor Hans Akkermans played the role of advisor and mentor. Hans has the gift to be inspiring and motivating on the right moments, while leaving sufficient room and freedom to explore my own ideas and directions. Moreover, I enjoyed especially our discussions on how to do research in an applied field as information science is.

House reconstruction may require some capital injections to deliver the final results. In the context of my research, Edwin Paalvast and Maarten van Steen provided intellectual and spiritual capital. Edwin was always very keen on getting me on the right projects at Bakkenist Management Consultants, and by doing so contributed largely to the Action Research like approach followed in this thesis. Also, his feeling for business development is fabulous and directed me more than once towards the more innovative e-commerce ideas. Maarten has always been a motivator for me and introduced me in the arena of academic research and education. In addition he was a tremendous help in starting this research some years ago.

In my direct neighborhood, I am not the only person rebuilding attics, pulling down walls and alike. Many others take similar actions. These fellow-sufferers also exist in research tracks. First I would like to acknowledge my roommate Nico Lassing for his willingness to listen to and react on new ideas and for sharing his LaTeX knowledge with me. The past four years, he was good company to share a room with. In addition, I want to salute my other (ex) colleges, with whom lunch is an everyday debate on various topics: Arno Bakker, Gerco Ballintijn, Mirna Bognar, Michel Klein, Frank Niessink, Michel Oey, Bastiaan Schönhage, Chris Verhoef,

and Martijn van Welie. Moreover, thanks go to Hans de Bruin and Hans van Vliet for many fruitful discussions.

Finally, a long-term home reconstruction has a major impact on direct relatives. The same holds for a research project. As such I would like to thank my parents, but above all, I am very grateful to Erna and my kids Anne and Femke for their patience, love and support. In a while, I will start to work on Femke's writing desk, the dining room, . . .

Jaap Gordijn, Soest, March 2002

Contents

Chapter 1

Introduction

Over the past few years, many innovative e-commerce ideas have been considered. Such ideas reveal new value propositions, which are enabled by new technological possibilities, such as the widespread use of the Internet and technologies on top of it.

During 1998-1999, the e-commerce hype reached its top. Recently, it became clear that many e-commerce ideas are not successful (Shama 2001). Many enterprises doing e-commerce have not been able to create profit with their e-commerce ideas. Some of these companies who relied entirely on future e-commerce profits have gone bankrupt.

An important reason for the failure of e-commerce ideas is the lack of a sound value proposition to customers. Moreover, many ideas did not contribute sufficiently to profitability of enterprises. Rather, many enterprises focused on maximizing market share and establishing a trusted brand name.

However, we still believe that many potential successful e-commerce ideas exist, which utilize enabling Internet related technical innovations in a profitable way. Moreover, some industries are forced to find new value propositions. For instance, the digital content industry is facing challenges with respect to new value propositions utilizing Internet technology, e.g. how to earn money by streaming music to an end-consumer's device.

A challenge in putting e-commerce ideas into operation, in addition to satisfying a profitability requirement, is that business *and* technology closely interwork. This greatly expands the e-commerce 'design space'. A new technological feature enables more than one business idea, while new business ideas are only possible if technological constraints are satisfied. This close interaction between on the one hand designing a sound value proposition and on the other hand designing an information system enabling this proposition is very typical for e-commerce projects, and results in more than only an information system or business design problem. Moreover, innovative e-commerce ideas tend to be formulated very vaguely initially. Such an idea is a statement about a combination of an innovative value proposition utilizing a new technological possibility, but it often lacks a precise description. As a result, many innovative e-commerce ideas are somewhat unfocused and inaccurate. This makes it different to put the idea into operation, and to develop a supporting information system. What is needed is an in-depth exploration process of an e-commerce idea, to understand the idea better as well as to formulate it more precisely, and to focus the idea into a direction that is feasible from an economical and technical perspective.

This thesis discusses how such an innovative e-commerce idea can be explored taking into account business and technological perspectives. Our *e^3-value* approach to do so is on the one hand based on the analysis of *economic value* creation, distribution, and consumption in a multi-actor network. On the other hand, *e^3-value* is founded on *requirements engineering* and underlying *conceptual modeling* techniques, borrowed from the information systems community. Requirements engineering is the process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained (Loucopoulos & Karakostas 1995).

In this thesis, we focus on the use of a requirements engineering and conceptual modeling approach to articulate, analyze and validate a value proposition more thoroughly. One of the observations we have made during e-commerce idea exploration tracks (see also section 2.2) is that initially these tracks are about finding an Internet enabled value proposition. Therefore, in this thesis much attention is paid to finding, representing, analyzing and evaluating such a value proposition. We describe a value proposition using a conceptual value model that shows how actors create, distribute, and consume objects of *economic value*. The motivation to use a more formal, conceptual approach for exploring a value proposition is twofold. First, modeling a value proposition explicitly, may contribute to a common understanding of the proposition by all stakeholders involved. While doing so, special attention should be paid to stakeholders with an information technology interest. Such stakeholders should understand the proposition well, because they have to design and implement an information system that puts the proposition into operation. Moreover, these stakeholders often have in-depth technological knowledge and thus can provide valuable input for designing an Internet enabled value proposition. Because this stakeholder group is used to more formal conceptual models, we hope that a conceptual value proposition model may help their understanding of the e-commerce idea. Second, a more formal model of the e-commerce idea allows for evaluation of the idea, which is in our case biased towards assessment of potential profitability of the idea.

In addition, e^3-*value* is a lightweight approach. In our experience, e-commerce idea exploration tracks typically may take a few weeks to a month, so an approach supporting idea exploration should facilitate a relatively short exploration track.

1.1 Research context

The research described in this thesis was done in two different environments: (1) a business environment, and (2) an academic environment.

Business environment. During the time the research was conducted, I worked as a consultant for Deloitte & Touche Bakkenist / e-commerce department, and also for Cisco's Internet Business Solution Group (IBSG). Part of the work was to guide customers in e-commerce exploration tracks. The projects outlined in chapter 7, 8, and 9 are all examples of such tracks. I used these projects to construct and validate my e^3-*value* method for innovative e-commerce idea exploration.

Academic environment. Also, I have been working for the Vrije Universiteit / Amsterdam on the STW project *A Framework for the Electronic Sale of Information Products* (VWI 4949). The aim of this project is to develop a business and technical framework for the sale of information products. An information product is everything that can be represented by bits, and which can be sold. Therefore, I often refer to *digital* products rather than to *information* products. Examples are music, movies, news, and software. My work focuses on the business framework, and more specifically on how to explore and represent such a framework using requirements engineering and conceptual modeling techniques. The resulting e^3-*value* approach is based on projects and studies performed as a consultant, and on the reflection on these projects in a more academic setting.

1.2 Research question

We began our research when the Internet-enabled e-commerce hype was just taking off (about five years ago). The research question at that time was:

- How to develop an e-commerce intensive information system?

From innovative e-commerce exploration projects we carried out, we concluded that new Internet enabled e-commerce ideas with which development tracks start with are rather unclear and unfocussed (see Gordijn & van Vliet (1999)). Moreover, it is very doubtful if many of these ideas are likely to be profitable. Thus, developing an e-commerce intensive information system is not so much the problem, but doing so *in combination* with a new value proposition. As a result, we formulate a different research question than the one we started with:

- How can we precisely define an innovative e-commerce idea such that it is clear to all stakeholders and allows for profitability evaluation?

We have observed a number of innovative e-commerce exploration tracks issues, which are related to this question (see also section 2.2 for a more detailed discussion):

1. *Information technology knowledge is key to many e-commerce ideas*. Typical innovative e-commerce ideas exploit new technological possibilities to create advantage. How to account properly for technological knowledge needed during e-commerce exploration tracks?

2. *A wide range of stakeholders, ranging from CxO's to information technology concerned persons is involved*. How to deal with the wide range of interests of various stakeholder groups involved during e-commerce idea exploration? How to ensure that discussions include different perspectives but maintain focus?

3. *Many e-commerce ideas are described only vaguely, thereby leaving room for multiple interpretations*. How to represent an e-commerce idea in a more unambiguous way, and such that still all stakeholders reasonably can understand the idea?

4. *Idea exploration may take only a limited period of time, typically a few weeks*. How to explore an e-commerce idea thoroughly, while having a time constraint for doing so imposed?

5. *A focused and unambiguous e-commerce idea should also be feasible*. How to assess an e-commerce idea from a profitability (economic feasibility) and technological feasibility perspective?

6. *Finding an innovative e-commerce idea itself is difficult*. Finding an innovative e-commerce idea is a very creative task by nature. How can such a creative process be facilitated?

   In this thesis, we address issues 1-5. With respect to finding an innovative e-commerce idea, we explicitly acknowledge the creative nature of this task and assume that such an idea exists. As a consequence, our e^3-*value* approach can not be used to find e-commerce ideas themselves.

1.3 Research design

The research approach that we follow in this thesis comes close to Action Research. Action Research is an iterative research process involving researchers and practitioners acting together in a particular cycle of activities, including problem diagnoses, action intervention, and reflective learning. A particular strength of methods like Action Research is their value in explaining what goes on in organizations (Avison et al. 1999). As innovative e-commerce idea exploration is an (inter-) organizational process, Action Research is a way to shed light on such a process. Moreover, Action Research is well suited to address problems that are not well defined and ill-structured. E-commerce idea exploration is a typical example of such a problem.

Action Research is a useful approach to associate research with practice and visa versa, because it is about taking action (e.g. participating in client projects), and it is about reflection (analyzing and learning from action, as well as enhancing a next line of action using lessons.) As explained in section 1.1, we have been working both in a consultancy and academic environment, so Action Research seems to be an appropriate strategy for our research context. In the consultancy environment, the action part of Action Research was carried out, while the academic environment was used for reflection.

The aim of Action Research is not to develop a general acceptable theory, but more a theory which is appropriate for a specific case. Our ambition is to develop a more general theory on how to do innovative e-commerce idea exploration. To construct such a theory, we have also been influenced by Grounded Theory (see e.g. Glaser & Strauss (1967)). The Grounded Theory approach constructs a theory consisting of categories, properties of these, and linkages between these categories based on data collection in the field. Grounded theorists propose interviews to do so, but we have used a more participatory approach for this. Also, we have been influenced by a variety of literature on organizational science and requirements engineering.

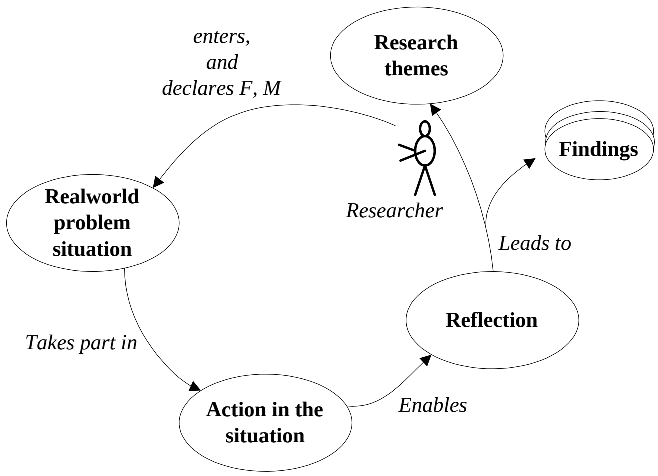


Figure 1.1: The cycle of Action Research, according to Checkland & Holwell (1995).

We outline our research process using the Action Research cycle of Checkland (1991) (see figure 1.1). According to him, Action Research consists of a number of cyclic steps. First, a researcher identifies a research theme and seeks a real-world situation *A* in which the research theme is expected to play a role. Also, the researcher declares a framework *F* and a methodology *M*, which are used by researcher if s/he actively participates in the real-world situation. At some point in time, the researcher leaves the real-world situation, and reflects on the gained experience and records the learned in relation to *F*, *M*, and *A*. Experiences can even lead to a shift in a research theme.

Shift in research theme. Our initial research theme was the design of an e-commerce intensive information system. As a result of our first participation in an explorative e-commerce project (Gordijn & van Vliet 1999), we learned that some e-commerce ideas are highly innovative. For such ideas, knowledge on exploration, articulation and evaluation of the idea, and more specifically the value proposition in close relation to the information systems implementing such a proposition, is hardly available. So, this led to a shift in our research theme towards innovative e-commerce idea exploration from multiple perspectives. We focus on the question how to design, represent, and evaluate a value proposition more thoroughly, so that all stakeholders can understand and agree on it, and the value proposition can be evaluated.

A framework F for e-commerce idea exploration. We used the Action Research cycle also to construct a framework on e-commerce idea exploration. We call this framework an ontology, because it consists of a number of generic concepts, relations between these concepts, and rules on which a group of stakeholders reasonably can agree. The e^3-*value* ontology is explained in chapter 3. To construct this ontology, we used a number of Action Research cycles. Each cycle resulted in a change or enrichment of the ontology. The role Action Research plays in our e^3-*value* construction process, is explained more in depth in section 3.6.

A methodology M for e-commerce idea exploration. Our e^3-*value* ontology provides a framework for representing essentials of an e-commerce idea, but does not say much on how to explore such an idea from a more methodological point of view. So, the rest of the projects we carried out focused on how to put the e^3-*value* ontology into practice. This has lead to a global e-commerce idea exploration process (see chapter 5), and to guidelines how to evaluate an e-commerce idea. This evaluation process is driven by an economic value assessment from a profitability and consumer value perspective (chapters 7 and 8). Moreover we have performed a study how exploration of an e-commerce idea from an economic perspective can be extended and enriched with early information system exploration (see chapter 9).

1.4 Contributions

How to develop an economic value perspective representing an initially vaguely articulated e-commerce idea more thoroughly, using principles and techniques which stem from requirements engineering and conceptual modeling, is the main achievement of this thesis. So far, requirements engineering has focused on exploring information system requirements and at best on organizational goals having implications for information systems (Yu & Mylopoulos 1998). We extend the use of requirements engineering and conceptual modeling approaches to the exploration, articulation and evaluation of innovative business cases, and thereby exploit the strengths of these approaches, being more formal and thorough. By doing so, e^3-*value* addresses a drawback of business science approaches which tend to yield imprecise representations of an e-commerce idea, leaving room for ambiguous interpretations. In brief, our approach contributes:

1. *an economic value-based ontology which precisely defines what a value model is;*
2. *a simple to use yet rigorous graphical technique to present a value model;*

3. *a scenario-based methodology to explore and analyze a value model;*

4. *a quantitative method to evaluate the feasibility of a value model;*

5. *a deconstruction and reconstruction approach to find variations on a baseline value model;*

6. *a validation of our methodology by a series of real-life projects.*

1.5 Structure of this thesis

This thesis is structured as follows:

- Chapter 2 provides an overview of value-based requirements engineering for e-commerce. We introduce the notions of innovative e-commerce and requirements engineering. Also, general observations made during exploration of innovative e-commerce ideas are presented, which are used to motivate the baseline of our e^3-*value* approach.

- Chapter 3 outlines an ontology plus scenario mechanism which can be used to capture e-commerce ideas from an *economic value* perspective.

- Chapter 4 explains the difference between process modeling and value modeling. Although ideas, which stem from process modeling can be found in our approach, value modeling is not equal to process modeling.

- Chapter 5 presents a global e-commerce idea exploration process, with a focus on the aforementioned economic value perspective. Also, practical guidelines are presented.

- Chapter 6 introduces value model deconstruction and reconstruction as a way to find variations on e-commerce ideas.

How e^3-*value* practically works out is presented in the following three chapters:

- Chapter 7 shows e^3-*value* from a consumer value perspective.

- Chapter 8 does the same for the profitability perspective.

- Chapter 9 illustrates that exploration of an economic value perspective goes hand in hand with the early exploration of other perspectives, such as a business process and information system perspective.

It is our experience that successful use of the e^3-*value* methodology depends to a certain extent on adequate tool support. Chapter 10 discusses existing and envisioned tool support for e^3-*value* . Finally, chapter 11 presents our conclusions and directions for further research.

1.6 Acknowledgements

I am grateful to the participants of STW project VWI 4949: Deloitte & Touche Bakkenist Management Consultants, Océ Research & Development, KPN Research, NOB Interactive, The Netherlands Audiovisual Archive (NAA), The Free Ads Papers International Association (FAPIA) and PersCombinatie (PCM). More specifically I would like to thank Gert-Jan Kruyt from FAPIA and Maarten Kleyn from PCM for their willingness to let me participate in their e-commerce exploration tracks. Also, I am grateful to the Stichting Technische Wetenschappen (STW) for sponsoring project VWI 4949.

Fortunately, I had the opportunity to see e-commerce from a business development perspective at Deloitte & Touche Bakkenist and Cisco Systems and their customers. I would especially like to thank Bert Hazelaar and Hans van Berkel from the Stichting Exploitatie NAburige rechten (SENA) for the opportunity to explore new ways of selling and distributing music.

Koen Schuitemaker should be acknowledged for his implementation of the e^3-*value* ontology in Prolog. Also, he made many constructive remarks on earlier versions of the e^3-*value* ontology.

STW project VWI 4949 is a project which is carried out jointly by the Computer Systems and IMSE department of the division of Exact Sciences of the Vrije Universiteit / Amsterdam. As such, I would like to thank Andy Tanenbaum, Chandana Gamage and Wilfred Dittmer for cooperating in this project. I am also grateful to Hans de Bruin for introducing Buhr's Use Case Maps (Buhr 1998) to me, to represent operational scenarios. Finally, I owe a special debt of gratitude to Hans van Vliet for the many comments and remarks he made.

1.7 Publications

Parts of the material presented in this thesis has been published before.

- Chapter 2 on value-based requirements engineering uses an exploratory study reported on in Gordijn & van Vliet (1999) (addendum to the European Software Engineering Conference / Foundations of Software Engineering). Also this chapter explains e-commerce requirements viewpoints, which were published in Gordijn et al. (2000*a*) (33rd Hawaii International Conference On System Sciences (HICSS-33)).

- Chapter 3 presents the e^3-*value* ontology for e-commerce value models, and is based on Gordijn et al. (2000*b*) (12th International Conference on Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, (EKAW 2000)).

- Chapter 5 explains the difference between an economic value oriented modeling approach and process modeling and is based on Gordijn et al. (2000*c*) (First International Workshop on Conceptual Modeling Approaches for e-Business).

- Chapter 6 discusses value model deconstruction and reconstruction, which was published at the First International Conference on Knowledge Capture (Gordijn & Akkermans 2001*c*).

- Chapter 7 presents value model evaluation from a consumer perspective, and is based on Gordijn et al. (2000*d*) (First International Conference on Electronic Commerce and Web Technologies (EC-Web 2000)).

- Chapter 8 exemplifies profitability assessment of an e-commerce idea, which was published as a journal article in IEEE Intelligent Systems (Gordijn & Akkermans 2001*a*), and as a proceedings paper on the Workshop on Intelligent e-Business (part of the First International Conference on Knowledge Capture) (Gordijn & Akkermans 2001*b*).

- Chapter 9 discusses multi-viewpoint requirements engineering for e-commerce and is also presented in Gordijn et al. (2001) and Gordijn et al. (1999) (34rd Hawaii International Conference On System Sciences (HICSS-34), and The First BeNeLux Conference on the State-of-the-Art of ICT (Information and Communication Technology) Architecture respectively).

Other dissemination

Value-based requirements engineering as presented in this thesis has also found its way to educational fora. Executive courses have been given as post academic education for PAO (Dutch: Post Academisch Onderwijs), and as master classes (e.g. for ASZ (Dutch: Automatisering Sociale Zekerheid)). Also, e^3-*value* has been tought on the SIKS (Graduate Research School for Information & Knowledge Systems) course on conceptual modeling. Classes have been given to consultancy parties (Deloitte & Touche Bakkenist as well as KMPG Worldclass IT). Finally, the thoughts in this thesis are used in academic courses on e-commerce at the Universität Klagenfurt, Fakultät für Wirtschaftswissenschaften und Informatik, the Institutt for Datateknikk og Informasjonsvitenskap, Trondheim, Norway, and the Vrije Universiteit Amsterdam / Faculty of Sciences. Furthermore, e^3-*value* will be used as the foundation for several European projects (see Obelix consortium (2001), BusMod consortium (2001)) that start in early 2002.

Chapter 2

Value-based requirements engineering

Value-based requirements engineering exploits the concept of *economic value* during the requirements engineering process and is especially useful when doing requirements engineering for innovative e-commerce information systems. Such systems have in common that they presuppose a *new*, hardly understood, e-commerce idea with which actors potentially can make a profit, or when put into operation, produces something of economic value for actors. In fact, a new and profitable information technology intensive *value proposition* has to be invented, and stakeholders have to agree on this.

In this thesis, we concentrate on the very first phase of developing innovative e-commerce information systems. Typically, such a track starts with one or more, often vaguely articulated, innovative ideas, utilizing new technological possibilities. What is needed during this phase is a sufficiently precise definition of an e-commerce idea, so that stakeholders have a common understanding of the idea. Moreover, it should be assessed whether the idea is feasible, and worthwhile to execute. These activities should be carried out in a relatively short timeframe.

After introducing some terminology (see section 2.1), we present in this chapter general observations made while doing projects in the field of innovative e-commerce idea development (see section 2.2). These observations are based on our consultancy experience, but are also based on an exploratory e-commerce project we carried out. We think that a lightweight requirements engineering approach, as known from the information technology community, can help in exploring these information technology intensive ideas, provided that terminology used is recognizable for business oriented stakeholders. In sections 2.3, 2.4, and 2.5 we discuss the outline and rationale of such an *economic value*-based requirements engineering approach for exploring innovative e-commerce information systems. Finally, section 2.6 presents an outlook on the rest of this thesis.

2.1 e-Commerce and requirements engineering

In this section we define terminology needed for the rest of this thesis. We focus on the notions of *innovative e-commerce information systems* and *requirements engineering*.

2.1.1 e-Commerce information systems

E-business and *e-commerce* are often used concepts in conjunction with the use of internet technology by enterprises and end-consumers. There is a subtle but important difference between e-commerce and e-business. To explain the difference, we use the definition introduced by Hartman et al. (2000) (similar definitions exist, see e.g. Turban et al. (2002)):

An e-business initiative is any Internet initiative - tactical or strategic - that transforms business relationships, whether those relationships be business-to-consumer, business-to-business, intra-business, or consumer-to-consumer.

An e-commerce initiative is a particular type of e-business initiative that is focused around individual business transactions that use the Internet as medium of exchange, including business-to-business as well as business-to-consumer.

These definitions differ mainly in scope: e-business is about supporting and enabling business relationships in general, while e-commerce is about business transactions between different companies and/or end-consumers. We see these companies and end-consumers as *actors* whose goal is to create profit, or to obtain products or services which are of economic value for them. To do so, they perform value activities, for which they need to exchange objects of economic value with each other.

e-Commerce initiatives have in common that they heavily rely on information systems connected by the Internet, which are exploited by one or more actors. Consequently, we defined e-commerce information systems as follows:

e-Commerce information systems are a specific kind of information systems, interconnected via the Internet, and exploited by one or more actors, which support and enable the exchange of objects of economic value *between various actors*.

2.1.2 Innovative e-commerce information systems

The field of Internet enabled e-commerce information systems is very immature, and is only in its initial phase of exploitation. Business executives are now becoming aware that such information systems are key strategic factors in industry, but it is likely that most of the social and economic impact of e-commerce lies ahead (see e.g. Akkermans (2001*b*)). Many e-commerce information systems operate in uncharted territory, and put into practice a new way of doing business, which was hardly possible before the widespread use of the web. We call such systems operating in uncharted territory *innovative* e-commerce information systems. Rogers (1995) defines the notion of innovation as follows:

Innovation is an idea, practice or object that is perceived as new.

In the context of e-commerce information systems, we narrow this definition to information systems which support a *new value proposition*. It is this new value proposition which makes the early phases of e-commerce projects so difficult, and which results in an explosion of design choices during such a project.

New technologies such as the Internet, and more specifically the web, may *enable* such a new value proposition, but can even *enforce* a need for a new value proposition. The latter is best seen for products, which can not only be ordered, but also fulfilled using the Internet or a similar technology. We refer to this phenomenon in general as *selling bits* (Gordijn et al. 2000*d*). Products such as music, movies, software and news are all examples of products which can be delivered using the Internet. For such products, new ways of doing business are enforced, for instance because of the intellectual property right problem. Technology such as the Internet enables end-consumers to copy and redistribute content very easily illegally on a large scale. This forces parties such as intellectual property right owners to rethink their existing way of doing business.

In conclusion, we define *innovative* e-commerce information systems as follows:

Innovative e-commerce information systems are a specific kind of information systems, interconnected via the Internet, and exploited by one or more actors, which support and enable the exchange of objects of economic *value between various actors, with a* value proposition *that is perceived as new, either* enabled *or* enforced *by technological possibilities.*

2.1.3 Requirements engineering

Requirements engineering stands for an approach, often used by information technologists, to develop information system requirements, which can be used as a

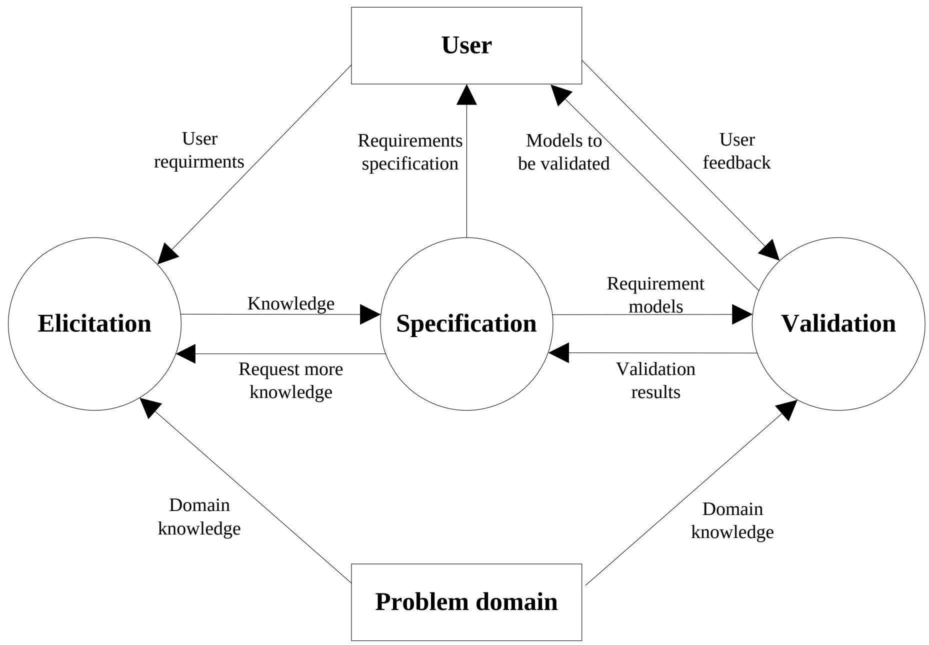


Figure 2.1: Requirements engineering consists of requirements elicitation, specification and validation, according to Loucopoulos & Karakostas (1995).

starting point for system design and implementation. We think that requirements engineering as an approach can be of help in finding, formulating and assessing e-commerce requirements.

Loucopoulos & Karakostas (1995) define requirements engineering as follows:

Requirements engineering is the process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.

Figure 2.1 shows the process of requirements engineering in general. We discuss elements of this process very briefly below.

Elicitation

The purpose of requirements elicitation is to gain knowledge relevant to a problem in order to produce a requirements model. Its input is often dormant knowledge at stakeholders, users, domain experts, literature, and so on. A number of techniques exist to elicit requirements such as interviews, goal analysis, task analysis and scenario analysis. The result of the elicitation process should be a succession of conceptual models in such a way that the performer (e.g. an analyst) understands the problem domain well.

Specification

The requirements specification has a twofold purpose:

- a specification serves as an agreement between stakeholders on the problem to be solved (by a software system);
- a specification serves as a blueprint to allow for the next step; in case of an information system to develop the software.

The elicitation process needs to provide input for the requirements specification process. Output is a specification model, or models which correspond to different views. These models formalize the tacit knowledge of stakeholders.

Validation

The purpose of requirements validation is to check whether the requirements specification complies with stakeholders' intentions. Input is the requirements specification, output is a requirements model which is in sync and consistent with the stakeholders' expectations.

2.2 Innovative e-commerce project observations

In this section, we outline observations made during projects in the field of innovative e-commerce information systems. Our first group of observations concentrates on the exploration of an innovative e-commerce idea. As it turns out, in the first phase of e-commerce information system development, an e-commerce *idea* should be developed into an e-commerce *model*, in which both business- and technical considerations play an important role, and on which stakeholders should agree. Moreover, stakeholders should be confident that the e-commerce idea is feasible from an economic perspective. Such a model then can be used to do a more detailed requirements engineering track. Our second group of observations concentrates on experiences with finding such a model.

2.2.1 e-Commerce idea *exploration* observations

Observation 1: *Exploration of innovative e-commerce ideas is initially about finding a value proposition.*
Explanation. As defined before, *innovative* e-commerce information systems have in common that they exploit a new value proposition. Consequently, e-commerce development tracks start with a new e-commerce idea, which is often articulated vaguely and is subject to change and refinement. Compare this to the development of more traditional systems, which can rely on a known enterprise mission and enterprise goals to derive system requirements.

Example. One of the areas heavily impacted by the popularity of the Internet is the music industry. We carried out a project for a consortium of large music companies in 1998. The aim of this project was to design a music distribution system for the entire music value chain. The underlying e-commerce idea was to distribute and re-sell music from creative performer, potentially via middlemen, to the end-consumer, via the Internet. One of the lessons of this project was that the traditional value proposition to end-consumers was too narrow defined, to be useable as a starting point an investigation of requirements for a music distribution system. The traditional value proposition, e.g. buying a compact disc (CD) by end-consumers, comprises the right to listen a unlimited number of times to music tracks on that CD. This right can be exercised if an end consumer physically has the CD in his/her possession. The Internet, however, enables an end consumer to select a music track online, and to listen to the track using streaming technology. The value proposition is then to listen to a track immediately, and to pay for listening once to such a track. This proposition as well as others were not known at the time the project was carried out, but were necessary to understand to be able to develop a blueprint for a new music distribution system. Consequently, development of new value propositions, which are enabled by new (Internet) technology was an important part of the project.

Observation 2: *Knowledge of information technology plays a crucial role in e-commerce idea exploration.*
Explanation. Following our definition of innovative e-commerce information systems (see section 2.1.2), information technology plays an enabling or enforcing role in such systems. In other words: information technology is an intrinsic part of the value proposition rather than only supporting it. This is different to more old-fashioned information systems, which support internal processes of an enterprise. Such information systems are not visible to consumers directly, while at least parts of e-commerce information systems are directly visible to, and sometimes even of

value for consumers.

Example. By law, radio stations have to pay fees (so called second use rights) to intellectual property right owners (artists, producers, song writers and so on) if they broadcast music tracks. Such a fee is based on the size of the audience listening to the station. During exploration of e-commerce ideas for a Dutch intellectual property right collection society, the possibility came up to offer a highly automated service for clearing second use rights for Internet radio stations. Such radio stations distribute their program using the Internet, rather than using ether frequencies. To offer such a service, it is necessary to have a charging scheme for second use rights in place. A charging scheme indicates how Internet radio stations are billed, so in this case e.g. how the size of the audience is measured. While developing such a scheme, we experienced it is necessary to understand the difference between uni-casting and multi-casting on a technical level. For uni-casting, it is easier to obtain an indication on the size of the audience, while for multi-casting this is more difficult. Understanding of technology was necessary on the boardroom level (CEO, CFO, COO) of the company who initiated the e-commerce idea, rather than only on an operational level.

Observation 3: *A concern of stakeholders regarding an e-commerce idea can sometimes be addressed on the business level* and *on the technical level.*
Explanation. If a concern comes up during the exploration of the e-commerce idea, it may be possible to address this issue by technical measures, but sometimes it is possible to solve this issue by changing the value proposition slightly. This results in an *explosion* of the design space: in traditional system development tracks it is difficult enough to make system design decisions in a systematic way; in an e-commerce development track even more design options become available, because we can easily change the value proposition.

Example. We carried out an explorative project to find a new value proposition for a directory service (Gordijn & van Vliet 1999). A directory service matches needs of a potentially buyer with products or services offered by sellers. The new value proposition was to offer such a matching service, while the seller only pays for *mediated* transactions by the directory service. A mediated transaction is an exchange of services/goods/money between buyer and seller, which is caused by information about a seller, provided by a directory service to a buyer. To implement such a service, the directory service must know the mediated transaction, to be able to charge sellers correctly. While exploring the new directory service, it showed up that some concerns (e.g. denial of transactions reported by the seller) are possible to address on the technical level using encryption technology (see Menezes, van

Oorschot & Vanstone (1996)), while other concerns (e.g. concealment of transactions reported by the seller) should be addressed on the value proposition level. This was not clear while designing a first version of the value proposition.

2.2.2 From an e-commerce *idea* to an e-commerce *model*

From the aforementioned observations can be concluded that developing an innovative e-commerce information system is much more than only developing an information system. It is about finding a promising value proposition in which information systems play a crucial role. Consequently, we first have to find an innovative e-commerce idea, define it more accurately such that stakeholders have a common understanding of an idea, compare it with other ideas, see weaknesses and strong points, and use such a definition to build systems. During this process of idea finding and exploration, the value proposition plays an important role, and so does information technology which enables or enforces the value proposition. Moreover, a value proposition and technological possibilities can interact, leading to changed or even new e-commerce ideas. Consequently, we must find first the direction into which an e-commerce idea goes, rather than starting with a full-blown system requirements engineering track for an often initially vaguely formulated idea.

The result of such an exploration process is an e-commerce *model* rather than an *idea*, which articulates the idea more precisely, so that stakeholders can agree on it, and can validate it. Also, such a model provides a starting point for a more detailed requirements engineering process.

While doing such exploration tracks in e-commerce oriented projects, we made some observations to be presented in the next sections. These observations focus on the first phase of an e-commerce development track: the exploration of an e-commerce idea, and the formulation of an initial e-commerce model.

2.2.3 e-Commerce model *elicitation* observations

Observation 4: *e-Commerce models are created rather than elicited.*
Explanation. Elicitation of e-commerce models supposes that stakeholders have dormant knowledge of an innovative e-commerce idea. It is our experience that in most cases, such an idea has to be invented rather than that such an idea can be elicited (Gordijn et al. 2000*a*). It requires a *paradigm shift* of stakeholders. The paradigm shift theory of Kuhn (1970) recognizes that people think within an

accepted frame of reference and that to be able to find new ideas, people have to step out of that frame of reference.

Example. We have been involved in a number of projects on free Internet access. The idea behind this is to offer a consumer Internet access or even some content (e.g. a newspaper) for free, and to get funded by telephone fees to be paid by a consumer to a telecommunication provider. This proposition was more invented, as a result of new telecommunication regulation, rather than that an elicitation process was started.

Observation 5: *A wide range of stakeholders is involved, thereby mixing up discussions.*
Explanation. In a typical innovative e-commerce exploration track stakeholders range from CxO's (e.g. Chief Executive/Financial/Operation Officers) to information technology persons. The first group of stakeholders is involved because innovative e-commerce projects are about new value propositions which touch the core of companies: how they are making money. The information technology oriented stakeholders play a role to ensure that the enabling or enforcing role of information technology is accounted for. These stakeholder groups have very different foci, which result in mixed-up and inefficient discussions between those stakeholders.

Example. During the mentioned directory service project, initial sessions with the organization exploiting the directory service were held. The sessions were attended by top-level management and information technology staff. Discussions were on a new value proposition and the implementation of this proposition with information technology, all during the same session. In many cases, top level management engaged in information technology issues, and technical people discussed on value propositions. This led to time consuming discussions, because each type of stakeholder had a very different perspective on the issues, and was not knowledgeable in depth on both business and information technology issues.

2.2.4 e-Commerce model *specification* observations

Observation 6: *An e-commerce model is often specified very informally, thereby leading to different interpretations by stakeholders, and hindering analysis and evaluation.*
Explanation. Specification of an e-commerce model, if done at all, tends to be very informal. Often, especially for the value proposition perspective, natural language is used. Such a specification leads to different interpretations by various stakeholder groups. Also, it makes a specification more difficult to analyze and to

evaluate. Moreover, information technology oriented stakeholders require a more precise specification to develop e-commerce information systems.

Example. In the aforementioned directory service, the e-commerce idea initially was stated by a few sentences in natural language. There was some confidence in the commercial potential of the idea, but the general suspicion was that the idea was sensitive to commitments of fraud. However, the way the idea was stated, hindered the articulation and analysis of possible ways to commit frauds. In this specific project, we dealt with this issue by using high level UML sequence diagrams (Rumbaugh et al. 1999) to model message flows between software components of various parties participating in the execution of the idea. We also used another technique, called fault-tree analysis (see Leveson (1996)), to do the actual assessment. This model-based approach assisted us in formulating to stakeholders potential commitments of frauds more precisely.

Observation 7: *A model-based specification mechanism for the value proposition is lacking.*
Explanation. Model-based specification techniques, such as the Unified Modeling Language (UML) exist to represent various information technology requirements from different angles, but there is no technique available for representing a value proposition in such a way. Modeling a value proposition is needed, amongst others to reach common understanding, to be able to evaluate the e-commerce idea more thoroughly, and to allow for a more detailed system requirement elicitation track.

Example. During the development of a free Internet access service for a news paper, two different value propositions were possible (see also chapter 8). These propositions were articulated verbally and gave rise to all kinds of misunderstanding. Modeling value propositions explicitly, using our value-based modeling technique to be introduced in this thesis, helped stakeholders to understand the exact differences between the two value propositions.

Observation 8: *Innovative e-commerce idea exploration is to be carried out in a limited timeframe; typically a few weeks.*
Explanation. To bring an initial e-commerce idea into execution a limited timeframe (typically three to six months) is available. Consequently, the exploration of the idea can take only a fraction of this timeframe. This limitation is caused by rapidly increasing technological possibilities, which cause ideas to become obsolete fast. Moreover companies want to create a high volume operation fast, before a competitor takes market share by developing a similar idea. Therefore, companies typically demand a quick, first execution of the e-commerce idea comprising the

essentials of the idea, rather than a long-term implementation track, which delivers a full blown operation.

Example. The aforementioned free Internet access service had to be realized in three months: from initial idea statement until large scale operational roll-out for hundred thousand (plus) users.

2.2.5 e-Commerce model *validation* observations

Observation 9: *Validation of an e-commerce model initially focuses on feasibility.*
Explanation. The main concern of stakeholders is the issue whether the e-commerce idea is feasible from an economical, but also from a technical perspective. Feasibility study can also mean an investigation of other major concerns, e.g. security.

Example. In the many e-commerce projects we have carried out, in the initial phase each stakeholder wants to see whether 'something is in it' for him/her. During the development of free Internet commerce access ideas, indeed in an early phase value propositions were proposed that were not profitable for a particular actor involved. After evaluation, this led to adjustments of the value propositions.

2.3 Requirements engineering for e-commerce

The key question to be answered in the early phases of requirements engineering for innovative e-commerce applications is the feasibility of the e-commerce idea. This question must be answered for new and fuzzy e-commerce ideas, with many different types of stakeholders involved, and in a short timeframe. Also, stakeholders must have a common understanding of the e-commerce idea, before they can start a more detailed requirements engineering track. In this section we discuss how we address these issues in this thesis. Elements of our solution are:

- a *lightweight* approach to carry out the exploration track in a limited timeframe (observation 8);
- a *graphical conceptual modeling* approach to create a common understanding of an e-commerce idea (observation 6), and to allow for evaluation of an e-commerce idea (observation 9);
- a *multi-viewpoint* approach to deal with a wide range of stakeholders (observation 5);

- a *scenario* approach to create a more common understanding of an e-commerce idea (observation 6), to capture a value proposition (observation 7), and to evaluate such an e-commerce idea (observation 9);
- an *economic value* aware approach to capture a value proposition (observation 7) and to evaluate a value proposition (observation 9).

2.3.1 A lightweight approach

e-Commerce development tracks are characterized by short development times. A typical timeframe is three to six months: from idea to a first implementation. Only a portion of this timeframe is available for exploration of e-commerce ideas. Moreover, to our experience, exploration of such ideas is executed by a small number of persons. So, within a certain timeframe, only limited manpower is available. Consequently, the first phase of e-commerce requirements engineering should be a *lightweight* approach.

The notion of lightweight methods is often used in the realm of formal methods. Although our approach is not formally, but semi-formal (see section 2.3.2), we can use the perspective the formal community takes on lightweight approaches to discuss our own choices. Gervasi & Nuseibeh (2000) define lightweight methods as methods whose adaptation costs is a small fraction of that of the overall requirements engineering process including training, application, and computational costs. To do so, a lightweight approach should support *partiality* in language (it should be tractable), in modeling, in analysis, and in composition (Jackson & Wing 1996).

Regarding *partiality in language*, we employ only a limited number of language concepts on our viewpoints to express requirements. Also, we limit the number of requirement viewpoints to be developed (see section 2.3.3). A similar approach is followed by Nunes & Cunha (2000) to construct a lightweight version of the UML. Although a language can be tractable, it is still possible to construct heavyweight models using such a language, To have *partiality in modeling*, it is necessary to emphasize the goals to be reached with the help of developing models, to avoid modeling details which do not contribute to those goals. A first goal is to create common understanding of the essentials of an e-commerce idea. A second goal is to gain confidence in the economic feasibility of an e-commerce idea. To this end, we focus only on *substantial* expenses and revenues related to the idea, to create confidence. This is sufficient to do a sensitivity analysis. Moreover, it is impossible to find in a first phase detailed financial effects. Therefore, we have only *partial analysis*: we only analyze value propositions, business processes and information systems with respect to substantial revenues and expenses. Finally, lightweight methods should have *partiality in composition*. In practice, for a large system many different requirement viewpoints exist (see also section 2.3.3), which should be composed to allow some analysis of consistency. We use a number of viewpoints, and we relate these viewpoints only very loosely by using operational scenario's (see also section 2.3.4).

2.3.2 A graphical conceptual modeling approach

A *conceptual modeling* approach comprises the activity of formally defining aspects of the physical and social world around us for the purpose of understanding and communication (Mylopoulos 1992). *Formal* in this context means the abstraction, structure, and representation of knowledge in a way that makes it possible to reason about this knowledge (Loucopoulos & Karakostas 1995). The activity of modeling is well-known and accepted in the information technology community for describing information system requirements, but it is our experience that business-oriented stakeholders are often unaware of this approach. Such stakeholders use natural language requirement representations. There are a number of drawbacks with such representations, such as noise (irrelevant information), silence (omission of important information), overspecification, contradictions, ambiguity, forward references, and wishful thinking (Meyer 1985).

Our experience is that a conceptual modeling approach can be useful for the exploration of e-commerce ideas, provided that models can be easily communicated to business oriented stakeholders. Our goals to exploit a conceptual modeling approach are (1) to enhance the common understanding of an e-commerce idea amongst stakeholders (compared to informal, textual outlines of the e-commerce idea), and (2) to be able to evaluate an e-commerce idea with respect to economic feasibility. For both purposes, it is necessary to have a language which can be used to express e-commerce models. The semantics of this language should be well and commonly understood by stakeholders to facilitate a common understanding of models expressed in the language. Moreover, to facilitate a common understanding, we choose our language constructs in such a way, that they closely resemble the perspective stakeholders have on the e-commerce idea. To allow for evaluation of the e-commerce idea, semantics should be chosen in such a way that assessment of economic feasibility is possible. We use a semi-formal approach rather than a strict formal approach because many stakeholders involved in this phase of idea exploration do not understand very formal models well.

To allow for an *easy communication* with stakeholders, we prefer a lightweight approach (see section 2.3.1), but also a language with a *graphical* syntax. Many approaches used in the realm of information systems employ a graphical approach for representing requirements to contribute to an easy communication with stakeholders (see e.g. Wiegers (1999)).

2.3.3 A multi-viewpoint approach

It is widely accepted that the development of an information system can be very complex, amongst others caused by a wide range of perspectives taken by various stakeholders on the system to be developed. These perspectives are grounded in differences in skills, responsibilities, knowledge and expertise of stakeholders (Finkelstein et al. 1992). This is even truer for the development of innovative e-commerce information systems, where besides stakeholders with a technical- or traditional business background, also value proposition oriented stakeholders are involved. It is our experience that during innovative e-commerce projects, value proposition oriented stakeholders play a dominant role, because such projects create new products or services for an enterprise.

In the realm of requirements engineering, viewpoints are seen as a mechanism to deal with the aforementioned multi-perspective problem, by decomposing complicated requirement issues into self-contained perspectives, which can be addressed and decided on relatively independent from each other. As such, Finkelstein et al. (1992) define a viewpoint as a combination of the idea of an ‘actor’, ‘knowledge source’, ‘role’, or ‘agent’ in the development process and the idea of a ‘view’ or ‘perspective’ which such an entity maintains.

One of the problems with viewpoint approaches is to find suitable viewpoints in the first place. Because we want to use viewpoints as a way to clarify stakeholder discussions, we use the various kinds of stakeholders as an important driver for viewpoint identification. In this thesis we distinguish three stakeholder type related viewpoints. Section 2.4 discusses these viewpoints and the criteria and assumptions we used to identify these viewpoints in more detail.

2.3.4 A scenario approach

Scenarios are used in several communities (e.g. requirements engineering (Rolland et al. 1998), human computer interaction (Carroll & Rosson 1992), and strategic decision making (van der Heijden 1996)), amongst others to engage stakeholders in the requirements engineering process by finding requirements by example, to explain requirements and to validate and to assess requirements more easily.

Rolland et al. (1998) present a framework for classifying scenarios, which we use to discuss the way we employ various forms of scenarios in this thesis. They distinguish four aspects of scenarios:

- the purpose of applying a scenario;
- the contents of scenario: what does it describe;
- the form of a scenario: is it e.g. informal, semi-formal, or formal;
- the lifecycle of a scenario: is it used throughout the entire engineering process, or is it thrown away after use.

With respect to the purpose of a scenario, Antón & Potts (1998) distinguish (1) *operational* scenarios, and (2) *evolutionary* scenarios. By describing system behavior, operational scenarios may contribute to a better understanding of such a system by stakeholders. Evolutionary scenarios are used to envision events in the life of a system that may cause the system to change. The notion of system should be interpreted rather broad; we see a network of actors exchanging things of value with each other as a system also. We discuss the use of operational and evolutionary scenarios in our approach along the lines of model elicitation, specification, and validation.

Elicitation

The *purpose* of operational scenarios during elicitation is to capture an e-commerce idea, thereby providing a starting point for modeling such an idea. From a *content* point of view these scenarios put into operation an end-consumer's need. We choose end-consumer needs as our starting point for scenario specification, because this enforces stakeholders to take an outside-in perspective that is to think from the consumer's perspective. Scenarios in this stage have a narrative, informal and textual *form*. Often, it are a few one-liners, together stating parts of the e-commerce idea. Regarding the *lifecycle*, these scenarios are used throughout the exploration process.

Specification

During specification of an e-commerce model, the mentioned operational scenarios are detailed and defined more precisely by *scenario paths*. A scenario path can be seen as a specific instance of a scenario, and shows causal relations between events. Scenarios paths take the *form* of use case maps (UCMs) (Buhr 1998). UCMs are a lightweight, semi-formal, graphical way of specifying scenario paths. The *content* captured by scenario paths depends on the viewpoint they are defined for. The *purpose* of scenarios during specification is twofold. First, scenarios are used to create a common understanding between stakeholders, and are used to explain viewpoint models to stakeholders. Second, scenarios are used to relate and integrate viewpoints. A danger of using various requirement viewpoints is that they become unrelated, and/or inconsistent (see e.g. Finkelstein et al. (1994)), while they should refer to the same phenomena in reality. To address this, we use the *same* scenarios, which stem from the elicitation phase, and detail these scenarios on the three viewpoints with *different* scenario paths. Because scenario paths on the various viewpoints are based on the same scenario, we think viewpoints remain related. This approach is based on Kruchten (1995), who uses scenarios to integrate multiple viewpoints also. Again, these scenarios are persistent over the lifetime of the requirements engineering process for the e-commerce idea at hand.

Validation

A main requirement with respect to the e-commerce idea is that a reasonable possibility should exist for all actors that the e-commerce idea will be profitable, or in case of end-consumers, produces something of economic value for them. It is not possible to proof this, but doing sensitivity analysis may contribute to an increased confidence in the commerce idea. Validation of the e-commerce model is therefore mainly about assessing profitability. For doing so, we exploit two types of scenarios: (1) the aforementioned operational UCMs, and (2) evolutionary scenarios. The scenario paths expressed by UCMs facilitate assessment of the e-commerce idea from an economic perspective. They are used to calculate estimations on revenues and expenses on a per actor basis.

The *purpose* of evolutionary scenarios is to do a sensitivity analysis. These scenarios take an informal *form*, they are expressed in natural language. Their *content* comprises possible, likely changes in the future, with respect to an e-commerce model such as (dis)appearing actors, or a change in the way actors assign economic value to objects they receive or deliver. Concerning the *life cycle* aspect of scenarios, our evolutionary scenarios are only of relevance during assessment of the e-commerce model.

2.3.5 An economic value aware approach

In most cases, requirements engineering focuses on *information system* requirements. Over the past few years it has been understood that is also important to know the business goals an information system should contribute to. This is reflected in the realm of goal oriented requirements engineering (see e.g. Yu & Mylopoulos (1998)). In goal oriented requirements engineering approaches, often AND and OR goal trees are constructed to derive (alternative) system requirements supporting these goals. We tailor our approach more to the realm of innovative e-commerce information systems in a way that we see as a first goal that participating actors make profit or obtain products or services which are of economic value for them by exploiting and using the system.

The notion that an information system exploited by some actor (e.g. an enterprise or an individual) offers something of economic value to another actor, and that such a value proposition is an important profit generator for actors, is relatively new to the requirements engineering community, but is typical for many e-commerce projects. In contrast, development of valuable products or services in general is addressed by the various economic disciplines, and especially in the field of marketing. However, an integral approach for developing information technology intensive commerce ideas, which are perceived as valuable by customers can hardly be found. An exception is Strassmann (1997) who relates profits and revenues of enterprises to costs of information technology. His approach however is not supported by a lightweight graphical modeling technique, which can be used to explore e-commerce ideas, and is also not very well known in the requirements engineering community.

For *innovative* e-commerce ideas we think it is key to develop requirements (so not only information system requirements) in an integral way, from a business value and information system perspective. An important rationale is that business value and information systems are heavily intertwined in e-commerce development tracks. To this end, the approach of requirements engineering, being the elicitation of requirements, the specification requirement models, and the validation of requirements, may not only useable for developing system requirements, but may also be useable for developing business value requirements. We deal with this in more detail in section 2.5.

2.4 Three e-commerce viewpoints

In this thesis, we expect from a multi-viewpoint approach that discussions and decisions on requirements issues are done with the right group of stakeholders, to avoid time consuming stakeholder sessions. Consequently, our identification of viewpoints is mainly driven by the different types of stakeholders we distinguish during the e-commerce idea exploration phase. Before discussing these viewpoints in more depth, we first present basic assumptions and criteria we have used to identify these.

2.4.1 Viewpoint identification assumptions

We take the approach of identifying viewpoints *in advance* rather than doing so during the exploration process and we only use a *limited* number of viewpoints. Both these basic assumptions are discussed in the next paragraphs.

Viewpoints for profitability analysis are predefined

According to Motschnig-Pitrig et al. (1997) viewpoints can be *fixed*, and *predefined*, or viewpoint identification can be *part of the requirements engineering process*. An advantage of using fixed viewpoints with a fixed representation style is that they can be of help in the requirements elicitation process (Sommerville & Sawyer 1997), by helping to ask the 'right' questions. A drawback of fixed viewpoints is that they are not easily applicable across different application domains. Therefore, many multi-viewpoint approaches do not use predefined viewpoints but instead see viewpoint identification as a step in the requirements engineering process. A disadvantage of having no predefined viewpoints is that for every project, time is lost with viewpoint identification. As our major concern is profitability assessment in a specific domain, and because we want to allow for a fast exploration track, we use predefined viewpoints for the profitability concern.

As indicated before, during idea exploration other stakeholder concerns may come up. In such a case, suitable viewpoints must be found to represent, analyze and evaluate requirements addressing those concerns.

Viewpoints are limited in number

To execute the exploration track fast, it is necessary to limit the number of viewpoints to be developed. This is also stated by Sommerville & Sawyer (1997) who suggest, especially in the early phases of requirements engineering, to limit the number of views to be developed, at a maximum of six. More viewpoints can lead to an explosion of information which must be managed and results into an unduly expensive elicitation process.

2.4.2 Viewpoint identification criteria

Which kind of viewpoints should be used during the exploration phase of innovative e-commerce information systems? We use the following criteria: (1) a viewpoint should contribute from a content point of view to the assessment of economic feasibility of an e-commerce idea, (2) a viewpoint should be based on a similar focus of a group of stakeholders, and (3) a viewpoint's focus should have a minimum overlap with foci of other viewpoints.

Viewpoints should contribute to analysis of profitability

Sommerville & Sawyer (1997) define a *concern* as an organizational goal or critical success factor. Concerns are orthogonal to viewpoints, and thus can be relevant for each viewpoint. The major concern during exploration of an e-commerce idea is the profitability of such an idea for all actors involved. So we need viewpoints that express information which can be used to evaluate profitability.

It is possible that other important critical success factors are discovered during exploration of the e-commerce idea. Gordijn & van Vliet (1999) discuss such a project, where security, or a possibility to commit a fraud is an important issue. In such as case, an additional concern is added to the profitability concern, and as such may need additional viewpoints to address these concerns.

A viewpoint should be based on a similar focus of a stakeholder group

Sommerville & Sawyer (1997) define the *focus* of a viewpoint as an explicit statement of the perspective taken by that viewpoint. It outlines the purpose or topic of a viewpoint (Motschnig-Pitrig et al. 1997). Viewpoint foci can be based on various sources (Sommerville & Sawyer 1997): (1) interactions between the information systems and humans or other information systems, (2) indirect stakeholders, being humans (possibly representing organizations), which have an interest in the system, but do not interact directly with it, and (3) domain characteristics, which can not be identified with a particular stakeholder or interaction, but nevertheless impose requirements which are implicit in the domain under consideration.

We base each viewpoint on similar foci of indirect groups of *stakeholders* participating in an e-commerce idea exploration track, thereby distinguishing groups of stakeholders taking a similar perspective on requirements. The reason for doing so is to focus and to clarify discussions. In these exploratory discussions, mainly *indirect* stakeholders participate, because in the initial phase of e-commerce information system development, stakeholders who decide on the execution of an idea participate, and such stakeholders need not to interact with the e-commerce information system directly. For the same reason of focused and clear discussions, we have only one focus per viewpoint.

A viewpoint's focus should have a minimum overlap with foci of other viewpoints

Foci of viewpoints should have a minimum overlap, to ensure that viewpoints become relatively *self-contained*. Otherwise requirements on a particular viewpoint should be developed with too much consultation of other stakeholders. Self-containment refers also to ability of stakeholders to *decide* on requirements expressed on a viewpoint.

This self-containment of viewpoints is also acknowledged by Finkelstein et al. (1992). According to them, a viewpoint is a *loosely coupled*, *locally managed* object which encapsulates partial knowledge about the system and domain, specified in a particular, suitable representation scheme, and partial knowledge of the process of design.

In conclusion, viewpoints for e-commerce idea exploration: (1) should be predefined, with additional, on the fly defined viewpoints if other major concerns arise, (2) should be limited in number, (3) should address profitability as a major concern, (4) should each have one focus, which is based on a similar focus of a stakeholder group, and (5) should each have one focus, which should not overlap too much with foci of other viewpoints.

2.4.3 e-Commerce viewpoints

Based on our assumptions and criteria, table 2.1 presents a limited number of predefined viewpoints, which we use for e-commerce idea exploration. We have derived these viewpoints by identifying stakeholder groups with a clear focus, we have encountered during e-commerce projects we have carried out. Table 2.1 shows the name of a viewpoint, the focus of that viewpoint, ways to represent the viewpoint, the viewpoint holders, and the viewpoint engineers. A viewpoint *holder*

Table 2.1: For the development of e-commerce information systems three distinct viewpoints are important: (1) the business *value* viewpoint, with a focus on the way economic value is created, exchanged and consumed in a multi-actor network, (2) the business *process* viewpoint, with a focus on a way to put the value viewpoint in operation in terms of business processes, and (3) the *information system* viewpoint, with a focus on the information systems that enable and support e-commerce processes. For the process- and information viewpoints, useable representation techniques are available, but for the value viewpoint such techniques are lacking.

Viewpoint name	*Viewpoint holder*	*Viewpoint engineer*	*Viewpoint focus*	*Viewpoint representation*
Value viewpoint	CxO's, marketeers, consumer groups	Business developer	*Economic value* object creation, distribution and consumption	e^3-*value* and UCM scenarios
Process viewpoint	Operational management	Business process (re)designer	*Process* ownership and flow, resources needed	UML activity, sequence, interaction diagrams, Petri Nets
Information *system* viewpoint	IT-department	System architect	*System component* ownership	Ownership diagrams

is someone with a direct stake in the viewpoint, while a viewpoint *engineer* is someone only facilitating the requirements engineering process (Motschnig-Pitrig et al. 1997).

The value viewpoint

The top-level viewpoint of our electronic commerce framework concerns the value viewpoint. The value viewpoint *focus* is the (new) way of economic value creation, distribution and consumption. For viewpoint *representation* we employ e^3-*value* models, which are to be explained in this thesis. Viewpoint *holders* are CxO's such as Chief Executive Officers, Chief Financial Officers, etc. Viewpoint *engineers* typically are business developers. The contribution of this viewpoint to the *evaluation* of an e-commerce idea is a statement of revenues and expenses, caused by the exchange of valuable object between actors.

Most CxO's have difficulties in developing and articulating value models themselves. Consequently, viewpoint engineers (business consultants and business developers) play a dominant role. Nevertheless, CxO's should at least understand the essentials and consequences of value models constructed.

The business process viewpoint

The business process viewpoint, the middle level in table 2.1, *focuses* on business processes, which are needed to put into practice a new value proposition, and focuses on ownership of these processes. To *represent* a business process view, a number of techniques are suitable, for instance the UML activity diagrams with swimming lanes to represent actors (Fowler & Scott 1995, Rumbaugh et al. 1999) interaction diagrams, and sequence diagrams, high-level Petri Nets (van Hee 1994), or role-based process-modeling techniques (Ould 1995). Also, business process (re)design approaches (see e.g. Davenport (1993)) are applicable here. The viewpoint *holders* are stakeholders responsible for the design and execution of operational processes. The viewpoint *engineers* are business process designers. For *evaluation* purposes, this viewpoint should highlight: (1) large capital and operational expenses, which are necessary to put the e-commerce idea into operation, and (2) business process themselves, so that stakeholders see that indeed processes can be developed which put into operation the requirements expressed on the value viewpoint.

The information system viewpoint

The information system viewpoint, the bottom of figure 2.1, *focuses* on constituting components of an information system to be developed at a course granularity. Techniques are available to *represent* this viewpoint, such as the techniques offered by the UML. Viewpoint *holders* are stakeholders responsible for development and exploitation of information technology, typically persons working in an IT-department. Information system architects are key viewpoint *engineers* for this viewpoint. From an *evaluation* point of view, this viewpoint is motivated because we want to highlight expected expensive system components, both from an operational expense perspective and a capital expense perspective.

2.4.4 Relations between viewpoints

The main relation between the aforementioned three viewpoints is a *put into operation* relation. The business process- and information system viewpoint requirements put into operation the value viewpoint requirements. Also, the information system viewpoint requirements can put into practice part of the business process defined. Consequently, it is our experience that an exploration track starts with a first definition of the value proposition on the value viewpoint, which is used to find a suitable supporting business process, which is turn is the foundation for an information system.

In conclusion, we distinguish a value-, a business process-, and an information system requirement viewpoint, to facilitate a clear communication- and decision taking process, and to evaluate the e-commerce idea. These viewpoints are grounded in the various stakeholder foci we encountered during e-commerce projects, and in their potential to contribute to an economic value-based evaluation of an e-commerce idea.

2.5 The economic value viewpoint

The previous section identified three requirement viewpoints for the exploration of innovative e-commerce information systems. For the business process- and information system viewpoints sufficiently rigorous representation techniques exist or can easily be thought of. However, such techniques, which mainly stem from information technology community, are not very suitable for a representation of the value viewpoint. They are not aware of the notion of *economic value*, and the role it plays in a multi-party network of actors.

To explore a suitable representation means for the value viewpoint, we first raise questions, which should be answered by requirement expressions on this viewpoint. Second, we discuss some existing ways of representing the value viewpoint in a lightweight, graphically model-based way. These representation styles stem from information management community. Also, we discuss some problems with these styles.

2.5.1 Requirement expressions on the value viewpoint

Innovative e-commerce ideas are about offering services or products (or both) of economic value to actors. In nearly every case, actors offer something of value (at least to someone else) to their environment, *and* request something of value in return. We see such value proposition expressions as important requirements underpinning the e-commerce idea. Typically, requirement expressions seen from an economic value perspective answer the following issues:

1. Who are the business actors involved? To discuss the profitability of an e-commerce idea on a per actor basis, it is important to distinguish them on an instance level.

2. Which objects of economic value are created, exchanged, and consumed by these actors? Each actor wants to know which valuable products s/he produces or consumes. In the context of e-commerce, it is important to show who is doing business with whom because it is relatively easy that intermediaries come in between parties, thereby 'stealing' margin, customer ownership or alike, or disappear between parties.

3. What do actors expect in return for an object of value delivered (the mechanism of economic reciprocity)? Actors want to make profit, often by getting a valuable object (e.g. money) in return for an object they deliver.

4. What phenomena cause exchanges of objects between actors? For instance, an exchange can be caused by an actor need, or by other value exchanges.

5. Which bundles of objects are offered or requested? Bundling assumes that some objects are only requested or offered *in combination.* There are several reasons for bundling. For instance, an actor may assume that two or more objects can be sold against a higher price if they are sold in combination, rather than that they were sold separately (Choi et al. 1997). Bundling is also important in the digital content industry. First, many digital products sold *are* bundles of other digital products (e.g. a game is a bundle of software, videos and sound tracks). Second, some digital products tend to be offered in combination with other valuables. For instance, think of bundles where a music track can only obtained in combination with a T-shirt, a concert ticket or alike.

6. Which value-creating or adding activities are performed by which actors? This is needed to discuss with actors who is doing what, and consequently who is making *profit* with what. Innovative e-commerce ideas often result in a (re)negotiation between actors about who is doing what.

7. What partnerships exist? Actors may decide to offer something of value together to their environment.

8. What are expected profits for actors involved, or in case of end-consumers, is it benifical to pay for a product or service paid? We assume that an e-commerce idea must be economically profitable for each actor involved. To answer this question, we need to know how actors assign economic value to objects received or delivered.

2.5.2 Existing ways for specifying a value viewpoint graphically

We discuss two approaches which are often used in the initial phase of e-commerce idea exploration, and (1) which are somewhat less informal than natural language, (2) which have a graphical representation, and (3) which are lightweight. First, we present the representation introduced by Porter & Millar (1985) (the value chain approach), and second we discuss Tapscott's value maps (Tapscott et al. 2000).

The value chain approach

In the first stage of developing an e-commerce idea, sometimes a *value system* is drawn. A value system comprises multiple enterprises. Each enterprise in the system has its own *value chain*. Such a chain shows the strategic relevant activities for an enterprise. The value chain is intended to analyze competitive advantage by explaining cost leadership, focus, or differentiation strategies. Using linkages between activities dependencies between activities can be shown, for instance the way one activity is performed and a cost influence on another activity.

A typical, more nowadays, use of the value system/chain notation is illustrated in figure 2.2 (Bollier 1996). It shows the sequence of the value-adding process (from manufacturer to consumer) for a networked-based value chain. Its aim is to communicate that the value chain for digital products changes as a result of the increased use of the Internet.

For the representation of requirements on our value viewpoint, this value chain approach misses power in expressiveness. A value system figure does not show who is exchanging objects of value with whom. It only shows the sequence of value adding processes, which is not the same. A value system also does not present the objects of value themselves, and moreover does not recognize the notion of economic reciprocity. Because of this, there is no way of assigning economic value to an object, which in turn is needed to assess profitability for actors involved. Also, it is not possible to show bundling. Finally, with a value system picture, we cannot communicate partnerships.

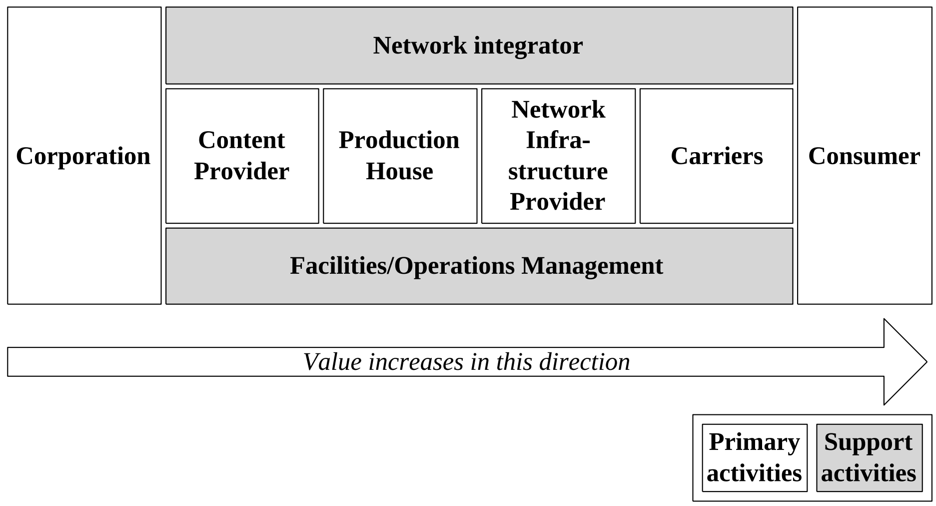


Figure 2.2: An enterprise offering digital content uses a content provider, a production house, a network infrastructure provider, and a telecommunication carrier to offer its product to a consumer. It uses a network integrator and facilities/operations management as support (see Bollier (1996)).

One criticism on the value chain theory is the lack of multi-party offerings (e.g. with partnerships). Normann & Ramírez (1994) have therefore introduced the notion of *value constellations*. Value constellations assume that a number of actors (even the end-consumer can be involved) produce valuable objects. Unfortunately, Normann's theory comes not with an (even informal) representation technique for value constellations.

The value map approach

Tapscott exploits *value maps* to communicate the e-commerce idea (see figure 2.3 for an example). A value map shows actors and exchanges of goods, services, revenues, knowledge and intangible benefits. Value maps are usable to draw quickly on a whiteboard, e.g. during brainstorm sessions, but can not express:

- who is offering what to whom and expects *what* in return (economic reciprocity);
- who is performing which value activities (only actors are recognized);
- the economic value of value objects for actors, needed to assess profitability;

- bundles of value objects;
- partnerships of actors.

Moreover, value maps do not distinguish various stakeholder perspectives very well because there is no explicit focus on valuable objects.

2.6 Requirements engineering for e-commerce revisited

As was motivated in the previous sections, we are searching for a (1) lightweight, (2) graphical, model-based, (3) multi-viewpoint, (4) economic value aware, (5) scenario-based approach to explore, specify and validate innovative e-commerce ideas. Seen from a requirements engineering perspective, an adequate way of developing requirements on the value viewpoint lacks. In our thesis, we focus on developing such a value viewpoint, while keeping in mind that requirements expressed on such a viewpoint should be related to requirements on the business process and information system viewpoints.

Revisiting the elements of requirements engineering (see figure 2.4), the outline of this thesis is as follows.

Chapter 3 and chapter 5 discuss a way of *specifying* requirements on the business value viewpoint, grounded on theory of economic value. In chapter 4, we compare our way of representation with process modeling oriented techniques. Regarding *elicitation* of an innovative e-commerce idea, we acknowledge that finding such an idea is a creative task, rather than an elicitive task. However, once invented an idea, we facilitate the elicitation of alternative ideas by value model reconstruction, which is discussed in chapter 6. Also, chapter 5 contains hints which are useable during an elicitation-like process and discusses how we *validate* an e-commerce idea by assessing its economic feasibility. Chapter 7 and chapter 8 present experiences with development of value oriented requirements. Chapter 9 relates requirements on the value viewpoint to requirements on business process- and information system viewpoint.

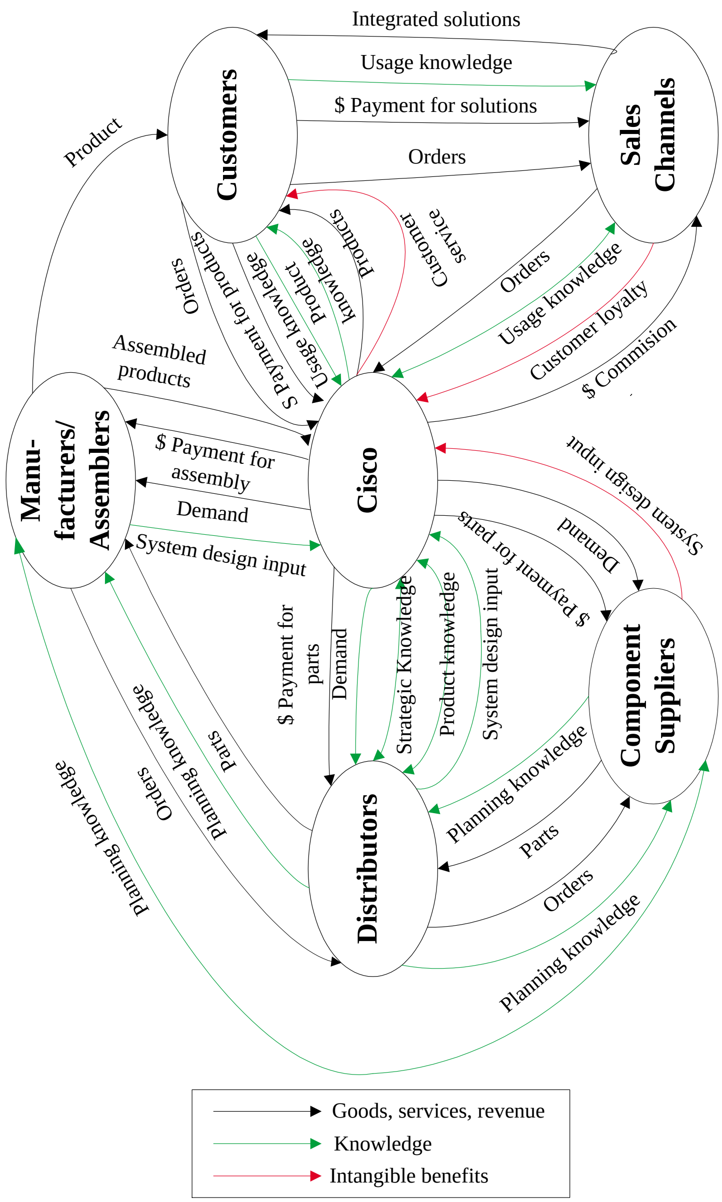


Figure 2.3: Value map for Cisco Systems according to Tapscott et al. (2000).

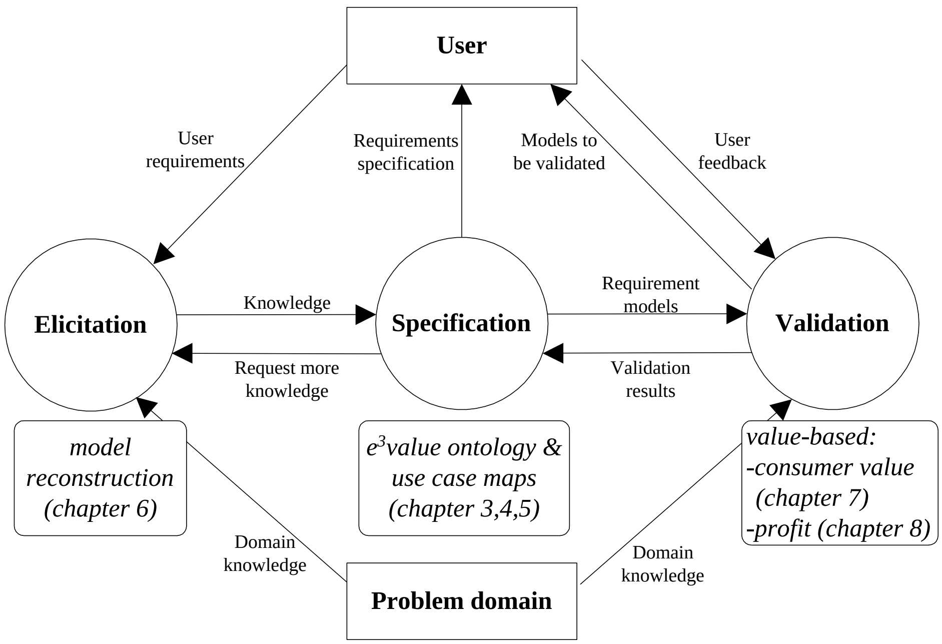


Figure 2.4: Value-based requirements engineering. Requirements elicitation is only partly supported, by value model reconstruction. Value requirements are represented using a lightweight ontology, with on top an operational scenario mechanism, which both can be graphically visualized. Validation is addressed by assessing profitability for actors involved, and assessing economic value of products or services obtained, using evolutionary scenarios.

Chapter 3

The e^3-*value* ontology and scenario techniques

In section 2.3.2 we discussed the need for a conceptual modeling approach for e-commerce idea exploration. Moreover, we argued (see section 2.3.5) that such an approach should be grounded in the notion of economic value. Also, we distinguished three important viewpoints. For one of these viewpoints, the value viewpoint, we need constructs which can be used to represent requirements on such a viewpoint. This chapter presents an *ontology*, called e^3-*value* , which offers such constructs. Cornerstone of this ontology is the notion of economic value, and how actors create, exchange, and consume objects of economic value.

This chapter is organized as follows. After introducing the notion of ontology (section 3.1), we propose an ontology for value models (section 3.2). This ontology is based on well known business concepts such as value chain and value constellation theory, but is also a result of e-commerce projects we have carried out. One of these projects, on free Internet access provisioning, is used to illustrate our ontology. Section 3.3 discusses related ontologies. On top of our ontology we utilize a scenario-based representation mechanism, which we present in section 3.4. In section 3.5 and 3.6, the development of the ontology is outlined and section 3.7 presents our conclusions.

3.1 Ontologies

3.1.1 What is an ontology?

According to Gruber (1994) an ontology can be defined as:

An ontology is an explicit specification of a conceptualization.

The term *ontology* is borrowed from philosophy, where an ontology is a systematic account of existence. In the realm of information systems and AI, ontology has a somewhat different interpretation: an ontology is not a theory of what exists, but what a community of practice believes to exist. This is close to the opinion of Quine (1961), who says that an ontology specifies things that we must assume to exist in order for our theories to be true.

What people believe to exist, we call a *conceptualization*. It represents an abstract, simplified view on the world. In our situation, the simplified world is the world of value propositions.

Modern definitions of ontology (see e.g. Borst (1997)) emphasize that there must be an agreement on the conceptualization that is specified:

An ontology is a formal specification of a shared conceptualization.

This notion of shared conceptualization is important to us, because we aim at a common understanding of value models by stakeholders involved. To contribute to a common understanding, we base our ontology on well known concepts from business science (see section 3.5). Moreover, concepts in ontology are inspired on phenomena articulated by stakeholders, while doing various e-commerce idea exploration tracks (see also section 3.6).

3.1.2 Positioning the *e*^{3}*-value* ontology

To position our ontology, we employ a framework for understanding and classifying applications of ontologies (Jasper & Uschold 1999). Ontologies are often used for information system integration; for instance as a specification of syntax and semantics of information to be exchanged, while we use an ontology to enhance a common understanding between stakeholders. Therefore, not all dimensions in Jasper & Uschold's framework are relevant to us. Framework dimensions which are of interest in our context are:

- the purpose and benefits of an ontology;
- the role of an ontology;

- the actors using an ontology;
- the maturity level of ontology application;
- the way meaning is represented.

Purpose and benefits. One of the purposes of the e^3-*value* ontology is to facilitate communication between people. According to Jasper & Uschold (1999) an informal ontology is then sufficient. Our ontology can be seen as somewhat more formal; this is needed to couple a graphical representation to our ontology and to allow for the way we evaluate value models, which is a second goal of our ontology. As a result of using ontologies, Jasper & Uschold (1999) also report on information system engineering benefits such as enabling re-use, more adequate searching in repositories, more reliability by automated consistency checking, assistance in the process of specification, reduction of maintenance costs, and increase of speed and reliability of knowledge acquisition. Although our value models do not model the information system directly, some of these benefits also are of relevance for value models. For instance, our ontology may assist in the process of identifying requirements and defining a specification. Knowledge acquisition may speed up and may be more reliable using our ontology as a starting point.

Ontology roles. Ontologies themselves can be on various abstraction levels. They vary from ontologies to represent other ontologies, to ontologies which describe things like existing products and their properties. Jasper & Uschold (1999) distinguish the following *roles* ontologies can play:

- role L_0: *operational data*. The ontology captures operational data. Information at L_0 is written using terms from a vocabulary defined at L_1.
- role L_1: *ontology*. The ontology specifies generic terms and definitions for important concepts in some domain. L_1 provides a vocabulary for the language used to author information at L_0.
- role L_2: *ontology representation language*. The ontology plays a role whereby the information is used by ontology authors to write ontologies at level L_1. The information is at L_2 is used to author information at L_1.

Our ontology plays a L_1 role. It provides generic concepts and relations to specify many domain instances, called value models. Such instances can be seen as operational data (role L_0).

Ontology actors. Jasper & Uschold (1999) introduce various *actors* that represent a role a person or application may play: (1) the ontology author, (2) the operational data author, and (3) the knowledge worker. The ontology author defines a L_1 ontology, which is used by the operational data author to create and maintain a L_0 ontology. The knowledge worker is the user of the L_0 ontology.

We play the role of ontology author ourselves. The business developer (see section 2.4) is the operational data author, as s/he develops the value model. Finally, the rest of the stakeholders can be seen as knowledge workers; they are the users of the operational data.

Ontology maturity. Regarding *maturity*, the *e*^{3}*-value* ontology has been developed and used during a number of projects. This is also discussed in section 3.5.

Representation of meaning. The *meaning* of our ontology is represented using UML class diagrams (see e.g. Rumbaugh et al. (1999)), thus by concepts and relations between concepts (see section 3.2). Concepts and relations may have properties. Moreover, in section 3.2.5 we present some constraints and rules well formed value models should comply to.

3.2 An ontology for value models

In this section, we present the *e*^{3}*-value* ontology (see figures 3.1, 3.6, and 3.8), and we illustrate the ontology by a project carried out in the free Internet service provisioning arena. The e-commerce idea underpinning this project is that users, in order to access the Internet, only have to pay a fee for a telephone connection, what they are used to do for other, paid, Internet access services also. In short, these telephone connection revenues are used to finance the entire operation. This e-commerce value model is shown in figures 3.2 (global actor viewpoint), 3.7 (detailed actor viewpoint), and 3.9 (value activity viewpoint).

Before discussing our ontology in detail, we briefly summarize requirements to be expressed using this ontology. We divide these requirements into three parts, which are discussed below.

3.2.1 Three sub-viewpoints

The *e*^{3}*-value* ontology is organized in three sub viewpoints, each discussing related requirement types (see also section 2.5.1).

- The global actor viewpoint shows:

  1. the *actors* involved;
  2. the *objects of economic value* created, exchanged, and consumed by these actors;
  3. objects of value, which actors expect in return for an object of value delivered, or the mechanism of *economic reciprocity*;
  4. objects which are offered or requested *in combination*;
  5. *phenomena* that cause *exchanges* of objects between actors.

- The detailed actor viewpoint(s) shows:

  6. *partnerships* between actors, which show that actors request or offer objects of value jointly;
  7. *constellations* of actors, which need not to be seen on the global actor viewpoint, e.g. to avoid unnecessary complexity;
  8. plus: *requirement expressions* as on the global actor viewpoint, but then only for actors expressed on the detailed viewpoint.

- The value activity viewpoint(s) shows:

  9. the value-creating or adding activities and their assignment to actors.

The main purpose of the *global actor* viewpoint is to explain the overall value model to all stakeholders, including CxO type of stakeholders, involved. It hides complexity, which can be shown on detailed actor viewpoints. The reason to introduce a *detailed actor* viewpoint can be twofold: (1) representation of constellations: a decomposition of a part of the global actor viewpoint to reduce complexity, and, (2) representation of partnerships: actors who decide to offer and/or request products or services as one virtual actor to/from other actors. The *value activity viewpoint(s)* shows what actors do to create profit or to increase value for themselves. Its main motivation is to separate discussions of who is participating in the e-commerce idea from who is doing what.

3.2.2 The global actor viewpoint

The explanation of our ontology is structured by presenting a description for each concept, properties of the concept, relations with other concepts, and the way of

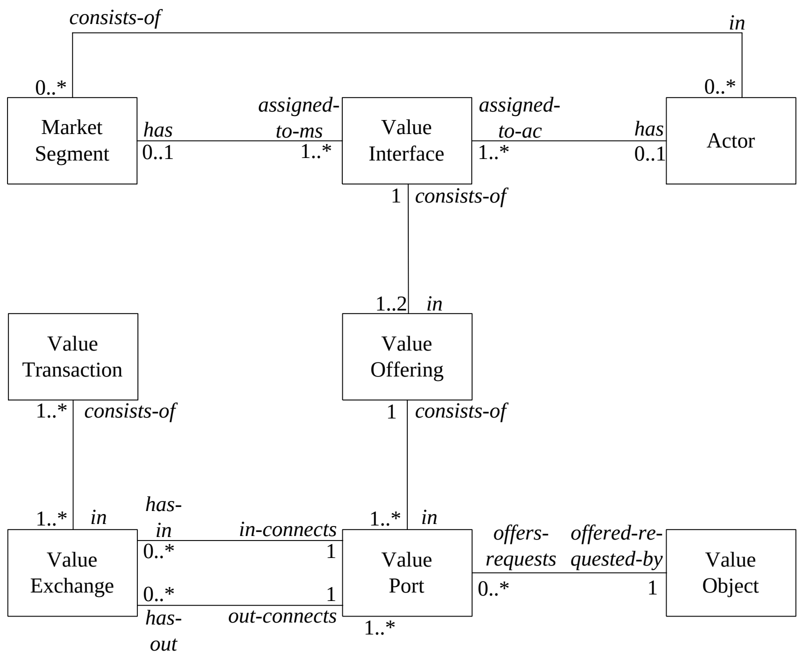


Figure 3.1: Concepts and relations of the *e*^{3}*-value* ontology (global actor viewpoint). The notation is based on UML class diagrams. Rectangles are concepts, related by associations (lines). Concepts play a role in an association. Also, cardinality constraints are expressed. For instance, the association between actor and value interfaces reads: a value interface is assigned to zero or one actor, and, an actor has one or more value interfaces.

visualization in a value model such as depicted in figure 3.2. A concept and relation is illustrated by one or more examples. Figure 3.1 presents the ontology graphically using UML class diagrams.

Actor. An actor is perceived by his/her environment as an economically independent (and often also legal) entity. Enterprises and end-consumers are examples of actors. A profit and loss responsible business unit, which can be seen as economically independent is an actor, although such a unit needs not to be a legal entity.

Economically independent refers to the ability of an actor to be profitable after a reasonable period of time (in case of an enterprise), or to increase value for him/herself (in case of an end-consumer). For a sound and viable e-commerce idea, we require that each actor can be profitable or can increase his/her value. Never-

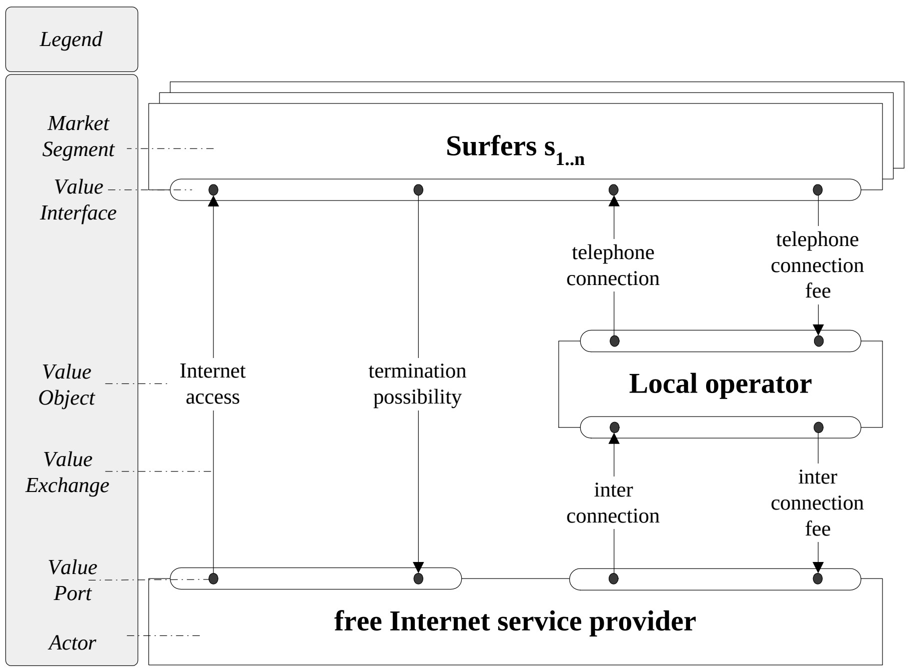


Figure 3.2: Value model for a free Internet access service: the global actor viewpoint.

theless, we acknowledge that in the recent past, many e-commerce ideas were put in operation were this was not case. Such ideas are not sustainable and are consequently not in the scope of our research.

Properties. An actor has a name, e.g. a company name, or a name that represents the role such an actor plays (see also section 5.3.2 on actor identification).

Visualization. An actor is depicted by a rectangle, with his/her enterprise or role name.

Example. The global actor viewpoint (see figure 3.2) shows a *free Internet service provider* and a *local operator*. Also, *surfers* are presented as a market segment (to be discussed), which essentially is a set of actors valuing objects equally. The free Internet service provider is an actor who offers a service the surfer is interested in: Internet access for free. The local operator exploits the local loop: the last mile of copper wire between a telephone switch and the home of a surfer. This loop is needed to set up a telephone connection between a surfer and the free Internet service provider. This telephone connection is used by the surfer's and provider's telecommunication equipment to access the Internet.

Value Object. Actors exchange value objects. A value object is a service, a product, or even an experience, which is of economic value for at least one of the actors involved in a value model. Actors may value an object differently and subjectively, according to their own valuation preferences (Holbrook 1999). We deal with valuation in more detail in subsequent chapters (see chapter 7 for end-consumers and chapter 8 for enterprises).

From a modeling point of view, we are interested in the *kind of* value objects which actors exchange, and not so much in the actual instances themselves. Therefore, when we speak about *value object*, we mean the kind of value object, or the prototype for all instances of a particular value object. In some cases, it is necessary to refer to the actual instances of objects of value exchanged by actors. We then call these objects *value object instances*.

Properties. A value object has a name. While choosing a name, one should keep in mind that it expresses the object from an economic value point of view.

Visualization. A value object is presented by showing the name of the object nearby a value exchange (to be discussed below), representing a potential trade of such an object, or by showing the name nearby value ports offering or requesting objects (see below).

Example. Many value objects in figure 3.2 speak for themselves. The value object *termination possibility* is however non-trivial. *Termination* in the world of telecommunication operators means that if someone tries to set up a telephone connection by dialing a telephone number, someone else must pick up the phone, that is, *terminate* the connection. If someone is willing to cause termination of a large quantity of telephone calls, most telecommunication operators are willing to pay such an actor for that. This is exactly what the free Internet service provider does: s/he aggregates a large number of *termination possibilities* from surfers and gets paid for that.

Also, the value object *interconnection* needs explanation. At the time the project was carried out there was in The Netherlands only one actor who operated the local loop, the last mile of copper wire between a telephone switch and the home of a surfer. From a surfer point of view, this local operator delivers an end-to-end telephone connection, in this case between the surfer and the free Internet service provider. However, the local operator does not operate a network that connects the surfer with the free Internet service provider directly. S/he only owns a part of that network. In such a case, the local operator must use an additional network, connected to the free Internet provider, which is owned by another operator to provide the surfer an end-to-end connection. In other words: the local operator must obtain *interconnection* from another Telco. In return for this, the local operator pays an interconnection fee.

Value Port. An actor uses a value port to provide or request value objects to or from his/her environment, consisting of other actors. Thus, a value port is used to interconnect actors so that they are able to exchange value objects. Such a value object flowing into or out an actor denotes a change of ownership, or a change in rights.

The concept of port is important, because it enables to abstract away from the internal business processes, and to focus only on how external actors and other components of the e-commerce value model can be 'plugged in'. This is the value analogue of the separate external interfaces familiar from technical systems theory (Borst, Akkermans & Top 1997). Take, for example, a bipolar in+out value multiport, which is a characteristic combination occurring in e-commerce value models: an e-service port out and a money port in, or the other way around. Such a bipolar value port combination can be very well compared to an electrical wall outlet. As an external user, you don't want to be involved in what happens behind the wall outlet as long as it gives the right quality of service. The same approach holds for how external parties in an e-commerce value model view the value ports of a service-offering actor: the ports only define how the external connections to other actors should be made.

Properties. A value port has a *direction*, which can have the values *in* (shortly called an in-port) or *out* (called an out-port) indicating whether a value object flows into or out an actor (seen from that actor).

Relations. A value port *offers* or *requests* one value object. This cardinality constraint again emphasizes that we are not so much interested in value object instances, but rather in the prototype for such instances. A value object can be *requested by* or *offered by* zero or more value ports.

Visualization. The value port is depicted by a small black filled circle (see figure 3.2). Value *in*-ports have an incoming arrow. The name of the value object offered/requested by the port can be depicted.

Value Offering. A value offering models what an actor offers to (an out-going offering) or requests from (an in-going offering) his/her environment, and closely relates to the *value interface concept* (see below). A value interface models an offering of an actor to his/her environment, *and* the offering such an actor requests in return from his/her environment. An offering is a set of equally directed value ports exchanging value objects, and implies that all ports in that offering should exchange value objects, or none at all.

A value offering is of use for representing a number of situations. First, some objects may only be of value for an actor if they are obtained in combination. In-ports exchanging such objects then form an in-going offering. Second, actors may decide to offer objects only in combination to their environment. Ports offering such objects then form an out-going offering. An example of an out-going offering is the case of *mixed bundling*. Mixed bundling refers to the mechanism that an actor wants to offer value objects in combination rather than separately, because that actor supposes that different products sold in combination yield more profit than that if they were sold separately (Choi et al. 1997).

Relation. A value offering *consists of* one or more equally directed value ports. A value port is *in* exactly one offering.

Value Interface. Actors have one or more value interfaces. In its simplest form, a value interface consists of one offering, but in many cases, a value interface groups one in-going and one out-going value offering. It shows then the mechanism of economic reciprocity. *Economic reciprocity* refers to rational acting actors. We suppose that actors are only willing to offer objects to someone else, if they receive adequate compensation (i.e. other value object(s) in an in-going offering) in return. So, with the value interface, we can model that an actor is willing to offer something of value to his/her environment but requests something in return, whereas a value offering models that objects can only requested or delivered in combination.

The exchange of value object instances is atomic at the level of the value interface. Either all ports in a value interface (via value offerings) each precisely exchange one value object instance, or none at all. This ensures that if an actor offers something of value to someone else, s/he always gets in return what s/he wants. How this is ensured is a matter of a robust business process design, trust and associated control mechanisms (see e.g. Tan (2002)), legal agreements, or sometimes use of technology, but this is not expressed by the value model.

Relations. A value interface is *assigned to* zero or one actor and *consists of* one or two value offerings, in the latter case being an out-going offering and an in-going offering. Each actor has its own value interface. Multiple value interfaces can be assigned to an actor and a value offering belongs to exactly one value interface.

Visualization. The value interface is visualized by a rounded box at the edge of an actor. Value ports are drawn in the interior of the rounded box. Note that a value offering is not visualized explicitly. However, value offerings can be easily seen by grouping all out-going value ports in a value interface (the out-going offering), or by grouping all in-going value ports in a value interfaces (the in-going offering).

Example. Consider in figure 3.2 the surfer. The in-going offering consists of telephone connection and Internet access. These objects are seen as one offering because they are only of value in combination for the surfer. An Internet connection is worthless without the telephone connection that is used for data transport. Also, for a surfer, the telephone connection is not of value without Internet access. The out-going offering contains the compensations for the obtained telephone connection and Internet access. These two offerings are grouped into a value interface to show that a surfer compensates its environment for obtaining a telephone connection and Internet access, with a fee and a termination possibility.

Value Exchange. A value exchange is used to connect two value ports with each other. It represents one or more potential trades of value object instances between value ports. As such, it is a prototype for actual trades between actors. It shows which actors are willing to exchange value object instances with each other. So, it does not model *actual* exchanges of value object instances, which we call *value exchange instances*.

Relations. The value ports involved in a value exchange are represented by the *has in* and *has out* relations, which relate to exactly one in-port and exactly one out-port. A value port may *connect* to zero or more value exchanges.

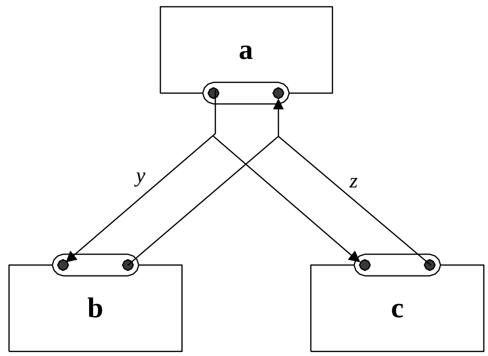


Figure 3.3: Actor *a* can decide to exchange value objects with actor *b*, *or* actor *c*.

Figure 3.3 exemplifies a situation with a port connected to more than one value exchange. Value ports of actor *a*, offering/requesting value objects *y* and *z*, connect via value exchanges to ports of actor *b*, but also connect to ports of actor *c*. This situation models that actor *a* and actor *b* are willing to exchange objects of value, and so do actor *a* and actor *c*. Note that the model does not represent the number of value exchange instances over time, nor their ordering in time.

Visualization. A value exchange is shown as line between value ports. The name of the value object which is exchanged, is presented nearby the value exchange.

Value transaction. A value interface prescribes the value exchanges that should occur, seen from the perspective of an actor the value interface is connected to, because all ports in a value interface should exchange objects, or none at all. Sometimes, it is convenient to have a concept that aggregates all value exchanges, which define the value exchange instances that must occur as consequence of how value exchanges are connected, via value interfaces to actors. We call this concept a value transaction. In its simplest form, a transaction is between two actors. However, a transaction can also be between more than two actors. We call such a transaction a *multi-party* transaction. Figure 3.2 shows a multi-party transaction between a surfer, a local operator, and a free Internet service provider.

Relation. A value transaction *consists of* one or more value exchanges. Note that the exchanges in a transaction should be consistent with the way these exchanges are connected to value interfaces. A value interface requires that if a value object is exchanged via a port, also exchanges must occur via all its other ports. These exchanges must be also part of the transaction.

Figure 3.4 exemplifies why a value exchange can be in multiple transactions. In

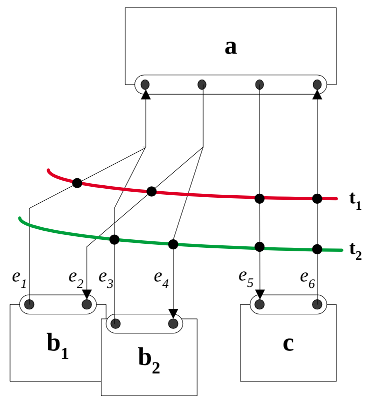


Figure 3.4: A value exchange can be in multiple transactions.

this example, actor *a* offers two value objects, and wants to have two value objects in return. There are two sets of actors who are a capable of participating in the exchange of values with actor *a*: actors $\{b_1, c\}$, and actors $\{b_2, c\}$. Clearly, actor *a* must exchange values with actor *c* (there is no alternative), but there is a choice between actor b_1 and actor b_2 for the other exchanges. Consequently, we can distinguish two transactions with overlapping value exchanges. Transaction 1 consists of the value exchanges $\{e_1, e_2, e_5, e_6\}$ and transaction 2 consist of the value exchanges $\{e_3, e_4, e_5, e_6\}$. Value exchanges, which are in more than one transaction, occur in multi-party transactions, of which figure 3.4 is an example.

Visualization. A value offering is shown by a line intersecting the value exchanges it contains. The intersection points are shown by small filled circles.

Example. Figure 3.2 shows a three-party offering between the free Internet service provider, a surfer, and a local operator. A surfer needs both to obtain Internet access, and to obtain a telephone connection, to be able to browse the Internet. From the surfer's value interface can be concluded that all four value exchanges connected to it are part of one transaction: either all ports of surfer's interface each exchange a value object or none at all.

Market segment. In marketing literature (Kotler 1988), a market segment is defined as a concept that breaks a market (consisting of actors) into segments that share common properties. We employ the notion of market segment to show that a number of actors assign economic value to objects equally. This construct is often used to model that there is a large group of end-consumers who value objects equally. We realize that in practice no actor will value objects exactly the same, but supposing an equal valuation for some actor groups is a simplification needed to arrive at comprehensible value models.

In most cases, the individual actors of a market segment are left implicit. With *implicit* we mean that we do not model these actors individually. This is also the modeling purpose of the market segment construct: to have a shorthand for a large number of actors. However, actors are independent companies or individuals. As such, a specific actor, being part of a market segment, may exchange also other value objects than those mentioned in that market segment. Consequently, a market segment groups *value interfaces* of actors, exchanging objects that are valued equally, rather than that it groups actors themselves. If an actor, who is part of a market segment, has additional value interfaces, which other actors in that segment do not have, we model such an actor also *explicitly*.

Finally, value exchanges drawn to a segment can be seen as a shorthand notation for value exchanges to all actors in that segment. If we assume that market segment b (implicitly) consists of actors b_1, b_2, and b_3, *and* these actors value objects the same way, figure 3.5 (b) is a shorthand notation for figure 3.5(a).

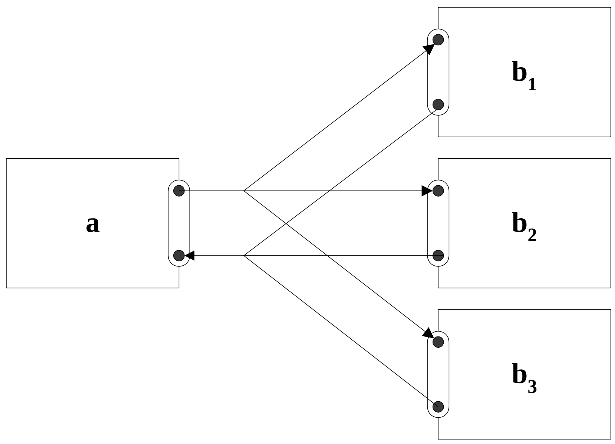
Properties. A market segment is given a name, in must cases in plural form, such as customers, surfers, or alike. A market segment has a *count*, which indicates the number of actors in the segment. The count can be a number, unbound, or unknown.

Relations. Because a market segment is a set of actors, a value interface can be *assigned to* zero or one market segment, just as an interface can be assigned to an actor. Objects exchanged via this value interface are valued equally by actors in the segment.

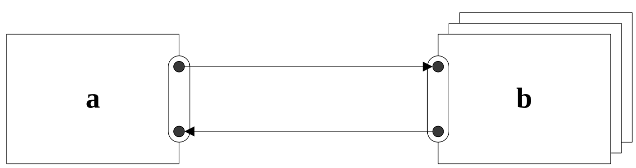
An actor can be *in* a market segment. This relationship is needed to represent actors who have, besides value interfaces of a market segment, additional value interfaces of themselves. The additional interfaces are then related to the actor him/herself, while the relationship between actor and market segment is used to represent an actor's interfaces s/he has as a result of his/her membership in a market segment.

Visualization. A market segment is shown as three stacked actors. A value interface of a market segment is presented on one of the edges of the topmost actor. An explicitly modeled actor who is also part of a market segment is mentioned in the name of the market segment.

Example. The *surfers* segment (figure 3.2) consists of implicit actors who whant to access the Internet.



(a) Actor a exchanges value objects with actors b_1,b_2,*or* b_3,
who may value these objects differently



(b) Actor a exchanges value objects with actors b_1,b_2,*or* b_n,
who value these objects equally

Figure 3.5: A value model without and with market segment.

Summary. In conclusion, the global actor viewpoint shows the top level actors in a value model, without discussing constellations and partnerships yet. Also, the assignment of value activities to actors is not shown by this viewpoint. The global actor viewpoint shows the objects of value exchanged between actors. The market segment notion is useful if a large number of actors exists, who are supposed to assign economic value to value objects the same way.

The global actor viewpoint can be constructed in brainstorm sessions and workshops with all key actors. Also, this viewpoint can be used to present and explain the overall value model to stakeholders.

For the free Internet access service, the global actor viewpoint illustrates that the so-called free service is offered to surfers, but is not for free at all, since the surfer has to pay for a telephone connection. Also, this viewpoint shows that a local operator is needed to offer an Internet access service to surfers.

3.2.3 The detailed actor viewpoint

The purpose of a *detailed actor viewpoint* (see figure 3.7) is twofold. First, a detailed actor viewpoint can be used to *detail* an actor identified on the global actor viewpoint into more actors. We call such an actor a *value constellation*. A value constellation can be used to isolate parts of the value model to a limited number of actors, who can decide on that specific part without consulting other actors participating in the e-commerce idea too much. A value constellation is also a way to reduce complexity on the global actor viewpoint, such that all actors can understand this viewpoint. A second reason to introduce a detailed viewpoint is the representation of *partnerships* between actors. As such, a number of actors may decide to present themselves, as a virtual enterprise actor, to their environment (see e.g. Davidow & Malone (1992)). These actors then decide on one common value interface to their environment.

Composite actor and elementary actor. For both aforementioned modeling purposes, we specialize the actor concept into a composite actor, and an elementary actor (see figure 3.6).

A *composite actor* groups value interfaces of other actors. Also, a composite actor has its own value interfaces to its environment. These composite actor's value interfaces allow us to (1) abstract away from the composite's internals, or (2) to show a common value interface from actors who decide to present themselves as a virtual enterprise.

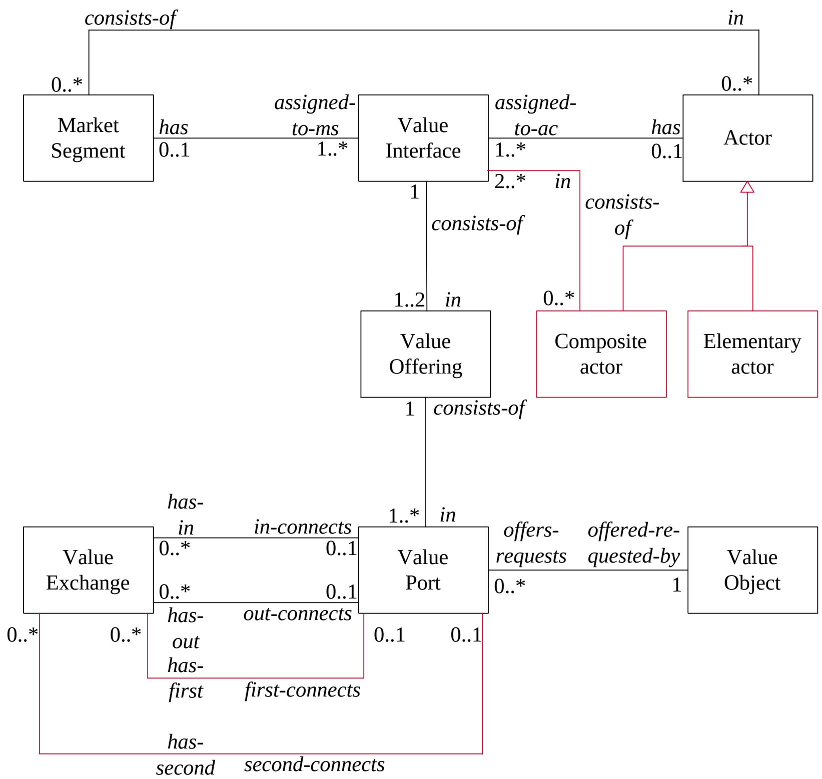


Figure 3.6: Concepts and relations of the e^3-*value* ontology extended for the detailed actor viewpoint. A composite actor and an elementary actor are generalized into an actor.

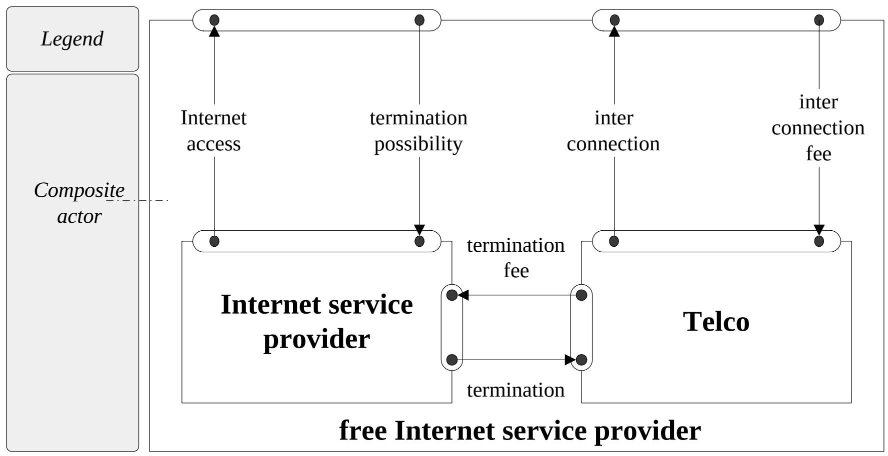


Figure 3.7: Value model for the free Internet case: the detailed free Internet service provider actor view.

An *elementary actor* does not contain value interfaces of other actors. Such an actor is the lowest decomposition level that can be reached from an actor perspective.

Note we group *value interfaces* and not *actors* into a composite actor. The reason for this is that in case of partnerships, an actor may decide to offer objects jointly with objects of other actors, but also may decide to offer other objects on its own. Consequently, it is not the actor that is grouped, but what s/he is offering for a specific case. The same holds for introducing a composite actor in case of value constellations. Such an actor can group a number of value interfaces of the actors it contains, while interfaces of these actors may also appear somewhere else in the value model.

Relations. A composite actor *is an* actor. An elementary actor *is* also *an* actor. This means that all properties and relations identified for actors, will also hold for composite and elementary actors. A composite actor *consists of* minimal two value interfaces of other actors. We need at least two interfaces to be able to group meaningfully.

Visualization. A composite actor is visualized by drawing a rectangle around the actors whose value interfaces are grouped. Inside this rectangle, the value interfaces of the actors must be shown, which are grouped by the composite actor.

Example. The free Internet service provider appears to be a value constellation, which consists of two other actors: (1) an *Internet service provider* offering Internet access (e.g. by exploiting access servers), and (2) a specific *Telco* handling

Table 3.1: Various value exchange types.

Value exchange type	*Relates port 1 of an*	*With port 2 of an*	*Ports have . . . direction*
1	Actor	Actor	Opposite
2	Composite actor	Actor	Equal
3	Elementary actor	Value Activity	Equal
4	Value Activity	Value Activity	Opposite

interconnection of telephone calls between the Internet service provider and the local operator.

The detailed actor viewpoint shows also exchanges of value objects between the Internet service provider and *Telco*. The provider terminates connections by exploiting an Internet access server (effectively a large modem-bank), which answers telephone calls made by the modems of surfers. Termination of large quantities of telephone calls is of value for *Telco*. Consequently, *Telco* pays the Internet service provider a termination fee.

Value exchange revisited. We have introduced the value exchange concept earlier to relate ports of actors exchanging objects. These connected ports have *opposite* directions. The value exchange construct is also used to relate value ports of a composite actor to value ports of actors being part of the composite. In this case, connected ports have *equal* directions. An object offered via an out-port of a composite actor still has to be offered via an out-port of one of the actors in the composite. Also an object requested via a composite actor's in-port must be requested by an in-port of one of the actors it contains.

Properties. To represent the various applications of value exchanges, we distinguish four types (see table 3.1). A type 1 exchange relates ports of actors trading objects, while a type 2 exchange relates ports of a composite actor with ports of the actors it contains. Other types are discussed in the remainder of this chapter.

Relations. To stress that a type 2 value exchange, which connects ports with equal directions is different from a type 1 value interface which connects ports with opposite directions, other associations are shown in the ontology. A value exchange *has a first* value port of the composite actor, and *has a second* value port of one the actors contained by the composite actor.

Example. Figure 3.7 exemplifies a type 2 value exchange. The ports of the composite actor free Internet service provider are mapped on ports of value interfaces of the Internet service provider and *Telco*.

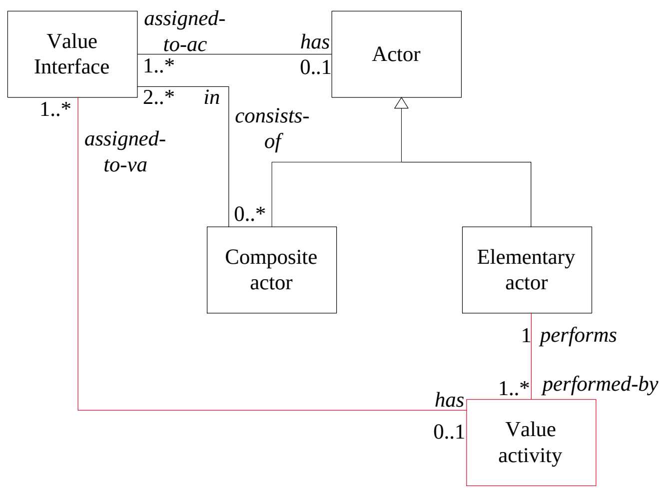


Figure 3.8: Concepts and relations of the e^3-*value* ontology extended for the value activity viewpoint.

Summary. The detailed actor viewpoint intends to represent actors jointly offering or requesting a product or service to their environment, also called a partnership. Moreover, the viewpoint is used to detail specific parts of an e-commerce value model, which are abstracted away on the global actor viewpoint (the value constellation). Strictly spoken, a composite actor groups value interfaces of other actors, not the actors themselves.

3.2.4 The value activity viewpoint

The main purpose of the *value activity* viewpoint is to illustrate the assignment of value activities to actors. Figure 3.9 shows this viewpoint for parts of the free Internet service provider. How value activities are assigned to the various possible actors is a free variable that, as a result of the extended enterprise network setting, leads to many design options and choices in e-commerce value models. Hence, this assignment is a key consideration in strategic e-commerce decision making.

Value Activity. An important issue in value model design is the *assignment* of value activities to actors. Therefore, we are interested in the collection of operational activities which can be assigned as a whole to actors. Such a collection we call a value activity. Actors perform value activities, and to do so, a value activity

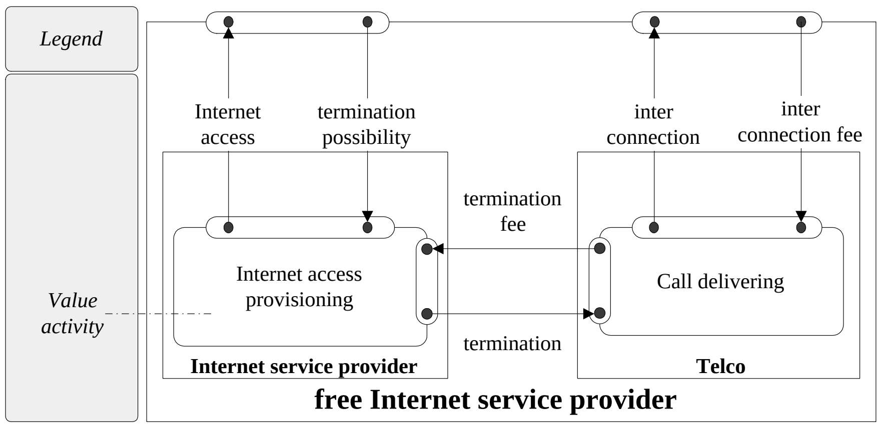


Figure 3.9: Value model for the free Internet case: the value activity viewpoint.

must yield profit or should increase economic value for the performing actor. Consequently, we only distinguish value activities if at least one actor, but hopefully more, believes that s/he can execute the activity profitable. Value activities can be decomposed into smaller activities, but the same requirement stays: the activity should yield profit. This also gives a decomposition stop rule.

Relations. A value activity *has* one or more value interfaces, just like actors and market segments. A value interface belongs to exactly zero or one value activity. A value activity is *performed by* precisely one elementary actor. Finally, multiple value activities can be *performed by* an actor.

Visualization. A value activity is graphically presented by a rounded box, which is drawn inside the actor who performs the activity.

To draw readable diagrams, we sometimes omit value interfaces, ports and exchanges. In figure 3.9, the Internet service provider shows no value interfaces anymore, while figure 3.7 shows for the same actor two value interfaces. If a value interface of an actor has the same structure as a value interface of a value activity s/he performs, we may decide not to present the value interface of the actor. Two value interfaces have the same structure if each port of the first value interface can be matched with precisely one port of the second value interface, and vica versa. Matching of two ports is possible if both ports have the same direction and if they exchange the same value object. However, an omitted value interface conceptually exists, and also value exchanges to connect an actor's value interface to a value interface of his/her value activity conceptually exist. The same holds for composite actors: we may decide to omit value interfaces of a composite actor if they have the same structure as the value interfaces of actors the composite actor exists of.

Example. The Internet service provider performs an Internet access provisioning activity. This activity comprises investment in and maintenance of Internet access servers. Another activity, which might be thought of is e.g. a web hosting service. *Telco* executes an activity named call delivering. This activity is the exploitation of a physical network between the local operator and the Internet service provider for data transport. For all these activities, we assume that they are, after some period, profitable for the actors performing these activities.

Value exchange revisited. We also use the value exchange to connect ports of value activities with ports of the actor performing these activities. These are called type 3 value exchanges. Such ports must have the same direction. Also, ports of value activities, which are performed by the same actor can be connected by using type 4 value exchanges. These exchanges represent 'internal' trades of an actor. Such exchanges connect ports with an opposite direction.

Summary. The value activity viewpoint represents the assignment of value activities to actors. By assuming that a value activity is commercially interesting to be performed by at least one actor, but preferably more actors, we can shift activities from one actor to another actor, thereby discussing who is doing what. Especially if roles of actors are not clear, which is often the case for innovative e-commerce projects, negotiating the assignment of activities to actors is an important part of the exploration track.

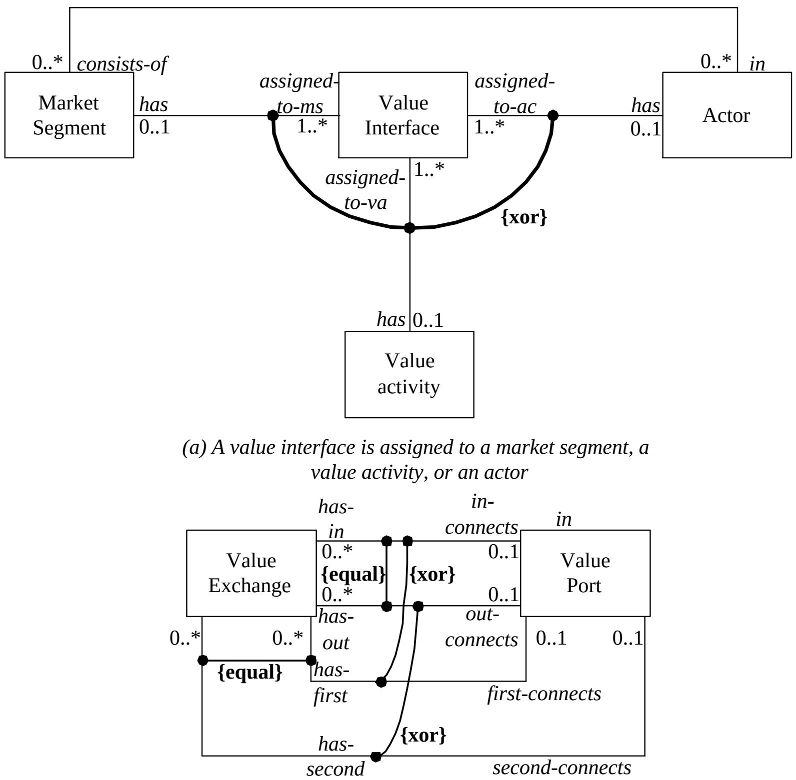
3.2.5 Rules and constraints

A value model is subject to various rules and constraints. Cardinality constraints have already been shown and discussed. Many other constraints can be thought of. Below, we give a non-exhaustive list of such constraints. Some constraints can be specified graphically, while others are specified textually.

Graphical rules and constraints

Figure 3.10 presents the following graphical constraints.

- Figure 3.10 (a): a value interface must be assigned to a market segment, or to an actor, or to a value activity. Moreover, a value interface must be assigned to precisely one of these constructs. This is depicted by the exclusive-or constraint.



(b) Value exchanges connect ports in different ways

Figure 3.10: Graphical constraints.

- Figure 3.10 (b): a value exchange relates, depending on its type, ports with equal or opposite directions, but not both. This is expressed by a number of graphical constraints. By means of *equality* constraints it is stated that if a value exchange has a first value port, it must also have a second port and vice versa (these relations suppose ports with equal directions). Similarly, if a value exchange has an in-port, it must also have an out-port and vice versa (these relations suppose ports with opposite directions). Also, a value exchange must have an in-port, or a first port, and must have an out-port or a second port.

OCL rules

Many rules and constraints are too complicated to express them in a graphical way. To specify these, the UML has the OCL notation (see *OMG Unified Modeling Language Specification, Version 1.3* (1999), Warmer & Kleppe (1999), and Blaha & Premerlani (1998) for ONN, the predecessor of OCL). In this section, we specify these rules and constraints in natural language. In appendix A, their OCL variants can be found.

Value exchange related rules and constraints

- Value exchanges may only connect ports, which exchange the same value objects. The reason for this is to ensure that two actors exchanging a value object assign economic value to the same value object.

- The way value ports are connected by value exchanges is restricted by the *type* of a value exchange, depending on the modeling purpose (see also table 3.1). A value exchange of:

  - type 1 must connect two opposite directed ports in value interfaces of different actors. The modeling purpose is to show trading or exchanging objects from economic point of view; e.g. change of ownership or a grant of rights.
  - type 2 must connect two equally directed ports in value interfaces of different actors, where the first port is in a value interface of a composite actor, and the second port is in a value interface of another actor and the latter value interface must also be in the set of value interfaces grouped by the composite actor. The modeling purpose here is to represent how an actor participates in a partnership or a constellation. In a way, an actor exports its ports to a composite actor.

  - type 3 must connect two equally directed ports, where the first port is in a value interface of an actor, and the second port is in a value interface of a value activity which is performed by that actor. The modeling purpose in this case is to show that ports of an activity are exported to the environment of a performing actor.
  - type 4 must connect two opposite directed ports in value interfaces of different value activities, which are both performed by the same actor. The modeling purpose here is to show actor-internal trades.

- A value exchange is uniquely identified by the ports it connects. A value exchange models the willingness to exchange object value instances between two ports; it is meaningless to represent such a fact twice or more.

Offering related rules and constraints

- A value offering contains only equally directed value ports. This is due to the semantics of a value offering: it models what an actor offers to its environment, *or* what the actor wants to be offered by its environment.

- A value interface contains one value offering, or contains two value offerings. A value interface with only one offering has consequently only in-ports, *or* only out-ports. It models that an actor wants nothing in return (in case of only out-ports), or wants to have a free ride (in case of only in-ports). If a value interface has two offerings, one value offering contains only ports with direction *in*, while the other offering contains only ports with direction *out*. This is the most common situation and models economic reprocity.

Transaction related rules and constraints

- A transaction only contains value exchanges of equal types (being type 1, 2, 3 or 4). Different types of value exchanges connecting ports exist, each with own modeling purposes. The same holds for transactions, which essentially connect value interfaces containing ports, which in turn are connected by value exchanges.

- A transaction indirectly relates value interfaces, because it groups value exchanges, which connect ports of these value interfaces. For each value interface related by such a transaction must hold that each port of such a value interface is connected to a value exchange in that transaction. Otherwise, the semantics of the value interface (exchange via each port of a value interface precisely one object, or none at all), is not obeyed.

- A port which is related to a transaction via one of its value exchanges, must only relate via that value exchange to that transaction. If this constraint does not hold, it possible to connect a same port multiple times with value exchanges part of the same transaction. The semantics of the value interface then would not be obeyed.

Actor composition

- A composite actor can not (even partially) consists of its own value interfaces. This constraint must hold over an entire decomposition chain. In other words: a composite actor can not be decomposed into (parts) of him/herself).

3.3 Related enterprise ontologies

3.3.1 AIAI enterprise ontology

The AIAI enterprise ontology (Uschold et al. 1998) defines a collection of terms and definitions relevant to business enterprises. Two enterprise ontology concepts relate to our ontology but have a different interpretation: (1) *activity* and (2) *sale*. In the enterprise ontology, *activity* is the notion of actually doing something, the how. Our related definition, *value activity*, abstracts from the internal process and in contrast stresses the externally visible outcome in terms of created value, independent from the nature of the operational process. Thus, the defining boundary of what an activity is differs: in the *e*^{3}-*value* ontology the decomposition stop rule is to look at economically independent activities; business process or workflow activities have different decomposition rules, as such activities need not be economically independent. The enterprise ontology further defines a *sale* as an agreement between two legal entities to exchange one good for another good. In our ontology, the concept of sale roughly corresponds to the concept of *transaction*, with the important difference that a sale is an actual agreement, while a transaction is only a potential one. A transaction contains *value exchanges*. In the enterprise ontology, only two goods are exchanged in a sale. In contrast, in our ontology a transaction contains an arbitrary number of value exchanges. This is needed to model a *bundle* of goods that is offered or requested as a whole. Furthermore, our ontology is capable of multi-party transactions. The project in this chapter illustrates the need for such a concept.

3.3.2 TOronto Virtual Enterprise ontology

The TOVE ontology (Fox & Gruninger 1998) identifies concepts for the design of an agile enterprise. An agile company integrates his/her structure, behavior and information. The TOVE ontology currently spans knowledge of activity, time and causality, resources, cost, quality, organization structure, product and agility. However, the interfaces an enterprise has to its environment are lacking in TOVE. Generally, the notion of the creation, distribution, and consumption of value in a stakeholder network is not present in the TOVE ontology. Hence, the TOVE ontology concentrates on the internal workflow of a company, whereas our ontology captures the outside value exchange network.

3.3.3 System-theoretic ontology

As pointed out earlier in this paper, the *e*^{3}*-value* ontology reuses several concepts from general and technical systems theory and associated ontologies (Borst, Akkermans & Top 1997). In particular, the introduction of the concepts of ports and interfaces of a (network) system helps to abstract away from the internal workings of an activity (or subsystem), and to independently specify the connection to the environment (external subsystems). This is an important advance over what is typically done in business process and workflow modeling (Gordijn et al. 2000*c*).

3.4 The *e*^{3}*-value* ontology and operational scenarios

In section 2.3.4, we have discussed the roles scenarios play in our work. *Operational* scenarios are used to capture parts of the e-commerce idea and to contribute to a common understanding between stakeholders. Moreover, we use operational scenarios to integrate viewpoints (see chapter 9), and to evaluate an e-commerce model, in conjunction with evolutionary scenarios (see chapter 5, 7, 8, and 9).

In this section, we focus on a scenario's role to capture parts of an e-commerce value model, and more specifically we show how scenarios are used to specify by what phenomena exchanges of objects are caused (see requirement type 5). To represent operational scenarios, we utilize Use Case Maps (UCMs) (Buhr & Casselman 1999, Buhr 1998), a generic lightweight scenario representation mechanism. The following sections discuss UCMs, bind UCMs to our *e*^{3}*-value* ontology, and discuss differences between our use of UCMs, and Buhr's UCMs.

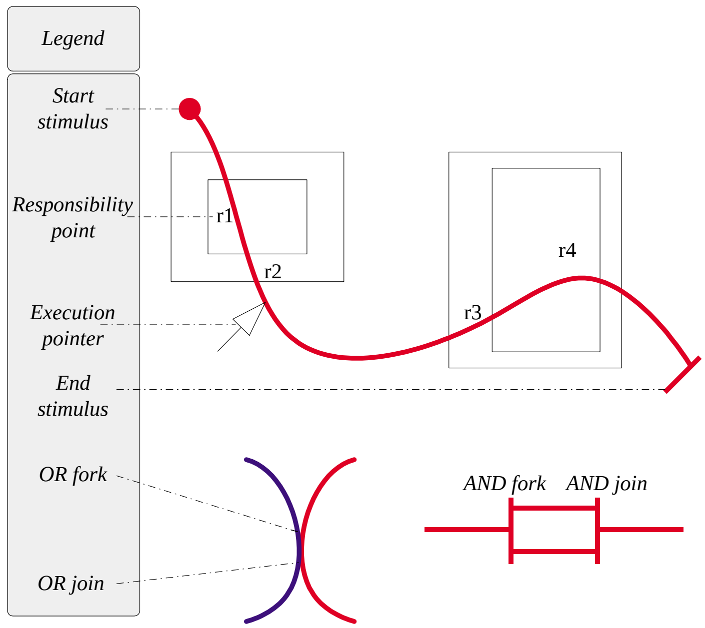


Figure 3.11: UCM constructs.

3.4.1 Use Case Maps

A UCM is a visual notation to be used by humans to understand the behavior of a system at a high level of abstraction (Buhr 1998). It is a scenario-based approach intended to explicate cause-effect relationships by traveling over paths through a system.

The basic UCM notation is very simple, and consists of three basic elements: responsibilities, paths and components. The term component should be interpreted in a broad sense: it may be a software component, but it can also represent a human actor or a hardware system. A simple UCM exemplifying the basic elements is shown in figure 3.11. A path is executed as a result of the receipt of an external stimulus. Imagine that an execution pointer is now placed on the start position (bullet at the top). Next, the pointer moves along the indicated scenario path, thereby entering and leaving components, and touching responsibility points. A responsibility point represents a place where the state of a system is affected or interrogated. The effect of touching a responsibility point is not defined in the UCM itself since the concept of state is not part of a UCM; typically, this effect is described in natural language. Finally, the end position is reached (stroke perpendicular to the scenario path) and the pointer is removed from the diagram.

In the same figure 3.11, two frequently used UCM constructs are shown. The AND construct is used to spawn (AND-fork) and synchronize (AND-join) multiple parallel scenario paths. The OR construct is a means to express that a scenario path continuous in alternative directions.

To be meaningful, the UCM notation must be bound to some other notation, in our case the *e*^{3}*-value* ontology. More specifically, we have to articulate the components UCM scenario paths can touch using responsibility points. Therefore, we present UCM's the same way as we did for our *e*^{3}*-value* ontology, and relate scenario paths to *e*^{3}*-value* ontology constructs.

3.4.2 An ontology for Use Case Maps

A UML-model for the representation of Use Case Maps is shown in figure 3.12 and figure 3.13. It is based on a UCM UML model by Amyot & Mussbacher (2000). Below we discuss the various UCM constructs, and exemplify their use in the free Internet access project. Value viewpoints enriched with Use Case Maps are shown in figure 3.14 (the global actor viewpoint), and figure 3.15 (a detailed actor viewpoint).

Path element. A path element is the generic construct to build Use Case Maps. Path elements are used to relate value interfaces with each other. By doing so, we represent which exchanges of value objects via a value interface cause other exchanges, via other value interfaces.

Properties. Each path element can have a textual label for naming purposes.

Relations. A path element has zero or more *successors*, and has zero or more *predecessors*. The cardinality constraint must be *zero* or more, because path elements can be also be start and end stimuli, and these do not have a predecessor or successor element respectively.

Visualization. Connections (represented by a predecessor-successor relation) between path elements are drawn using normal lines. A path element itself has various visualizations, depending on its type.

Stimulus element. Use case maps start with one or more *start stimuli*. A start stimulus represents an event, possibly caused by an actor. If an actor causes an event, the start stimulus is drawn within the box representing the actor. A use case map also has one or more *end stimuli*. They have no sucessors.

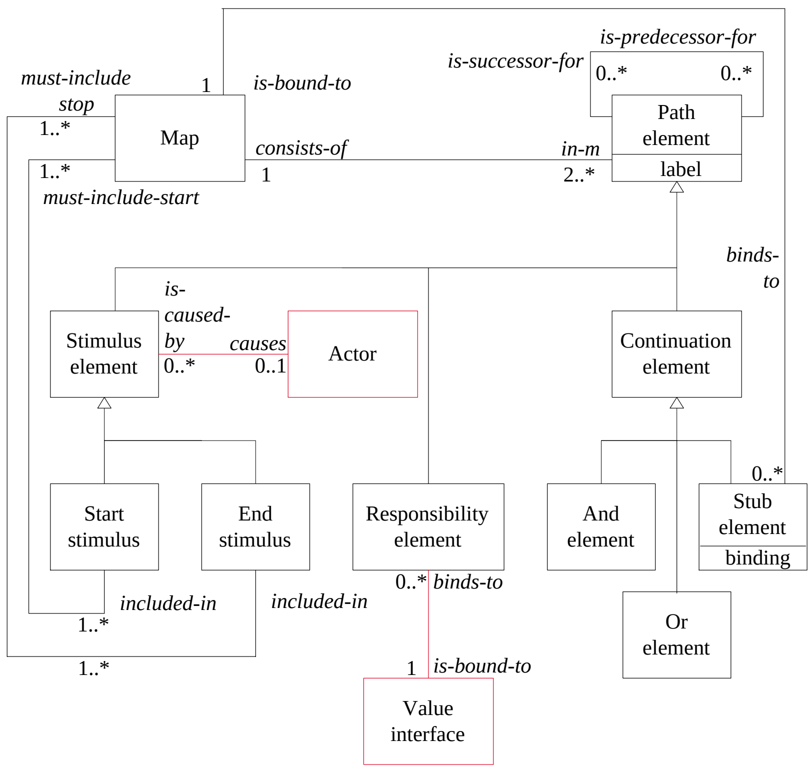


Figure 3.12: An ontology of Use Case Maps based on Amyot and Mussbacher. Concepts of our e^3-*value* ontology are shown in red.

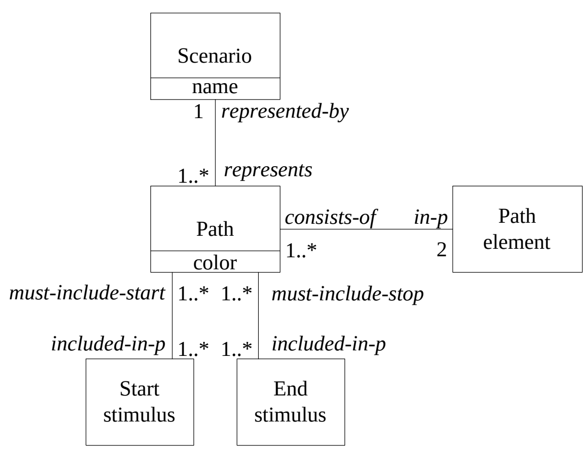


Figure 3.13: UCM ontology extended with concepts for *path* and *scenario*.

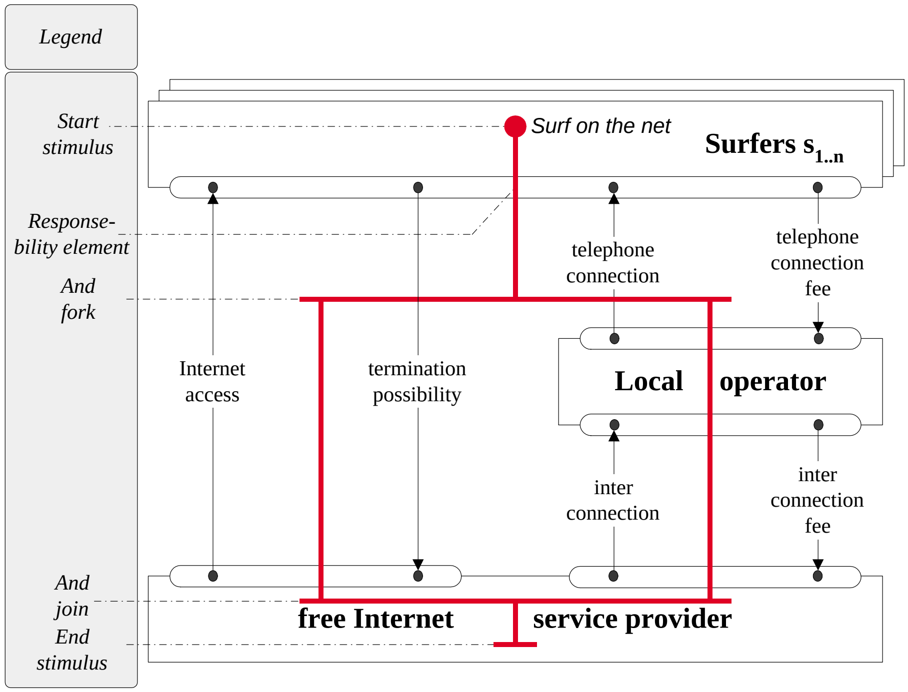


Figure 3.14: Use Case Maps applied to the global actor viewpoint.

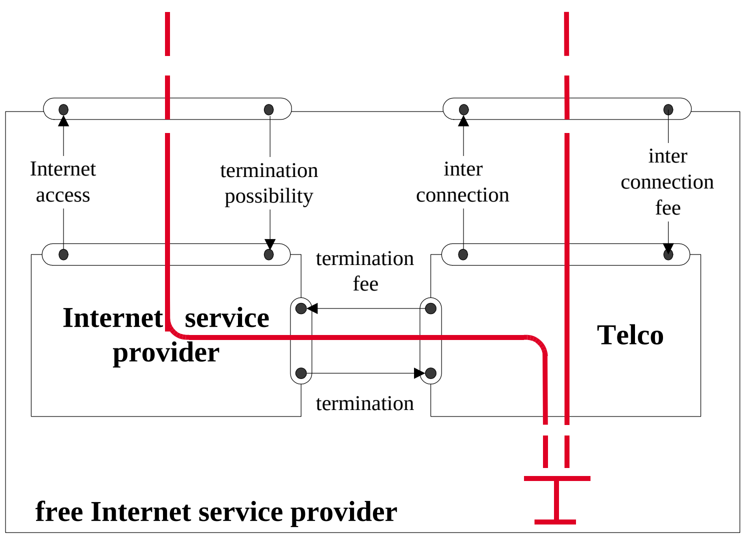


Figure 3.15: Use Case Maps applied to the detailed actor viewpoint.

Relations. A use case map *must include* at least one start and end stimulus. Also, a stimulus element *is caused by* zero or more actors. This relation shows a connection of scenario paths with our *e^3-value* ontology. A stimulus may be caused by an actor, but this not necessary the case.

Visualization. A start stimulus is visualized by a filled circle, an end-stimulus is presented by a line, placed in an angle of ninety degrees on the line visualizing an predecessor-successor relation between path elements (see for an example figure 3.14). If an actor causes a stimulus, it is drawn in the interior of such an actor.

Example. The need for an actor to surf on the Internet is an example of a start stimulus. Such a stimulus results in a number of value exchanges between the actors participating in the value model.

AND and OR continuation elements. An *AND fork* connects a scenario element to one or more other elements, while the *AND join* connects one or more elements to one other element. It splits a path into more sub paths or merges sub paths into one path (see for a path the discussion below). An *OR fork* models a continuation of the scenario path into one direction, to be chosen from a number of alternatives. The *OR join* merges two or paths into one path.

Visualization. An *AND* fork/join is shown as a line, placed in an angle of ninety degrees between lines visualizing predecessor-successor relations between path elements (see for an example figure 3.14). An *OR* fork/join is presented by a number of lines joining into one (a join), or by a line splitting into more lines (a fork).

Example. AND and OR forks/joins can be used to specify a scenario execution in general, but sometimes must be used to comply with the semantics of a value interface. Figure 3.14 shows an AND fork and an AND join. If a surfer wants to access the Internet, s/he needs to obtain Internet access and a telephone connection, which are offered by two different actors. Therefore, the scenario path splits into two sub paths: a first one connecting to the free Internet service provider, and a second one connecting to the local operator. The scenario sub paths are joined again if the free Internet service provider exchanges values using both his/her value interfaces.

Responsibility element. Another way to connect path elements is to use a responsibility element. A responsibility point hits a value interface. These points are important, because they show, for a specific scenario path, when value objects are leaving or entering an actor, market segment or value activity. We use this information to create profitability sheets on a per actor basis to assess profitability (see chapter 5, 7, 8, and 9). Such a sheet shows when objects of value are leaving or entering an actor as a result of scenario path execution.

Relations. A responsibility point *binds to* exactly one value interface. Because multiple scenario paths can touch a value interface, an interface *is bound to* zero or more responsibility points.

Visualization. A responsibility point is shown by intersecting a value interface with a scenario path.

Map. A use case map, or shortly a *map* is a collection of connected path elements.

Relations. A map *consists of* at least two path elements: a start stimulus and an end stimulus (represented by the *must include* relations). Path elements are *in* exactly one map.

Moreover, each path element in a map should have a predecessor and successor element. In case of a start stimulus, no predecesor element exists, and in case of an end stimulus, no successor element exists. Using predecessor and successor relations, elements of a map should be (indirectly) connected with each other.

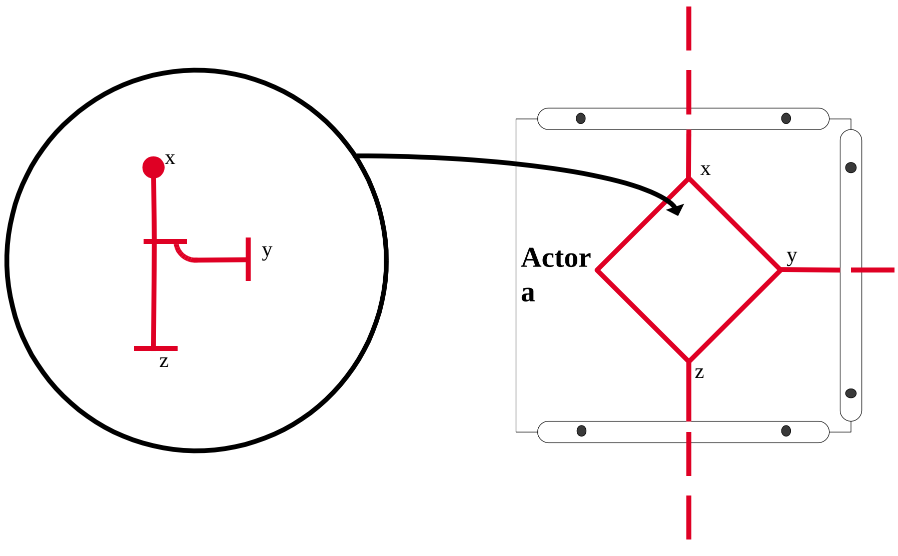
Stub element. A stub is a means to abstract away from complex details of maps. It allows to plug in a map into a stub of another map. An example stub is shown in figure 3.16.

Properties. A stub contains binding properties, which are used to identify the connectors the stub uses to connect to its map (in figure 3.16 *x*, *y*, and *z*). These properties are used to connect the stub connectors to equally named start and end stimuli of the map bound to the stub (see below).

Relations. A stub *binds* exactly one map, and a map can be used in multiple stubs.

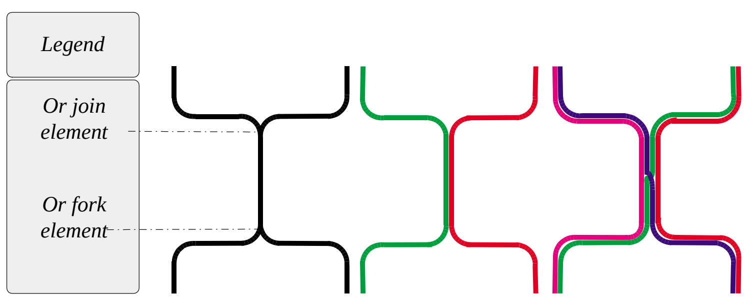
Visualization. A stub is visualized by a diamond.

Path. According to Buhr & Casselman (1999), the OR-forks and joins have no decision logic associated (see also Buhr (1998)). This can result in an explosion of possible routes, or paths, through a map. Not all these paths need to exist in the Universe of Discourse (see e.g. figure 3.17). Also, only some paths may be of interest. Therefore, we introduce a *path* as a way to identify a specific route through a use case map, which may occur in the Universe of Discourse and is of interest (e.g. for evaluation).



(a) UCM elements fitting in a stub *(b) Part of a UCM path containing a stub*

Figure 3.16: A part of scenario path (a) that can be plugged into a stub (b).



(a) Connected path elements *(b) Two paths* *(c) Four paths*

Figure 3.17: Path elements in use case map (a) can be used in more than path, e.g. in two paths (b), or in four paths (c).

Properties. A scenario path has a color property, which identifies the path in a map.

Relations. A scenario path *consists of* multiple path elements, but *must include* at least one start and one end stimulus. Consequently, a path consists of at least two path elements. A path element can be in multiple paths (see figure 3.17 for an example). This contrasts a path to a map: an element can only be in one map.

Path elements which are in a path must all be selected from the same use case map, and must all have one or more successor and predecessor elements (except stimuli). Using predecessor and successor relations, elements of a path should be (indirectly) connected with each other.

OR fork and join elements can be part of a path. However, only *one* successor and one predecessor connected to the OR element can be chosen from the use case map. By doing so, we select a specific route through the map. Note that AND forks and joins do not imply routing decisions, but result in multiple sub paths (in case of a fork), or result in a merge of sub paths. Therefore, a path still may contain AND forks and joins with multiple predecessor and successor elements.

Visualization. A path can be visualized by a specific color, or a specific pattern (such as various kinds of dashed lines).

Scenario. Our notion of scenario is based on the effect that an customer wants to reach. By doing so, we hope to stimulate 'outside-in' discussions with stakeholders. These are discussions which take a customer effect as a starting point. Generally, such a scenario can be put into operation by executing alternative scenario paths. To this end, scenario paths are a way to represent a scenario.

Properties. A scenario has a name, which captures the consumer effect.

Relations. A scenario *is represented by* one or more scenario paths. Such paths show alternative exchanges of values, which may contribute to a same customer effect. A scenario path *represents* one scenario.

Visualization. The relation between a scenario and scenario paths is represented using a textual table. For each scenario, its paths are mentioned.

Example. A scenario in the free Internet access project is the scenario 'surfing the net'. It is represented by one path.

3.4.3 Differences between *e*^{3}*-value* and Buhr's Use Case Maps

Buhr's use case maps are discussed in Buhr & Casselman (1999) and Buhr (1998). Moreover, a conceptualization of use case maps is discussed by Amyot & Mussbacher (2000). Below, we discuss the most important differences between our use case maps as employed in value models, and Buhr's original use case maps as well as the conceptualization by Amyot & Mussbacher (2000).

Differences with Buhr's use case maps

Difference 1: no cycles. Buhr's UCMs may contain cycles. In such a case, a part of a path is executed a number of times and then continues with other parts or stops. However, none of the projects we have carried out, required such a cycle.

Difference 2: no clocks. The original UCMs may contain clocks on a scenario path, denoting that the entering path must wait on an event and then either continue normally or follow a timeout path. UCMs used in value models only model which value exchanges (or start stimuli) cause other exchanges and have no notion of time, let alone time-outs.

Difference 3: no slots and pools. Buhr exploits the notion of slots, which can contain various components obtained from pools. This is used to model self-modifying systems. We see value models not as self-modifying systems, and therefore this feature is not needed.

Difference 4: no dynamic stubs. Stubs are sub use case maps, used to defer details, which can be plugged in another use case map. However, Buhr also distinguishes dynamic stubs. With dynamic stubs, a particular sub path is selected from a set of paths to change the way a scenario is executed on the fly. We assume that for the analysis of an value model, the scenario path is static.

Differences with Amyot and Mussbacher's conceptualization of use case maps

Amyot & Mussbacher (2000) have published a UML conceptualization of Buhr's use case maps. Because we do not use all features of Buhr's use case maps (see above), our UML conceptualization omits concepts for clocks, slots, pools and dynamic stubs. Additional differences are summarized below:

Difference 1: path construct. Amyot & Mussbacher (2000) group path elements into a map. A path element is part of exactly one map. A drawback of doing so, is that is not possible to represent different paths (or routes) through a map, as a result of applying OR-elements (see figure 3.17) explictly. Because we need paths to reason about profitability, we introduce a path construct.

Difference 2: scenario construct. Our scenario construct is not available in the UML conceptualization of Amyot & Mussbacher (2000). We want to capture that a textual customer-based scenario is modeled explicitly by one or more scenario paths. Moreover, our notion of scenario serves as a means to integrate requirement viewpoints (see chapter 9).

3.5 Development of *e*^{3}*-value* : business science perspective

As mentioned in section 2.5.2, value chain theory is often used as a starting point for the design of new e-commerce value models. Normann's value constellation theory builds on the value chain theory but recognizes that actors nowadays work more as a web rather than as a linear sequence of value adding actors. Moreover, Normann stresses the issue of co-production for the creation of valuable objects with a consortium of actors rather than doing so solely. The *e*^{3}*-value* ontology has been influenced by both these approaches as we discuss below.

3.5.1 Value chains are not value models

Porter & Millar (1985) have introduced the value chain theory to explain competitive advantage of firms (see also Porter (1985)). To this end, Porter models a firm as a linear chain of value activities, an approach he still utilizes to analyze the consequences of Internet enabled e-commerce (Porter 2001).

Our notion of value activity is inspired by what Porter calls a value activity. In Porter (1985) a value activity is defined as:

Value activities are the physically and technologically distinct activities a firm performs. These are the building blocks by which a firm creates a product valuable to its buyers.

Value activities are related by linkages, which are defined as:

Linkages are relationships between the way one value activity is performed and the cost or performance of another.

Porter introduces the concepts of value activity and linkage to explain for a particular firm competitive advantage in terms of cost leadership, differentiation, and focus. To do so, linkages between value activities are often cost and performance relations between activities to represent trade-offs, e.g. purchasing (which is value activity) high quality steel (more costly), results in a simplified (cheaper) manufacturing process (again a value activity). Moreover, value activities themselves need not to be profitable.

Our notion of value activity focuses on economic independence, rather than on physical and technological independence. With an economically independent value activity we mean that it is possible, at least in principle, to make profit by performing such an activity. Economically independent activities are needed to discuss with stakeholders the assignment of these activities to performing actors. To facilitate this discussion, we assume that a value activity has the potential to be profitable, and that therefore at least one actor (but preferably more) is interested in performing such an activity. Consequently, activities which only result in expenses should not directly occur at the level of a value analysis of profitability. They are however distinguished in the value chain approach to explain e.g. cost leadership.

Value exchanges can be seen as relations between value activities in the *e^3-value* ontology, as linkages can be seen as relations between value activities in the value chain approach. These exchanges express the willingness to exchange objects of value, rather than that they explain e.g. cost effects of a measure, which is often case with value chain linkages. Using value exchanges, we want to explain which activities (and actors also) want to exchange objects of value with each other.

In conclusion, value activities in *e^3-value* differ from value activities in the value chain theory in a way that *e^3-value* value activities are assumed to be potentially profitable. Relations between *e^3-value* value activities show the willingness to exchange objects of value.

3.5.2 Value constellations are not value models

Normann & Ramírez (1993) introduce the value constellation as a successor of the value chain (see also Normann & Ramírez (1994)). A value constellation is a construct where actors come together to co-produce value with each other. The aim of the value constellation theory is to help actors with continuously (re)designing their business.

Normann argues that relations between actors (enterprises and end-consumers) are not linear anymore such as in the value chain theory, but must be seen as webs or constellations. Such a constellation focuses on the products and services, which actors exchange, and on more long-term business-relations between companies.

A constellation must perform value adding activities to create products and services. With respect to these activities, Normann observes that: (1) actors will perform activities to co-produce a product or service, and (2) end-consumers play an increasing role in co-production.

From an *e*^{3}*-value* ontology point of view, the emphasis on co-production has resulted in a construct such as the composite actor, to model that actors jointly can decide to offer products or services (in case of partnerships), or to model that a number of actors can agree on value exchanges without consulting other actors to much (in case of value constellations). This composite actor aggregates value interfaces, and not actors themselves, because these actors may decide to co-produce specific products and services with other composite actors, or on their own.

Normann also mentions bundling as a concept related to co-production. A co-produced product or service consists of other products or services produced by individual actors. This notion of bundling is in *e*^{3}*-value* reflected by the value offering concept.

Finally, Normann observes that in a modern business environment there is a continuous shuffling of roles: who is doing what? He uses the example of IKEA: in the early days, furniture was sold, assembled and shipped to the end-consumer by the same actor. Nowadays, IKEA offers end-consumers only the parts of a specific product. The customer may decide to transport the product him/herself, or to hire a transport company for that. Moreover, the same customer must perform the value adding activity of assembling the product. In the *e*^{3}*-value* ontology we address the issue of who is doing what by explicitly separating actors from value activities needed to produce a valuable object.

In conclusion, we have used a number of ideas from the value constellation theory. However, this theory does not offer a limited number of related concepts, which can be used to conceptualize a value model. The value constellation theory also does not come with a graphical way of presenting value models. A conceptual modeling approach is however what is needed in a practical e-commerce idea exploration track.

In sum, the *e*^{3}*-value* ontology exploits a number of ideas of the value constellation theory. These ideas are conceptualized in a number of related constructs, with a graphical presentation means, and by a scenario mechanism.

3.6 Development of e^3-*value* : Action Research perspective

As argued in section 1.3, our research approach is close to an Action Research approach. In this section, we report on the way the e^3-*value* has been developed as a result of using Action Research.

The Action Research cycle (Checkland & Holwell 1995) as discussed in section 1.3 comprises the identification of a research theme, and the development of a framework of ideas F embodied in a methodology M to address research theme issues. The framework F, the methodology M and sometimes the research theme changes as a results of a number of research iterations in which the researcher actively participates, e.g. in projects. Each iteration, the researcher tries to use F and M found in a previous cycle, rethinks and even changes F and M during the action (in our case projects), and reflects on this.

We see our e^3-*value* ontology mainly as a declaration of a framework of ideas F in the context of Action Research. Development of such a framework has been the focus of our research (the methodology M how to use such a framework is dealt with in chapter 5). Consequently, we discuss the evolution of our ontology as a change in a framework F over the past few years.

Yellow Pages project

Our first project (September 1997 - November 1997) in the context of this research was on the exploration of a new service for a Yellow Pages like company. (Gordijn & van Vliet 1999). When we started this project, the research theme was broadly defined as ‘how to develop e-commerce information systems’. A finding of this project was that a number of design issues which came up during the development of such a service, can be addressed on the technical (information system) level, but also on the business level. To describe value aspects, we came up with a preliminary framework consisting of actors and exchanges of valuable objects. The framework was rather poor, but already identified the need to investigate value propositions. Also, this project narrowed down our research theme to ‘how to precisely define an innovative e-commerce idea such that it is clear to all stakeholders and such that it allows for profitability evaluation’.

Contact ad service project

Our second project (September 1998 - January 1999) (see Gordijn et al. (2000*a*)) was about the development of an Internet-based, world-wide contact ad service (see also chapter 9). We used the experiences of the Yellow Pages project to construct a first ontology for e-commerce value models. Also, we used the value chain and value constellation theory. The idea was that about 200 local free ad papers would offer to their home market (which is rather local) a new ad service in addition to the paper-based ad paper they each publish. Also, there was one organization responsible for the coordination of these free ad papers. During this project, various value models were developed. However, there was much discussion about the assignment of activities to performing actors. So far, we did not have the notion of value activity in our ontology to facilitate these discussions. A second issue, which came up during the project was that actors wanted to discuss 'fair exchanges'. This motivated us to introduce the value interface concept, and subsequently the value port concept, to address the issue of economic reciprocity.

A second reflection on the aforementioned contact ad service project was the need to explain the drivers for actors to exchange value objects. To model this, we extended the ontology with operational scenarios, and more specifically use case maps, which show stimuli causing the exchange of values between actors, and which present exchanges of values caused by other exchanges of values. In the project, this was used to show stakeholders what happens if an end-customer reads or places a contact ad, in terms of valuable objects exchanged.

Free Internet access project

During June 1999 - September 1999 we carried out our first project in the field on free Internet access (Gordijn et al. 2000*b*). Also, during December 1999 - February 2000, we did a second project on free Internet access for a newspaper (Gordijn & Akkermans 2001*a*). These projects yielded the need for a composite actor to model complex arrangements between actors and joint offerings. We were not able to represent these compositions adequately with the ontology we started with. Also, this project resulted in the introduction of a market segment, to model a large number of similar actors easily, and to differentiate between different segments, e.g. lightweight and heavyweight web surfers.

3.7 Conclusions

We have presented a graphical conceptual modeling approach for the development and representation of value models. The notion of *economic value*, and how objects are created, exchanged and consumed in a multi-actor network is the central theme in our ontology for value models.

Non-trivial e-business ideas, such as the free Internet access idea can be clearly represented using our *e*^{3}*-value* methodology. It has the capabilities to express and analyze many different general mechanisms that are important in e-commerce, including the causality of value exchanges, (un)bundling of value objects, partnerships, and assignment of activities to actors.

On top of our ontology, we exploit a well-known graphical scenario technique called Use Case Maps. Scenario paths are used to explain the causality of value exchanges. The UCM scenario mechanism is, like our ontology lightweight, in a way that it contains a limited number of concepts and relations between those concepts.

To construct the *e*^{3}*-value* ontology, we have used business literature (especially value chain and value constellation theory). However, the ontology construction has mainly been driven by carrying out a number of e-commerce exploration tracks using an Action Research approach. As a result, the ontology substantially differs from concepts found in value chain and value constellation theory. Most notably, we model who is doing business with whom, rather than the increase of value as value chains do.

From an ontology perspective, the AIAI and TOVE ontology both focus on businesses. AIAI comes closest to our ontology, but has no focus on economic value creation, distribution, and consumption, is heavyweight, and has no scenario mechanism. Concepts in TOVE are used to describe enterprises from an organizational, internal perspective, rather than from a value perspective.

Finally, a value model differs from a process model that outlines *how* enterprises create, distribute, and consume value. We elaborate more on this difference in the next chapter.

Chapter 4

A value model is not a process model

A value model outlines who exchanges objects of value with whom, while a process model describes the way a value model is put into operation: the activities needed, as well as their sequence, to create, distribute, and consume value.

Value models and process models can be represented using various techniques. For a process model standard process modeling technique such as the UML modeling language (activity diagrams) (Rumbaugh et al. 1999, Fowler & Scott 1995), Petri Nets (van Hee 1994), $IDEF_0$ (*$IDEF_0$ Method Report* 1981), or STRIM (Ould 1995) are suitable. Also XML-based languages are emerging for describing interorganizational business processes such as the Web Services Flow Language (WSFL) (Leymann 2001), and Web Services Description Language (WSDL) (Christensen, Curbera, Meredith & Weerawarana 2001). For the representation of a value model, we have introduced in chapter 3 *e^3-value* .

The aim of this chapter is to discuss differences between a value model and a process model. To this end, section 4.1 enumerates several types of differences between a value model and a process model. In short, a value model shows how objects of *economic value* are created, distributed and consumed in a multi-actor network, while a process model shows how such exchanges of value objects are put into operation from a business process perspective. Differences between value- and process models are discussed in more detail in sections 4.2, 4.3, 4.4, and 4.5. To exemplify the differences between both models, we show an example UML activity model and a value model. Finally, conclusions are presented in section 4.6.

4.1 Differences between value modeling and process modeling

E-commerce value modeling differs in several ways from process modeling. These modeling approaches each:

1. capture different *stakeholder decisions*. A value model captures decisions regarding *who* is offering and exchanging *what* with *whom* and expects *what* in return. A process model focuses on decisions with respect to *how* processes should be carried out, and by *whom*.

2. use different *modeling constructs*. The concepts in a value model are centered around the notion of *value*, while in process modeling concepts focus on *operational* aspects of a process.

3. represent different *Universe of Discourse statements*. A value model says to which extent actors are profitable, and whether actors are willing to exchange objects of value with each other. A process models states which activities should performed, in which order, and which objects (in which order) flow between activities.

4. exploit different *ways of decomposing activities*. In value modeling we use decomposition of value activities as a way to discover new profitable activities, for instance to discuss new alternative assignments of such activities to actors. Decomposition of activities in process modeling serves the goal of clarity, or studying various resource allocations (e.g. operational actors) to activities.

The aforementioned differences are explained in this chapter, and exemplified using an e-commerce exploration track.

4.2 Different stakeholder decisions

A value model captures other stakeholder decisions than a process model does. In short, a value model shows the essentials (the strategic intent) of the way of doing business in terms of actors creating and exchanging objects of *value* with each other, while a process model shows decisions regarding the way a business is put into *operation*.

4.2.1 Value model decisions

Most innovative e-commerce projects should start with the design of the way of doing business: the value model. Essentially, it provides the design rationale for e-commerce systems from a business point of view.

In our view, the main goal of a value model is to answer the question: '*who* is offering *what* to *whom* and expects *what* in return'. Therefore, the central notion in any value model should be the concept of *value*, in order to explain the creation and addition of value in an multi-party stakeholder network, as well as the exchange of value between stakeholders.

Consequently, the main design decisions to be represented in a value model are:

1. who are the value creating, exchanging and consuming actors involved;
2. what are objects of value created, exchanged, and consumed;
3. what do actors expect in return if they exchange objects of value;
4. which bundles exist: sets of objects which can only be obtained/delivered in combination;
5. which phenomena cause exchanges of value objects;
6. which partnerships of actors exist, jointly operating to the market;
7. which value creating and consuming activities exist;
8. to which actors are these activities assigned.

4.2.2 Process model decisions

A value model does only state *that* value-creating activities are carried out, not *how* these activities are put into operation. The latter is an important goal of process modeling. Other goals of process modeling are (Ould 1995, van Hee 1994): (1) creation of a common approach for work to be carried out, (2) incremental improvement of processes (e.g. efficiency), (3) support of processes by workflow management systems, and (4) analysis of properties of a process (e.g. deadlock free).

To present the *how*, a process model typically shows the following stakeholder decisions:

1. who are the *operational* actors involved, these can be different from the actor mentioned in section 4.2.1;

2. which *operational* activities can be distinguished to put value activities into operation;

3. which operational activities are executed by *which* actors;

4. what are the inputs and outputs of activities;

5. what is the sequence of activities to be carried out for a specific case;

6. which activities can be carried out in parallel for a specific case.

In sum, value models and process models clearly differ in the types of decisions they are able to support. Value models concentrate on the *what* aspect while process model show the *how* aspect. The importance of separating the *how* from the *what* concerns is anathema already for a long time in conceptual modeling, and it continues to be valid in value modeling as ever.

4.3 Different modeling constructs

The modeling constructs for value modeling have been discussed in chapter 3. In this section we summarize similar constructs in UML activity models, based on Rumbaugh et al. (1999). We see UML activity diagrams as a prototypical example of process modeling techniques.

Activity state. Central modeling construct in UML activity models is the activity state (shortly called an activity). It represents the execution of a statement in a procedure or the performance of an activity in a workflow. An activity is presented as a rounded rectangle with its name.

Transition. After some time, an activity completes. One or more other activities can then start. To this end, activities are related by transitions. By following the completion transitions connected to an activity, the next activities can be found. A transition is presented as an arrow.

Forks and joins. Two activities can be related by a simple transition. If an activity is completed, the other activity then starts. Also, activities can be related via forks and joins, for branching and unbranching, or synchronization. This is similar to the UCM AND/OR-constructs we have introduced in chapter 3.

Object flow. Activities can be related via transitions, but also via object flows. In such a case, an object is produced (output) by an activity, and is input for another activity. Objects flows relate value activities directly, or via forks and joins, similar to transitions. Object flows are visualized as dashed arrows, with the object superimposed.

Swimlanes. Activities can be assigned to performing actors by swimlanes. Swimlanes draw rectangles around activities and are given a name.

In sum, UML activity model concepts are centered around the notation of an *operational* activity, which represents work to be done, while e^3*-value* concepts are based on the notion of value. This results for instance in a different interpretation of *activity*; an e^3*-value* value activity assumes that an activity is profitable, and produces things of value for someone, whereas a UML activity represents work to be done by someone or something.

4.4 Different Universe of Discourse statements

A process model (e.g. a UML activity model) and an e^3*-value* value model make different statements on the Universe of Discourse. To exemplify the differences, we show a UML activity model and an e^3*-value* model for an e-commerce exploration track, we have carried out.

4.4.1 A contact ad service

An e-commerce idea. The e-commerce idea presented in this section is based on a real-life e-commerce project (see Gordijn et al. (2000*a*) and Gordijn et al. (2001)), and is about an e-contact service. The *Ad Association* is a company that coordinates about 200 local Free Ad Papers, shortly called FAPs. FAPs produce traditional, 'analogue' papers with ads. They are independent, often privately owned organizations, which are located around the world. A FAP serves a geographical region, for instance a large city or a county, because most goods offered in ads only reach a regional market. However, the *Ad Association* expects that *contact* *ads* may have a broader scope, even world-wide. Therefore, the *Ad Association* and the FAPs have decided to exploit their already locally known brand names to set up a contact ad service with a world-wide scope. Moreover, such a service will only be available as an Internet service; contact searchers can submit an ad using their browser, and can search in the ads database via their browser.

A value model. Figure 4.1 and figure 4.3 show a value model at various levels of abstraction. Note that this is only one of the possible value models. This value model shows that contact searchers, a number of FAPs and the *Ad Association* are involved. More importantly, it represents decisions regarding *who* is exchanging *what* with *whom* and expects *what* in return. For instance, figure 4.1 shows that a contact searcher is prepared to submit an ad, and expects a possible contact in return. A *possible contact* is someone who reads and reacts to the submitted ad. Also, the value model shows that a contact searcher who is reading an ad, must pay for doing so. Finally, ads are distributed by the *Ad Association*, who must pay a fee for obtaining ads, but receives a fee for delivering ads. Figure 4.3 shows a decomposition of the value activity *advertize ads* into three other activities: (1) trading ads, (2) checking ads, and (3) publishing ads. The activity *trading ads* refers to the commercial effort to obtain ads, and to resell these ads to parties interested in selling these ads to contact searchers. The *checking ads* activity checks an ad for correct and acceptable use of language. Some ads contain unacceptable phrases, which are rejected. Checking of such an ad can be done by the FAP who trades the ads, but also by another FAP. FAPs sometimes receive ads in foreign languages which they do not understand. They ask colleague FAPs then to check such ads, and pay for doing so. *Publish ads* is the activity that sells ads to contact searchers.

A process model. The process model is depicted in figure 4.2 and figure 4.4. Figure 4.2 shows main *operational* activities, as well as their performing actors. Strictly spoken, it is not an UML activity diagram (because it only shows communicating activities, and not the sequence of object flows), but such a diagram is often used to explain the overall process to actors involved. For instance, high level Petri-Nets (van Hee 1994) allow drawing such high level pictures. Figure 4.2 contains deliberately the same *names* for activities as we have used for the value model, but the objects flowing in and out activities differ. One difference is for instance a positive or negative confirmation by a FAP in reaction to an ad submission by a contact searcher. Figure 4.4 shows a more detailed process model, and focuses on the submission of ads. It decomposes parts of the activities mentioned in figure 4.2. Also, figure 4.4 presents the sequence of activities to be carried out for a submission of an ad.

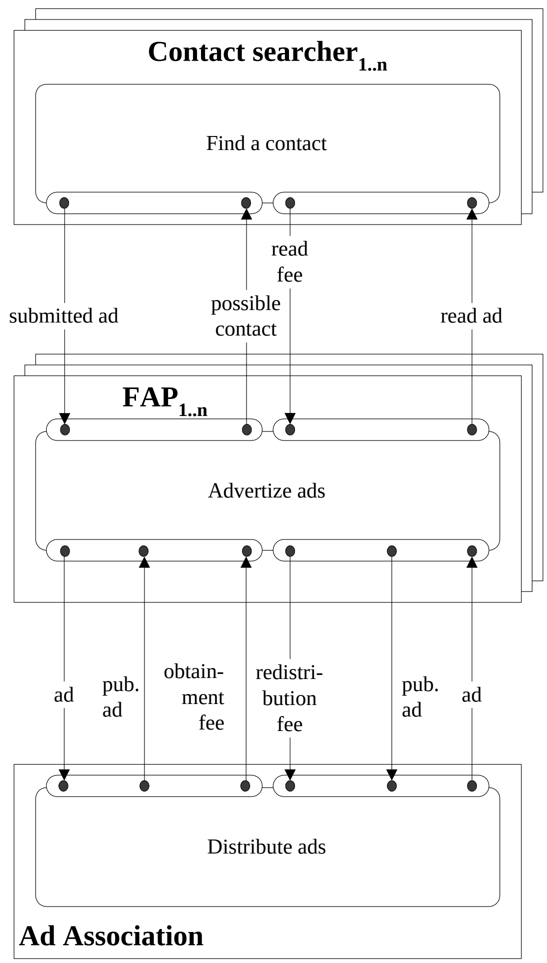


Figure 4.1: A top-level value model for the contact ad project.

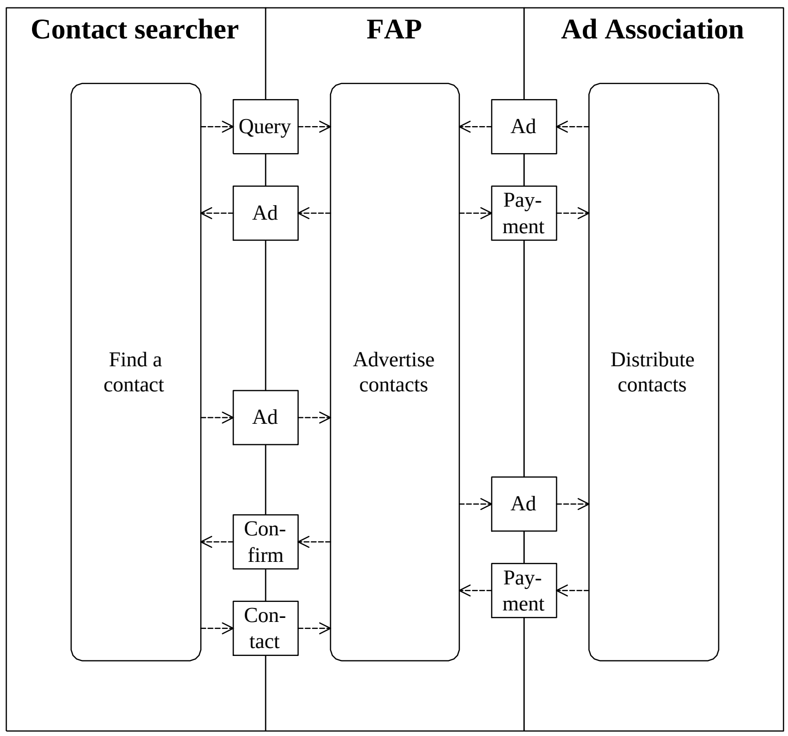


Figure 4.2: A top level activity model for the contact ad project.

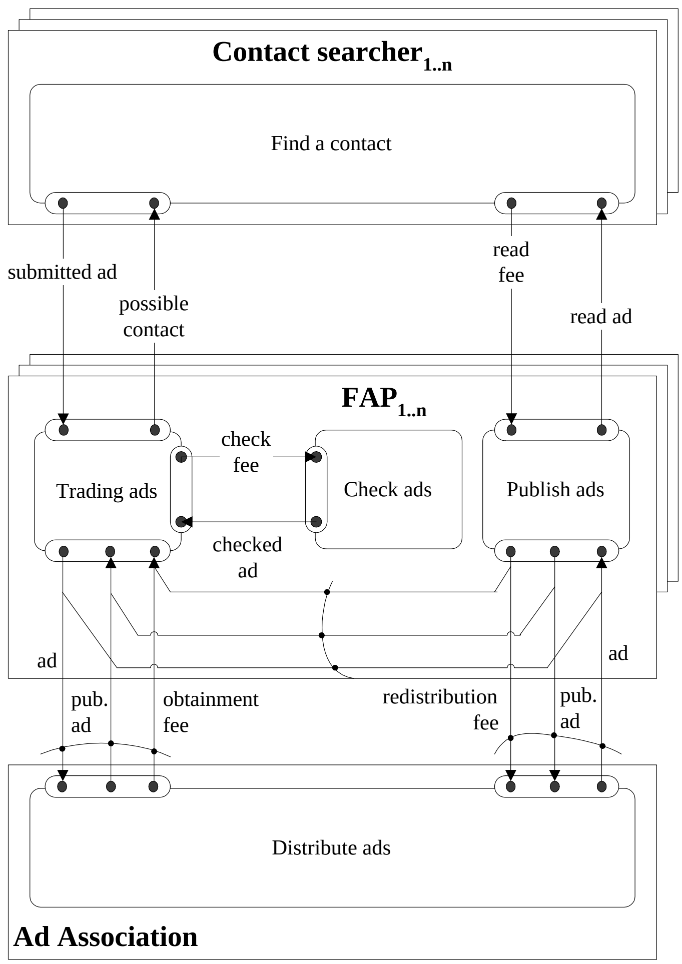


Figure 4.3: A decomposed value model for the contact ad project.

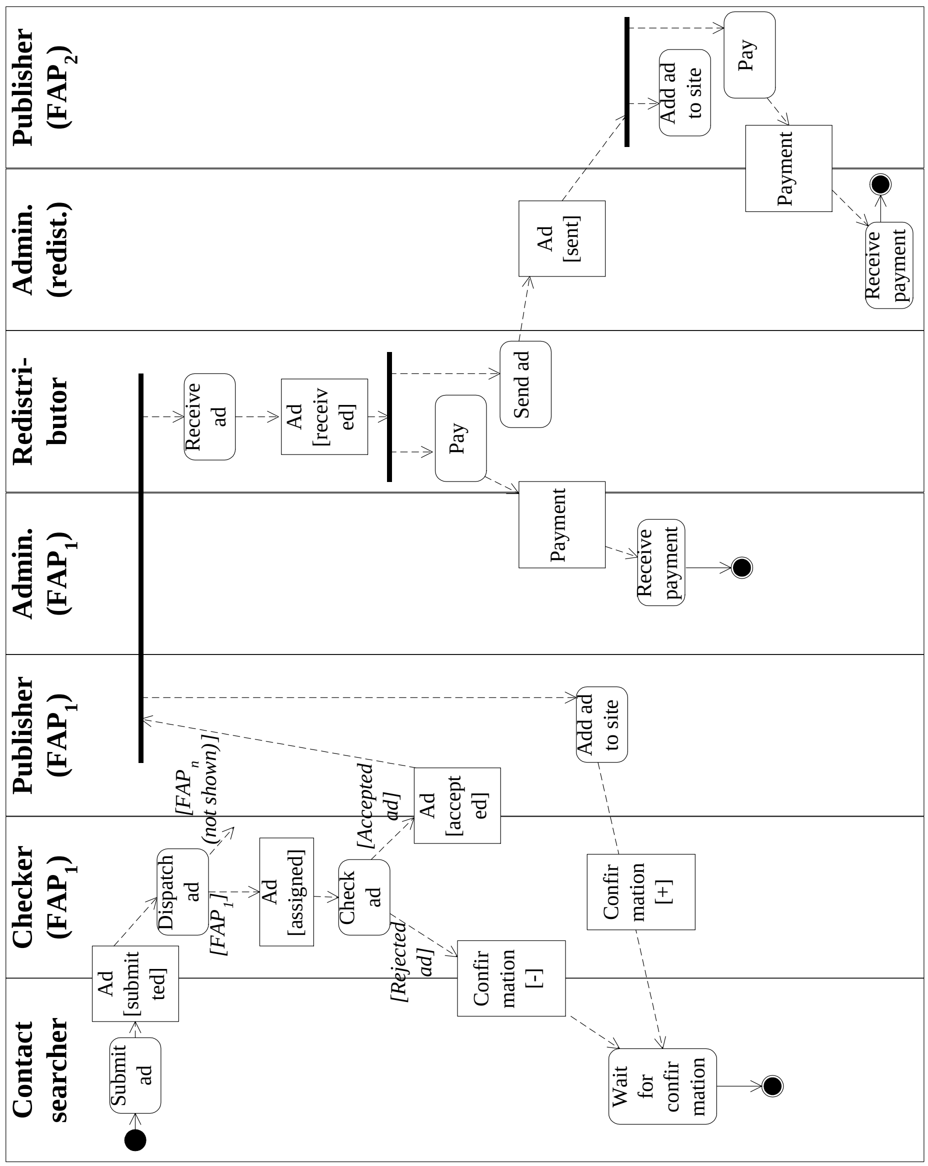


Figure 4.4: A decomposed process model for the contact ad project.

The following sections discuss in more detail differences between the value model and the UML activity model.

4.4.2 Value object and object

Value. In a value model, objects are only shown if they are of *economic value* to stakeholders. In a process model, objects are shown if they serve as required inputs of activities or are produced as outputs. As a consequence, not all objects that are part of a process model need to appear in a value model, because some objects may not be of direct value to someone; and a value model may identify objects that are not present in a process model.

Example: An object present in a process model, not present in a value model. The activity model (figure 4.2) shows an object *confirmation*, which models that a contact searcher receives a (positive or negative) confirmation after a submission of an ad. This *confirmation* object is not modeled in the value model (figure 4.1), because it is not of direct value to the contact searcher. It is only needed as control information, for instance to trigger the contact searcher to re-submit his/her ad after rejection.

Example: An object in a value model, not present in a process model. In the activity model (figure 4.2) a *possible contact* is not present, because there is no direct corresponding physical or information object flowing from the FAP to the contact searcher. A possible contact states a consumer experience of the contact searcher; namely that s/he found a contact s/he possibly likes. As a valuable consumer experience, it is present in the value model (figure 4.1).

Object properties. Different subsets of object properties are identified for value models and process models. A value model identifies those object properties, which can be used by a stakeholder to determine the *economic value* of the object, whereas object properties in process models can be used by an activity to determine a *state transition*.

Example: State transition property and value property. The *publishing date* of an ad on a website is a property useful in a process model, because it can be used to determine a state transition; from an invisible ad to a visible ad. This property, however, is not very useful in determining the *economic value* of an ad. Because the value model in figure 4.1 states that a contact searcher has to pay for reading an ad, an interesting value property for the searcher is the *likelihood* that an ad contains a contact the searcher is interested in (e.g. based on the searcher's personal profile). Such a property partly determines the value a searcher assigns to an ad read.

In sum, objects themselves as well as the kind of object properties differ between a value model and a process model. In a value model objects need to yield value to someone, while in a process model objects serve as inputs and outputs for activities. In a value model, properties of an object should be usable for valuing the object by an actor, whereas in an activity model properties can be used to determine a state transition.

4.4.3 Value exchange and control/object flow

Transfer of ownership. Objects of value are exchanged between actors/value activities through *value exchanges*. The goal of such a construct is to model a transfer of economically valuable objects from one actor to another actor. In a process model object flows are used to model which objects are output and input for activities, and model a transition from one activity to another. It is used to express *how* activities should be carried out in terms of sequences or parallelization of activities in case control and object flows collapse.

Example: Flows and experiences. The process model in figure 4.2 contains an object flow from the contact searcher to the FAP called *contact* that states that the contact searcher reports the experience of a desired contact to the FAP. This flow relates to the *possible contact* value exchange (figure 4.1) from the FAP to the contact searcher, but it is not the same. The *contact* flow is necessary as *control* information, for instance as a trigger to remove a published ad as soon as a contact occurs, while the *desired contact* value exchange models the *valuable experience* itself. Note that in the process model, the confirmation flows from contact searcher to FAP, while the possible contact flows from the FAP to the contact searcher.

No direct physical or information flow. A value exchange may coincide with a flow of a physical product or information if these are of value to a stakeholder. However, sometimes a value exchange states a consumer experience, which has no underlying direct physical or information flow. The previous example illustrates this case also.

In conclusion, a value exchange expresses a change of ownership (as an economic result, not as a process outcome), which is normally not expressed in process models. Moreover, some value exchanges do not imply a physical or information flow directly, but instead express an actors' consumer experience.

4.4.4 Value interface and value offering

In a value model, we have the notion of *value interface* expressing the principle of economic reciprocity (a rule or law of value exchange). This allows stakeholders to clarify to each other *what* objects of value they are prepared to exchange in return for other objects; a key decision during value modeling. Similarly, we have the notion of *value offering* denoting that objects are requested or delivered only in combination. Both these principle are not present in process models.

Example: Economic reciprocity. From figure 4.2 it cannot easily be concluded that a reader has to pay for reading an ad, while figure 4.1 clearly shows that a *read ad* is offered in return for a *payment*.

4.4.5 Actors

Individual actors. In a process model, the actor itself is usually not shown at the instance level. At most it is indicated that a number of actors capable of performing a particular activity, should be present, for instance to model resource management. When designing value models, it should be possible to identify the profitability of a value model for a particular actor. During value modeling, these individual actors are important stakeholders. Therefore, in a value model, actors sometimes are mentioned on an individual basis.

Operational actors and profitable actors. Actors in an process model are indicated for purposes such as resource allocation and scheduling. They perform an operation. However, in a value model we distinguish actors to facilitate reasoning about *profitability*. Therefore, actors are not individual agents performing activities, but economic and legal entities that engage in business transactions.

Example: Operational actors and profitable actors. In figure 4.3, actors performing activities are represented by swimlanes. The actual actor instances are not mentioned, while the value model (figure 4.1) indicates the existence of a number of FAPs which can be addressed on an individual basis. Moreover, in the value model we distinguish FAPs, being legal entities that engage in business transactions, whereas in the process model we identify resources carrying out work for such an entity, such as a checker, a publisher, a redistributor, and an administration officer (figure 4.4).

4.5 Different ways of decomposing activities

Value activities in a value model differ from activities in a process model. This leads to different ways of decompositions.

Value activity and operational activity. In process modeling, an operational activity denotes something to be done, in order to produce outputs as a result of inputs and resources. In a value model, we distinguish activities only if they are *profitable* for the performing stakeholder.

Decomposition. A different interpretation of the *activity* concept in value models and process models leads to different decompositions. In literature on process modeling, a number of motivations are given for the decomposition of activities into sub-activities. IDEF_0 (*$IDEF_0$ Method Report* 1981) indicates that an activity should be recursively decomposed in 5 to 7 sub-activities, until a common understanding about the activity is reached by stakeholders. In this case, decomposition serves the goal of *clarity* ro reach common understanding. In STRIM (Ould 1995), activities are decomposed until they can be regrouped and assigned to a particular role (i.e., operational actor). Decomposition then serves the goal of clarifying resources needed in carrying out tasks. In a value model, however, we only decompose a value activity if *all* resulting sub-activities themselves are profitable. In Porter & Millar (1985) and Timmers (1999), this is referred to as value chain deconstruction, as a way to discover new activities which can be successfully assigned to alternative commercial actors (see also chapter 6).

Example: Different decompositions. Activities in figure 4.2 can be decomposed into smaller operational activities (see figure 4.4). We focus on the submission of an ad. The purpose of figure 4.2 is to illustrate the main operational activities and objects flows between these to stakeholders, whereas the main goal of figure 4.4 is to explain how a submission process should be carried out and by whom. Also, this more detailed activity diagram shows the sequence of activities to be performed, which we do not represent with value models.

After an ad is submitted by a contact searcher, it is dispatched to a FAP who can read and check the ad (e.g. for absence of dirty language). In the meantime, the contact searcher waits for a confirmation. If the ad passes the check, it is added to the website of a FAP and the contact searcher receives a positive confirmation. Otherwise, if the ad is rejected, the contact searcher receives a negative confirmation. If the ad is accepted, it is offered to the *Ad Association*, who pays for it afterwards. The *Ad Association* supplies the ad to other FAPs. Also, the ad is published on the FAPs own website. Activities such as ad dispatching, doing and receiving payment, as well as waiting for a confirmation, are important to distinguish because they show how a process should be carried out, in which order activities must be performed, and who must sent confirmations, but are not shown in a value model, because these activities create no profit directly.

From a value perspective, figure 4.3 shows a decomposition of the value activities in figure 4.1 into *profitable* sub-activities. The purpose of decomposition here is to find new profitable activities. The decomposition operation is defined as follows: (1) a value activity can be decomposed in other (sub) value activities if each sub-value activity is profitable for one or more performing actors, and (2) consider for each pair of sub-value activities new value interfaces and value exchanges if required.

The *advertize contacts* value activity is decomposed into three sub-value activities, which are assumed to be profitable. Note that between these sub-value activities new value exchanges have been introduced.

In sum, in a process model, decomposition is often led by the motivation to show a process flow in detail or to discuss assignment of operational activities to operational actors, while in a value model it is led by a search for commercially viable sub-activities.

4.6 Conclusions

e-Commerce value modeling and process modeling are both forms of conceptual modeling, both are necessary for good e-commerce design, but they differ in several significant ways. First of all, the main goal of value modeling is to reach agreement amongst stakeholders regarding the question '*who* is offering *what of value* to *whom* and expects *what of value* in return'. In contrast, an important goal of process modeling is to reach a common understanding about *how* activities should be carried out (e.g. in which order). Also, a value model discusses who is doing what to make profit, while a process model models who is doing what for allocation of operational resources. These are different modeling goals, asking for different modeling methods with different constructs. Modeling strategic intent of e-commerce differs from modeling operational fulfillment.

As a result, the contents of a value model and a process model also differ in a number of ways:

1. the concepts in value modeling are centered around the notion of *value*, while

in process modeling concepts focus on *how* a process should be carried out in operational terms;

2. in a value model, an actor is profitable, whereas in a process model an actor performs an operational process;

3. in a value model, objects represent something of value to a stakeholder, while in a process model objects serve as inputs and outputs for activities and may be used to steer the process flow;

4. in a value model, object properties can be used by a stakeholder to determine the economic value of an object. In a process model, object properties are used to determine state transitions.

5. In a value model, value exchanges represent a transfer of ownership, while in a process model a flow of information or goods implies a change of state.

6. in a value model, we have the notion of economic reciprocity, which is conceptualized by the value interface. Such a notion is absent in process modeling. A similar construct, the value offering, is also not present in process modeling.

7. in a value model, we are only interested in activities which are profitable. Decomposition of such activities is done to discover smaller chunks of activities that still are profitable. Discovering these activities often leads to re-assignment of activities to actors. In a process model, decomposition serves the goal of clarification of the workflow or to show the assignment of activities to working actors. Hence, the model decomposition rules are different.

Chapter 5

From an e-commerce idea to a value model

In this chapter we present how to explore an e-commerce idea by developing one or more value models. We do so by giving a *prototypical* approach, consisting of steps and guidelines. The aim of sketching such a prototypical approach is to facilitate an inexperienced user of e^3-*value* . More experienced users will skip steps, or do them in parallel, or use a different sequence, depending on the idea explored and the context in which exploration takes place.

We focus our discussion on the development of the value viewpoint; how to develop a process and information system viewpoint is exemplified in chapter 9. Section 5.1 provides an overview of our value-focused exploration process. It consists of (1) having an e-commerce idea (section 5.2), (2) construction of one or more value models, that is capturing an e-commerce idea (section 5.3), (3) deconstruction and reconstruction of value models; elicition of variations on the earlier found value models (section 5.4), (4) exploration of other viewpoints (section 5.5), and (5) evaluation of the e-commerce idea (section 5.6). Finally, section 5.7 presents our conclusions

5.1 Exploration process overview

Figure 5.1 presents the main steps we carry out to explore an innovative e-commerce idea from a value modeling perspective.

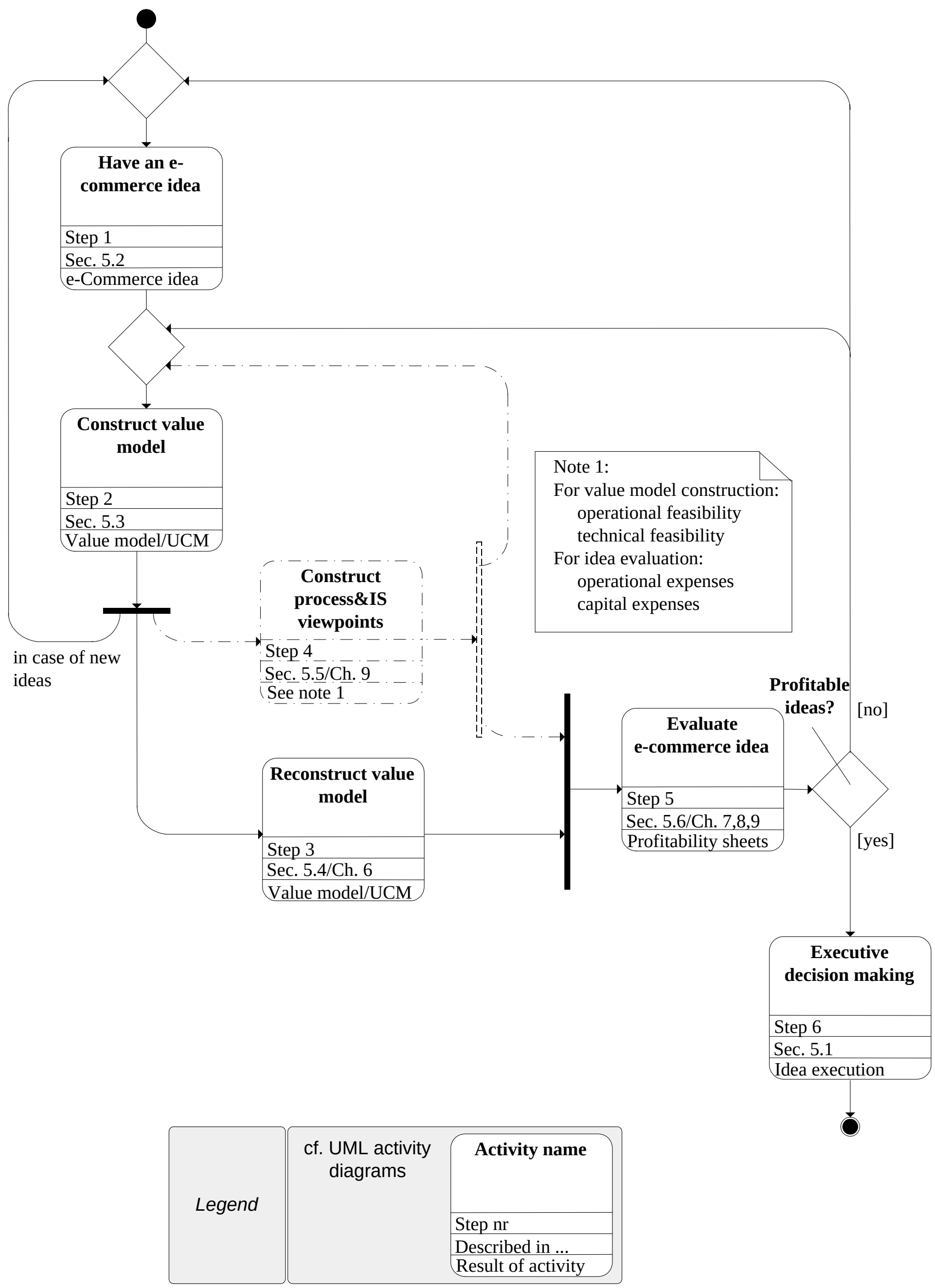


Figure 5.1: Exploring an e-commerce business idea: a value viewpoint perspective.

Step 1: *Have an innovative e-commerce idea.*
Our approach supposes an existing innovative e-commerce idea. The scope of this chapter is how come from such an idea to one or more value models, so section 5.2 discusses this step only briefly.

Step 2: *Construct a value model and set up a baseline.*
An e-commerce idea is used to construct a value model, which explains the idea by stating the actors involved, and the objects of value created, distributed and consumed by these actors. It serves as a baseline for finding alternatives as well as for evaluation. Value models are expressed using e^3-*value* concepts and scenario paths (see chapter 3). How to construct such a model is the topic of section 5.3.

Step 3: *Deconstruct and reconstruct a value model: find variations.*
After the articulation of an e-commerce idea by one or more value models, other value models are searched for. We do so by value model deconstruction and reconstruction. Deconstruction splits a value model into smaller parts, and reconstruction composes these parts in different ways (see section 5.4 and chapter 6 for a more detailed discussion).

Step 4: *Develop other viewpoints: process viewpoint and information system viewpoint.*
As we argued in chapter 2, to explore an e-commerce idea, a number of viewpoints have to be explored. The focus of this chapter is the exploration of the value viewpoint. Consequently, section 5.5 discusses exploration of other viewpoints only in relation to the value viewpoint.

Step 5: *Evaluate an e-commerce idea: is the idea profitable?*
Evaluation in the context of our approach means that we assess potential profitability of the e-commerce idea at hand. Moreover, we investigate profitability sensitivity of the idea for future events, to increase confidence in the sustainebility of the e-commerce idea. Evaluation is presented in section 5.6. Finally, if e-commerce ideas are found in which actors may have sufficient confidence, step 6, executive decision making takes place.

5.2 Have an innovative e-commerce idea

An exploration track starts with a vaguely articulated e-commerce idea. This idea is typically formulated by one or two sentences. We assume that this idea exists already in the mind of stakeholders. How to find and create such an idea is outside the scope of our research. Therefore, our approach must not be seen as a recipe to find new e-commerce ideas, but rather as an approach to explore, clarify and evaluate such ideas, as well as to find variations.

Although we assume the existence of an idea, it is our experience that during construction of a value model, stakeholders find other, new, e-commerce ideas themselves. This is a side effect of discussions between stakeholders to create a value model for the e-commerce started with. Figure 5.1 presents this effect by showing a feedback from the activity comprising the construction of a value model, to having the e-commerce ideas.

5.3 Construct a value model

This section shows how we construct a value model by presenting steps to be executed as well as guidelines used (see also figure 5.2). We distinguish the following steps: (1) identify scenarios (section 5.3.1), (2) identify actors (section 5.3.2), (3) decide on an actor versus market oriented approach (section 5.3.3), (4) identify value objects/ports and value offerings/interfaces (section 5.3.4), (5) identify value exchanges (section 5.3.5), and (6) identify scenario maps and paths (section 5.3.6).

5.3.1 Identify scenarios

Value model construction starts with identification of scenarios. Scenarios are at this point short sentences, denoting the product, service, or experience desired by a *customer* (see also guideline 2-2). It is our experience that it is hard to find these scenarios and to articulate them well in a first step. However, as can be seen from figure 5.2, construction of value model is a cyclic process. It is our experience that after a number of cycles, stakeholders can define scenarios more accurately.

Guideline 2.1: *Use fragments of the e-commerce idea, which refer to products or services to find scenarios.*
Explanation. Scenarios are elicited by using the e-commerce idea as a starting point. This idea should contain fragments of or indications to product/services

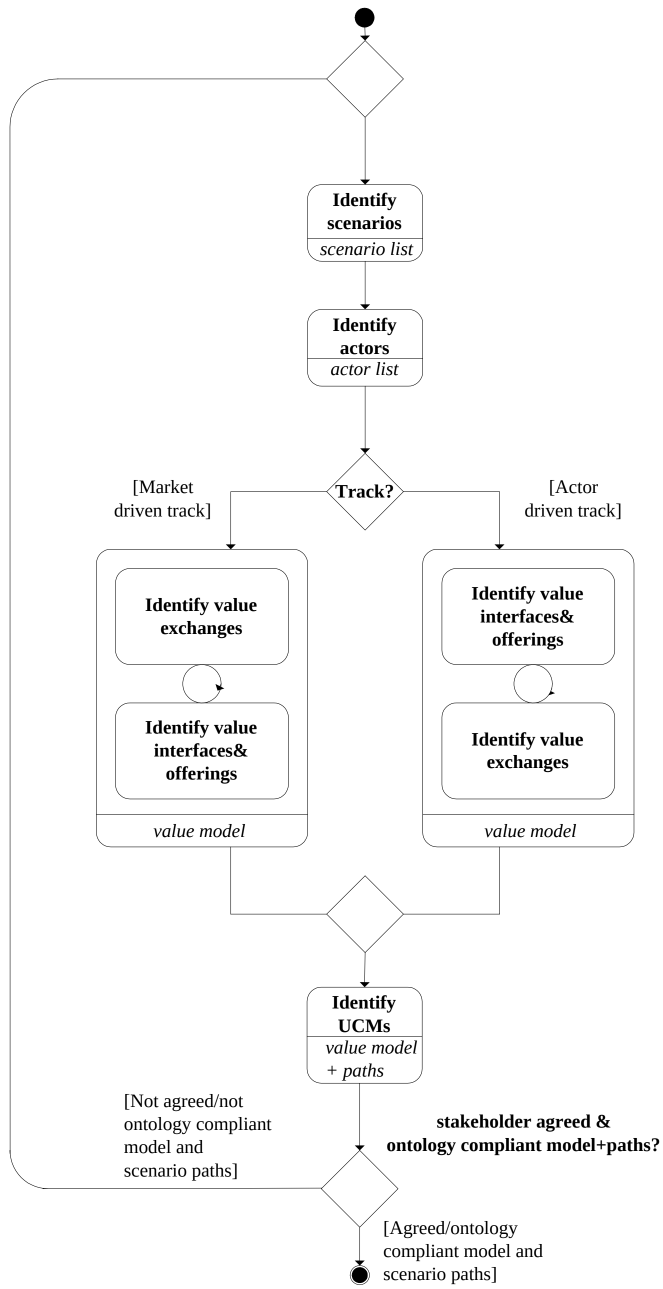


Figure 5.2: Construction of a value model.

wanted by someone. Typical examples are: *read an ad*, *place an ad* (based on the *Ad Association* project, see chapter 4, 9), and *surf on the net* (see chapter 3).

Guideline 2.2: *Formulate scenarios by taking products or servives wanted by a customer.*
Explanation. By asking actors to formulate a scenario by taking a customer perspective, we increase the chance that products and services are really wanted by them. It is our experience that many stakeholders have products or services in mind they want themselves, rather than those wanted by their customer. A similar approach is also suggested by Tapscott et al. (2000).

5.3.2 Identify actors

A list of actors is created, initially based on the actors initiating the idea, and the (end)-consumers they have in mind. After a number of cycles, some actors have been removed or added to this list. Actors are mentioned by listening their company name (e.g. *Ad Association*), or in the case of end-consumers by the role they play (e.g. *contact searchers*).

Guideline 2.3: *Deal with yet unknown actors by distinguishing identified and non-identified actors.*
Explanation. It sometimes occurs that it is known that a specific kind of actor is needed, who is not yet explicitly identified by name. This is for instance the case if some specific actors decide to explore an e-commerce idea, and discover they can not put into operation the idea solely by themselves. To this end, we distinguish two kinds of actors:

1. *identified actors*, such as named companies, who are known and consequently are identified by name;

2. *non-identified* actors, which are necessary for the value model, but yet unknown.

A similar distinction has also been made by Ould (1995) who distinguishes an *actor* (e.g. George Bush) from a *role instance* (e.g. the president of the United States).

Guideline 2.4: *Use* environmental *actors for actors which are needed to let the value model work but which are not of interest for profitability analysis.*
Explanation. Many modeling techniques have the notion of an *environment* of a model. A value model may also have an environment. The environment of a value model consists of actors (or value activities) we are not interested in from a profitability perspective, but who are needed to let the value model work. Such an environmental actor is only shown because another actor, who is part of the value model, must be able to obtain his/her value objects from someone.

5.3.3 Decide on an actor versus market oriented approach

After the actors are known, the next step is to state what actors are producing, distributing and consuming, and to identify what they want in return for objects they deliver. We distinguish two approaches for doing so: (1) the actor driven track, and (2) the market driven track.

The *actor driven* track starts with *one key actor* in the e-commerce idea, identifies the actor's offerings to and from his/her environment, and related concepts such as value interfaces, value ports and objects. Hereafter, value exchanges with other actors are identified.

In contrast, the *market driven* tracks starts with the *overall picture* of an e-commerce idea. First the value exchanges which should exist in the overall actor network are identified, as well as the objects exchanged. These exchanges are used to derive the individual actor's value interfaces, offerings, and ports.

It is our experience that in both tracks, identification of interfaces, offerings and ports on the one hand, and identification of exchanges on the other hand are heavily interrelated. This is depicted in figure 5.2 by a circle between value exchange- and value interface/value offering identification.

Guideline 2.5: *Follow the actor track if one key actor is involved, otherwise follow the market track.*
Explanation. We choose for the actor driven track if an e-commerce idea is initiated by one key actor. A market driven track is useful if the e-commerce idea is initiated by a number of actors, who act as consortium in exploring and implementing an e-commerce idea. In such a situation, the e-commerce idea can not be pinpointed to a single actor.

5.3.4 Identify value objects/ports and value offerings/interfaces

A value interface consists of one or two offerings. In turn, an offering contains ports which offer or request value objects, depending on the port's direction. For each actor, all these constructs have to be identified. To do so, we have the following steps:

1. identify value ports and objects exchanged by ports;
2. group value ports into value offerings;
3. group value offerings into value interfaces.

Identify value objects and ports

The way of identifying objects and ports depends on whether the actor- or market oriented approach has been chosen. In case of a market oriented approach value exchanges as well as the objects they exchange have already been identified. Ports are then the end-points of the value exchanges.

In contrast, by following an actor oriented track, one starts with searching value objects offered to, or requested by a key actor via his/her value ports. Ports are closely related to value objects: once an object offered or requested is known, a port should be identified for doing so.

We use a number of guidelines to find value objects and ports: (1) the e-commerce idea and scenarios should trigger value objects, (2) actors want something in return for value objects they offer (economic reciprocity), and (3) actors need to obtain other value objects to offer a value object themselves (causally related value objects).

Guideline 2.6: *Use products and services mentioned in the e-commerce idea and scenarios to find value objects.*
Explanation. The *e-commerce idea* and *scenarios* should trigger identification of value objects. If a scenario does not provide any ground for value objects, the scenario is likely not defined in terms of customers, but perhaps defined in terms of operational business processes.

Guideline 2.7: *Use the economic reciprocity property to find value objects.*
Explanation. A second guideline we use is to ask actors which value object(s) they want return for an already identified value object they offer. We call such value objects shortly *reciprocal* value objects. It is our experience that in nearly every situation reciprocal value objects can be found.

Guideline 2.8: *Use causally related value objects to find value objects.*
Explanation. Thirdly, we search for *causally related* value objects. To be able to offer a value object to his/her environment, it is likely that an actor must obtain one or more other objects, which we call causally related value objects. This is for instance the case for a trading company. Objects which are sold must also be bought.

Guideline 2.9: *How to determine if something is a value object?* A value object must be of economic value for at least one of the two actors exchanging the object.
Explanation. The criterion used for distinguishing value objects is that a value object must be of economic value for *at least* one actor. Following this formulation a value object needs not to be of value for both actors exchanging the object. This is motivated by the observation that valuation of objects depends largely on an individual actor (Holbrook 1999), and consequently not both actors have to assign economic value to an object.

Guideline 2.10: *How to determine the direction of ports?* The direction models the direction into which ownership will be transfered, or to whom rights are granted.
Explanation. Each value object delivered or requested by an actor results in a port for doing so. For such a port, the direction has to be determined. The criterion to decide whether a port has a direction *in* or *out* is to assess whether an actor will obtain (in-port) or loose ownership (out-port) once the object has been exchanged. For service oriented objects, the criterion is the grant (in-port) of the right to receive the service, or the obligation (out-port) to deliver the service.

Group ports into value offerings

We have in our e^3-*value* ontology two mechanisms for grouping value ports. The value offering is used to group equally directed ports, e.g. for showing *mixed* *bundling*, while the value interface is used to model the notion of *economic reciprocity* (to be discussed in the next section). In case of a value offering, different motivations apply for grouping in- and out-ports.

Guideline 2.11: *If value objects obtained via in-ports are only of value for an actor* in combination*, then group the in-ports into an offering.*
Explanation. In-going ports are grouped into an offering to express that an actor only assigns economic value to objects if they come *in combination*. This is exemplified in chapter 3: a surfer who wants to access the Internet, must obtain an Internet access connection from an Internet Service Provider (ISP) *and* must obtain a telephone connection between him/herself and the ISP for data transport.

Guideline 2.12: *If value objects offered via out-ports are only available* in combination*, e.g. as a result of mixed bundling or cost-effects, then group the out ports into an offering.*
Explanation. There can be several reasons to group out-going objects into one offering, rather than to offer these objects separately. Here we distinguish two considerations which can used as a guideline: (1) to model mixed bundling, and (2) to model cost avoidance.

Mixed bundling is a way to increase total profit for a supplier of objects. An actor then supposes that different products sold in combination yield more profit than that if they were sold separately (Choi et al. 1997). Suppose there are three customers (*Alice*, *Bob*, and *Charlie*), and two products *X* and *Y* offered by a supplier *S*. Product *X* is valued Euro 40, 50, and 60 by *Alice*, *Bob* and *Charlie* respectively. Product *Y* is valued the same, but in the reverse order of *Charlie*, *Bob*, and *Alice*. Also suppose that both products cost supplier *S* Euro 40, and that they are sold for Euro 50. In this situation, *X* is sold to *Bob* and *Charlie*, while product *Y* is sold to *Alice* and *Bob*. Total profit for *S* is Euro 40. As an alternative, supplier *S* may also consider to sell *X* and *Y* only in combination for Euro 100. In such a case, *Alice*, *Bob* and *Charlie* buy *X* and *Y*. Total profit is then Euro 60.

A second reason for grouping out-going ports is that some objects can only be *cost-effectively* offered in a bundle rather than separately. This is case in the project outlined in chapter 8, which is about a service of a newspaper offering its archive of news articles to its subscribers via the Internet also. In this e-commerce idea, the entire operation of Internet service provisioning (telephone connections, Internet access and web hosting) is outsourced to another party, a Telco. This Telco is capable of offering connectivity, access and hosting for a low fee, if all equipment is co-located at a telephone switch. In contrast, if e.g. the Internet access servers

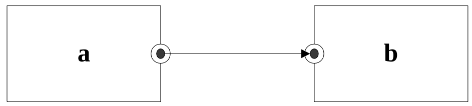


Figure 5.3: Value interfaces with each having an offering containing only one port.

and the telephone switch are hosted at a different location, a higher fee is asked. In this case, grouping is used to express that, for a low fee, the value objects denoting connectivity, Internet access and hosting can be obtained, but only in combination.

Group value offerings into value interfaces

Value interfaces are used to model the notion of economic reciprocity, a guideline we also use to find value objects and ports. For each port of an actor, which is part of an offering, other port(s) of opposite direction are searched for, which compensate for objects exchanged via the first port. The offerings which contain these ports are grouped into a value interface. To find value interfaces we use the following guidelines: (1) a value interface consists of two opposite offerings, and (2) causally related offerings are *not* grouped into a value interface.

Guideline 2.13: *A value interface should consist of two reciprocal offerings.*
Explanation. It is our experience that in nearly all cases, a value interface consists of two opposite directed offerings. The direction of an offering is equal to the direction of its ports. The reason for this guideline is that a rational actor only is willing to exchange an object o_{out}, if s/he obtains another object o_{in} in return. Moreover, s/he must assign to object o_{in} a higher economic value than to object o_{out}.

However, we did not formalize this rule in our e^3-*value* ontology (see chapter 3). The reason for this is that we can think of cases where the act of exchanging objects between actors is positively valued by both actors involved. In figure 5.3 actor *a* assigns value to delivering an object of value (this is e.g. the case if the object is waste from an actor *a* perspective), while actor *b* assigns value to obtaining the object (waste of someone else can be a resource for another party). However, in real-life projects we did not encounter such a situation. Therefore, if a value interface consists of only one offering, this is an indication for a yet undiscovered value object and port, and a motivation to redo identification of value ports and objects.

Guideline 2.14: *Never group causally related offerings.*
Explanation. We do *not* group value offerings which are causally related. Two offerings are causally related, if a port in the first offering is causally related to a port in the second offering. Two ports are causally related if, in order to produce a value object o_{out} by a port, a value object o_{in} must be obtained by the other port. An actor does so by performing a value activity: s/he adds value to object o_{in}, resulting in object o_{out}. Note that the direction of causally ports differs: the port offering object o_{out} is an out-port, while the port requesting object o_{in} is an in-port. We do not group these ports into one value interface, because the value interface is a construct that shows which objects are offered, and which objects are requested, *as a compensation*, in return. Instead, the causal relation between in- and out-ports is represented using a scenario path. Such a path shows which exchanges on value interfaces cause exchanges on other value interfaces.

5.3.5 Identify value exchanges

A market oriented track starts with the identification of value exchanges rather than ports. The difference between both tracks is that during the actor oriented track, we ask for a specific actor what s/he offers and request *to and from his/her environment* (other actors), while during the market oriented track, we ask a number of actors (in many cases two or three actors), what they offer *each other*.

Guideline 2.15: *Use guidelines 2.6-2.9 also for identification of value exchanges.*
Explanation. The aforementioned guidelines for finding ports following an actor oriented track can also be used to find value exchanges by using a market oriented track. Already identified *scenarios* provide a starting point for finding value exchanges (guideline 2.6). Also *reciprocal* value exchanges, similar to reciprocal value ports can be identified (guideline 2.7). Note that if an actor a_1 offers a value object to some other actor a_2, actor a_1 needs not to be compensated by the same actor a_2. As a third guideline, already identified value exchanges can be used to find *causally* related value exchanges, in the same way as we identify causally related value ports (guideline 2.8). Finally, value objects are only modeled if they are of economic value for at least one actor (guideline 2.9).

5.3.6 Identify scenario maps and paths

A scenario is modeled using one or more scenario paths. Scenario paths show which value objects need to be exchanged via actors' interfaces as a result of the execution of a scenario. As such, scenarios paths are traces through a use case map. To identify scenario paths, we first have to construct one or more use case maps on top of the value model, and hereafter we have to identify the paths through such maps (see figure 5.4).

Identify use case maps

Essentially, use case maps are developed by taking a start stimulus and finding value exchanges an actor must do, to fulfil needs expressed by such a stimulus. In doing so, we distinguish the following steps (see figure 5.4):

1. identification of the start stimuli themselves;
2. identification of parts of a use case map *within* an actor. Such a partial map models via which value interfaces an actor must exchange value objects as a result of: (1) a start stimulus, or (2) the exchange of value objects via one of his/her *other* value interfaces. In the first case, the partial map connects a start stimulus with one or more responsibility points touching value interfaces of the same actor, in the second case, the partial map connects responsibility points touching value interfaces of the same actor.
3. identification of parts of a use case map *between* actors. This partial map models which value exchanges (via which interfaces) must occur between actors, if one actor decides to exchange value objects.
4. identification of a stop stimulus. If an actor exchanges value objects via one of his/her value interfaces, s/he may need to exchange other value objects, *or* the scenario may end.

Guideline 2.16: *Base start-stimuli on end-customer needs.*
Explanation. A scenario description relies on customer needs. Some of the customers are *end-consumers*: they buy a product or service for consumption and do not re-sell it anymore. Such customers often cause start-stimuli, which cause a cascade of value exchanges.

Guideline 2.17: *If an actor can* choose *from more than one of his/her value interfaces to satisfy his/her needs caused by a stimulus or exchanges via another value interface, then use an OR element to connect the stimulus/responsibility points touching these interfaces.*

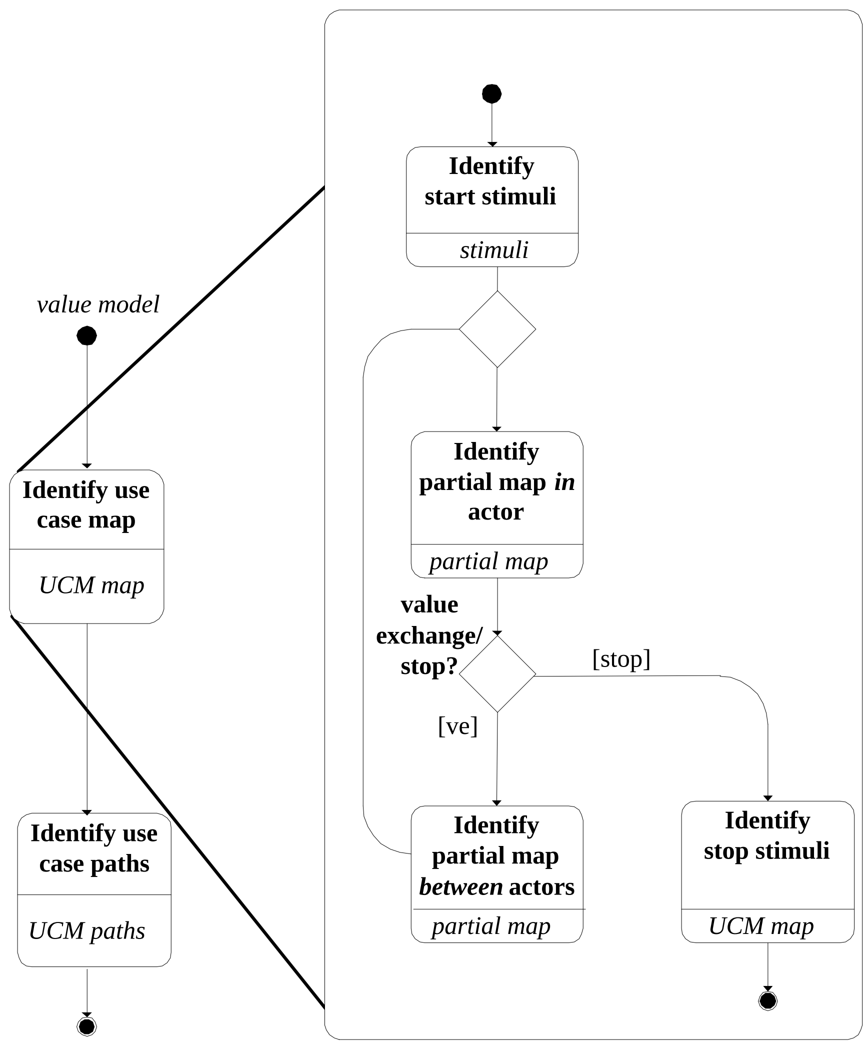


Figure 5.4: Identification of use case maps and paths.

Explanation. An actor will exchange value objects via one of his/her interfaces as a result of: (1) a start stimulus, or (2) exchanges via another value interface of the same actor, shown by a responsibility element touching such a value interface. Each of these cause that an actor exchanges objects via one of his/her other value interfaces to satisfy his/her needs (or a stop stimulus occurs). To do so, it can be the case that an actor may choose from *alternative* value interfaces s/he has. An OR fork is then needed to connect the stimulus/responsibility element with responsibility elements touching all these alternative value interfaces, modeling that different continuations along different paths of the scenario are possible. Note that introducing an OR fork results in different scenario paths for the same scenario.

It can also be the case that exchanges via a value interface can be caused by exchanges via other, alternative, value interfaces of the same actor, or by alternative start stimuli. An OR join is then needed to connect the start stimuli/responsibility points of these alternative value interfaces with the responsibility point of the first mentioned value interface.

Guideline 2.18: *If an actor must use* multiple *value interfaces of his/herself to satisfy his/her needs caused by a stimulus or exchanges via another value interface, then use an AND construct to connect the stimulus/responsibility points touching these interfaces.*
Explanation. It occurs that, to satisfy a need, *multiple* objects which are obtained via different value interfaces of a same actor, are required. In such a case, an AND fork is needed to connect the stimulus/responsibility elements with the responsibility elements touching the different value interfaces satisfying the need.

Similarly, it can be the case that as result of exchanging value objects via more than one value interface, an actor will exchange value objects via only one other interface. In such a case, an AND join must be used to connect the responsibility points touching the value interfaces.

Guideline 2.19: *If value ports of two value interfaces of different actors exchange value objects with each other, and all these ports are connected with each other by value exchanges, then connect the responsibility points touching these interfaces.*
Explanation. In many situations actors exchange objects of value with each other on a biliteral basis. Then two actors, each with one value interface, are connected by relating all value ports of these interfaces by value exchanges. In such a case, the two responsibility points touching the value interfaces must be connected.

Guideline 2.20: *If three or more value interfaces of different actors exchange value objects, then connect the responsibility points touching the interfaces with an AND construct.*
Explanation. It is possible that the exchange of value objects via an actor's interface results in exchanges with more actors and consequently with more value interfaces. For instance, in figure 3.14 a surfer needs to exchange value objects with both a local operator and a free Internet service provider. In such a case, an AND fork is needed on the scenario path to denote that the scenario path forks into to sub paths which each must be executed. Also, exchanges via multiple value interfaces may result into exchanges via one other value interface. In such a case, an AND join is needed

Guideline 2.21: *If an actor can choose from a number of value interfaces of other actors to satisfy his/her needs, then use an OR construct to connect responsibility points touching the value interfaces.*
Explanation. An actor's value ports in a particular value interface may be connected to multiple other value ports of different actors (see figure 3.3). This models that an actor can choose from a number of actors to fulfil his/her needs. This should also be reflected in the use case map by adding OR elements.

Identify paths

Different paths in a case map exist if OR constructs have been used. Note that OR constructs result in multiple routes, while AND constructs only introduce sub paths, which all are executed.

Guideline 2.22: *Find scenario paths by focusing on OR constructs in a use case map.*
Explanation. Paths can be identified by 'executing' the scenario by starting at the start stimulus, and traversing through the map. Each time an OR construct is encountered, multiple scenario paths, depending on the number of path continuation elements connected to the OR construct, can be identified. Note that it is not necessarely the case that such a path exists; not all theoretical possible paths through a use case map need to be paths which exist in the Universe of Discourse.

5.3.7 Global actor, detailed actor and value activity viewpoints

As discussed in chapter 3, a value model may consist of various sub viewpoints: (1) a global actor viewpoint explaining the overall value model to all stakeholders involved, (2) detailed actor viewpoints representing partnerships between actors or constellations of actors, and (3) value activity viewpoints representing who is doing what.

Global actor viewpoint

In simple exploration tracks, the viewpoint containing all actors *is* the global actor viewpoint, However, as discussed below, there can be motivations to detail the global actor viewpoint into constellations or partnerships.

Detailed actor viewpoints

There are two reasons to introduce viewpoints: (1) to model constellations of actors to reduce complexity, and (2) to model partnerships between actors: a joint offering of actors to their environment.

Guideline 2.23: *Use a detailed actor viewpoint (value constellations) to reduce complexity.*
Explanation. Value constellations capture parts of a value model. The main reason for doing so is reduction of complexity of discussions and the resulting model. Sometimes, discussions between stakeholders concern not all actors, but only a specific subset of actors. Moreover, these discussions may not contribute much to an overall understanding of the value model. In such a case, we introduce a detailed actor viewpoint to hide to complexity of these specific actor discussions for all other stakeholders involved.

Guideline 2.24: *Use a detailed actor viewpoint (partnerships) to model that actors have joint value interfaces.*
Explanation. A second reason to introduce a detailed actor viewpoint is to represent that actors are jointly offering or requesting objects to or from their environment. In such a case, two or more actors bundle objects they offer and request into one value interface of the composite actor. This can e.g. be used if objects are offered for a lower price as a whole, than that they were offered separately by individual

actors. In such a case, a detailed actor viewpoint *must* be developed, because is represents a case of bundling.

Value activity viewpoints

Value activities are introduced for the following reasons: (1) to discuss alternative assignments of activities to performing actors, and (2) to model the environment of a value model.

Guideline 2.25: *Use value activity viewpoints to discuss alternative assignments of value activities to actors.*
Explanation. Value activity viewpoint(s) show the assignment of value activities to performing actors. Multiple viewpoints can be used to show *alternative* assignments. During the development of a value model as described in this section, we assume the existence of one value activity for each elementary actor involved, with the same value interfaces as the actor has. This assumption is not modeled explicitly yet. Studying other activities as well alternative assignments of value activities to actors is part of deconstructing and reconstructing value models (see section 5.4 and chapter 6).

Guideline 2.26: *Use* environmental *value activities for activities which are needed to let the value model work but which are not of interest for profitability analysis.*
Explanation. We introduce *environmental* value activities if we are not interested in profitability analysis of these activities, but simply assume that they exist, and are capable of delivering objects of value. Such activities are typically introduced if an actor participating in the value model already performs an activity, and wants to develop some other activity. See also guideline 2.4.

5.3.8 A cyclic process

Figure 5.2 shows a number of steps to be executed to develop a value model. To our experience, these steps have to be taken a number of times, before actors agree on a value model and before they understand it. Decisions to be taken while executing a step are in practice too heavily interrelated with decisions in other steps. Also, to comply with concepts, relationships and constraints formulated by e^3-*value* ontology, a number of iterations are needed. In sum, we advocate that the formulation of a value model takes a number of exploration cycles.

Because steps are executed a number of times, sometimes steps can be left out in a cycle. As an example, it can be the case that a discussion on value interfaces reveals potential new actors, but no new scenarios. In a next cycle, scenario identification is then skipped, but attention is paid to actor identification.

After execution of a number of cycles, a value model should be found, such that (1) all stakeholders understand it and tentatively agree on it (execute decision making is done after evaluation of an e-commerce idea), and (2) it complies with the e^3-*value* ontology.

5.4 Deconstruct and reconstruct value models

If a value model is known, it can be used to find variations. A way to find such variations is to *deconstruct and reconstruct* a value model. This is discussed more in depth in chapter 6, so we only report briefly on value model deconstruction and reconstruction now.

Deconstruction and reconstruction takes the following steps. First, we deconstruct value objects and ports into smaller value objects and ports to find smaller portions, which can be requested or offered by an actor from or to his/her environment, Second, we debundle value interfaces and value offerings, into value interfaces and offerings with a smaller number of value ports. Third, we deconstruct value activities into smaller value activities. Finally, we reassemble new value models, by assigning the newly found value activities to actors.

5.5 Develop other viewpoints

The focus in this chapter is how to execute an exploration track from a value perspective. However, in chapter 2 we argued that it is important to develop other viewpoints, such as a business process viewpoint and an information system viewpoint. How to explore these other viewpoints is not a topic of this chapter (we exemplify how to do so in chapter 9). However, the outcomings are important. On the one hand, these viewpoints can indicate whether a value model is operational and technical feasible. As such, exploring these viewpoints may cause changes in a value model. In Gordijn & van Vliet (1999) we discuss how the exploration of a security viewpoint influences the value model at hand. On the other hand, exploration of process and information system viewpoints yields knowledge about operational and capital expenses, which are of use to construct profitability sheets.

Guideline 4.1: *Start with a value viewpoint, but develop other viewpoints as soon as possible to reveal insight in operational and technical feasibility as well as substantial expenses quickly.*
Explanation. It is our experience that the exploration tracks we have carried start out with an articulation of a value model. However, after the construction of such a model, it is worthwhile to explore other viewpoints also, because they can reveal important information regarding operational feasibility of an e-commerce idea. If it turns out that a value model is hardly operationally feasible, further exploration of the model can be stopped, or alternatives can be searched for. Moreover, early exploration may gain insight in substantial operational and capital expenses.

5.6 Evaluate e-commerce ideas

Step 4 explores other (process and information system) viewpoints (see chapter 9). This chapter focuses on exploration of the value viewpoint, so we continue with a discussion on e-commerce evaluation from a value perspective.

Evaluation of an e-commerce idea focuses on the question whether an idea is feasible from an economic point of view, that is whether an idea is profitable for each actor involved. It is our experience that *numbers* on profitability *themselves* are not are very useful for stakeholders involved, because it is not possible to predict profitability numbers for innovative e-commerce ideas accurately. Results of exploitating such innovative ideas are unknown by definition, which makes it very difficult, if not impossible, to estimate important numbers to determine profitability, e.g. the number of scenario occurrences per timeframe. What is however important for stakeholders, is to *reason* about profitability, and to do a sensitivity analysis. This contributes to a better understanding of the e-commerce idea, in this case from a profitability perspective. To do so, we (1) create profitability sheets for each actor involved in the value model, (2) ask actors to assign economic value to objects delivered and received, and (3) use evolutionary scenarios to determine effects of expected changes in the future that influence profitability.

5.6.1 Create profitability sheets

Profitability sheets. Profitability sheets are constructed for each actor involved, and present revenues and expenses associated with the execution of the e-commerce idea under consideration. The structure of a profitability sheet is shown in table 5.1. It contains for each actor value objects flowing into- and out as a result of scenario path execution. Also, substantial operational expenses as a result of performing activities and expenses caused by exploiting an information system and performing operational activities are shown by a profitability sheet. In this section, we focus on the creation of profitability sheets based on the value viewpoint. Chapter 9 exemplifies how we take in account expenses on the process- and information system viewpoint.

Guideline 5.1: *Create profitability sheets by following scenario paths.*
Explanation. To create profitability sheets for actors, we utilize our UCM scenario paths (see chapter 3). These paths put into operation a scenario, and show which value objects are exchanged by actors via their value interfaces, as a result of the occurrence of one or more start-stimuli. If, as a consequence of scenario path execution, an actor needs to exchange value objects, the path touches the value interface of that actor. Touching such a value interface by a scenario path is represented by a scenario path's responsibility point.

Profitability sheets are constructed by following for each scenario its scenario paths. By following a scenario path, and by searching for responsibility points on that path, we find the objects of value each actor exchanges as a result of executing the path. So each time we find a responsibility point, we examine the value interface it touches. The object(s) flowing out the interface of that actor are added to the actor's profitability sheet in the column *value object out*, while the objects flowing into an actor are added to the actor's profitability sheet the in column *value object in*.

Estimate scenario occurrences. To calculate profitability for each actor involved, we need to know the expected number of scenario occurrences per timeframe (e.g. per month), and the likelihood that a scenario path of a scenario is executed.

Guideline 5.2: *Estimate the number of scenario occurrences by estimating the number of start stimuli occurrences.*
Explanation. Scenarios are described by scenario paths, of which start stimuli are part of. These stimuli are the drivers for scenario paths. Consequently, to estimate the number of scenario occurrences, we must estimate the number of start stimuli.

Guideline 5.3: *The percentages of likelihoods for scenario paths which put a scenario into operation should sum up to 100 %.*
Explanation. A number of scenario paths put into operation a specific scenario. To

Table 5.1: Structure of a profitability sheet.

Actor	x	
Viewpoint	value viewpoint	
Scenario	a	
Occurrences/timeframe	...	
	Value Object In	*Value Object Out*
Scenario path	*1*	
Likelihood	...%	
	Euro x_1	Euro y_1
...	...	
Scenario path	n	
Likelihood	...%	
	Euro x_n	Euro y_n
Scenario	...	
Scenario	z	
Viewpoint	Business process	
Scenario	*Similar to the value viewpoint, but with potentially a different number of scenario paths.*	
Viewpoint	Information system	
Scenario	*Similar to the value viewpoint, but with potentially a different number of scenario paths.*	

calculate the profitability of the execution scenario occurrences per timeframe, we must therefore know the chance that each scenario path will be executed. Likelihood percentages for paths of such a scenario must sum up to 100 %. Otherwise, scenario paths have been forgotten, or estimations are not adequate.

5.6.2 Assign economic value to objects

After a profitability sheet for each actor has been constructed, actors are asked to assign economic value to objects flowing into or out themselves. We then can calculate profitability numbers for each actor. Note that if we only calculate this ‘profitability’ for the value viewpoint, we do not take in account operational expenses as a result of executing business processes and exploiting an information system. Also, investments needed are not part of this profitability number. However, if for one of the actors profitability is less or equal to zero, the e-commerce idea is not likely to be profitable for such an actor, given the identified model and estimations on scenario occurrences, on scenario path likelihoods, and on valuation of objects by actors.

We distinguish two actor types, who assign economic value to objects in a different way:

1. enterprises: these are actors who produce, resell, or distribute objects to *make profit*, or at least to cover their expenses;
2. end-consumers: these are actors who do not resell value objects, but use obtained objects to *create economic value* for themselves.

Assign economic value to objects: enterprise perspective

Guideline 5.4: *Assume enterprise actors strive for profit maximization: they value only money objects.*
Explanation. Enterprises want to maximize their profit: in short revenues minus expenses to generate revenues. As such, we only take in account value objects representing money flows to calculate an enterprise's profitability sheet. This also suggested by investment theory (see e.g. Horngren & Foster (1987)), who take in consideration cash-in and -outflows only. We assume that all other objects (not representing money) flow into an enterprise, and after some time flow out the same enterprise, and are not of relevance for determining profitability.

We distinguish the following steps in investigating valuation of objects by enterprise actors. First, for each value port representing the exchange of money objects, we determine its valuation function. The valuation function returns the *amount* of money to be paid for obtaining other, money, value objects. Second, we must assess whether each non-money value object, which flows into an enterprise, also flows out this enterprise. To do so, we *reduce* the profit sheet by removing non-money value objects which are flowing into and out an actor.

Determine valuation functions for ports exchanging money objects. Value objects are offered and requested via ports of a value interface. In many cases, at least one object exchanged via a port in an interface represents money. For such a port, we determine a valuation function. This function calculates the amount of money to be paid or to be received for obtaining another value object(s) via ports of the same value interface.

A valuation function is in many cases determined by the actor receiving a payment, but can also be a result of a negotiation between actors. The function uses a number of properties, e.g. properties of the product to be sold, to calculate the amount of money. Investigating these properties, as well as assigning values to these properties is part of determining a valuation function. Chapter 8 exemplifies a number of these properties for an enterprise actor.

Reduce other (non-money) value objects. Objects representing something else than money are not considered in the enterprise's profitability sheet. However, to check whether all in-flowing non-money objects are also leaving the enterprise actor (and vice versa) we *reduce* the profitability sheet of an actor. Reduction means removal of non-money value objects of a profitability sheet if these objects are causally related.

Guideline 5.5: *Use objects which can not be reduced to find yet undiscovered actors or value activities.*
Explanation. In some cases, it is not possible to reduce objects, because the causally related in- of out-going object have not been modeled. This can be an indication that actors have been forgotten, or part of the environment of the value model has been omitted (e.g. actors or value activities in which we are not interested from a profit perspective, but which are needed to let the value model work, see section5.3). Then the value model itself needs to be reconsidered.

In sum, valuation from an enterprise perspective consists of finding valuation functions for objects exchanging money. Non-money objects are removed from an actor's profitability sheet, if other causally related non-money objects can be found. Chapter 8 exemplifies in more detail how we deal with valuation of objects by enterprises in a project carried out on offering news articles online.

Assign economic value to objects: end-consumer perspective

Guideline 5.6: *Assume end-consumers strive for consumer value maximization: they value all objects.*
Explanation. End-consumer actors do not aim at profit. Rather, they want to satisfy their needs. To do so, end-consumers can generally select from a number of different value objects offered by others. In general, these value objects satisfy end consumer's needs not to an equal extent. Some objects will fulfill end-consumer's needs nearly completely, while others do so only very limited. Which object will be chosen by an end-consumer?

To make a decision, an end-consumer assigns an economic utility to each object (see e.g. Kotler (1988)). Second, to obtain an object, an end-consumer must give another object in return. In most cases this is a fee in Euros or Dollars. According to Kotler (1988), the end-consumer then will choose for the object that delivers the most utility per Euro, if s/he is a rational acting person. This is in axiology literature also known as *consumer value* maximization (Holbrook 1999). As a consequence, to assess to what extent an end-consumer maximizes his/her consumer value, we need to know how an end-consumer assigns economic value, especially to non-monetary objects. To do so, we identify market segments to find actors who value objects equally, and then identify *valuation functions* for value objects exchanged via ports of the aforementioned market segments. These functions return the *utility* assigned to an object in terms of an monetary unit (Euros or Dollars). By doing so, we make non-monetary objects comparable with monetary objects seen from a utility perspective.

Determine valuation functions for ports of an end-consumer For each value object exchanged by an end-consumer we provide a valuation function. This function returns, given a number of properties, the economic value in terms of a monetary unit assigned to an object by an end-consumer. These properties can be observable object properties, but can also be consumer specific properties.

Guideline 5.7: *Assume end-consumers use a multi-criteria approach to assign value to object.*
Explanation. We assume end-consumer actors use a number of criteria to 'calculate' the value of a non-money object. Also, these criteria can be weighted differently. To elicit these criteria, we utilize Holbrook's (Holbrook 1999) consumer value framework. In chapter 7, we show how to do so in a practical study on the legal and illegal distribution of music.

For money objects, also a valuation function must be available. In most cases, an enterprise selling an object determines this valuation function, or the function is a result of a negotiation process.

Related valuation theory

Investment theory. Profitability sheets are inspired on investment theory (see e.g. Horngren & Foster (1987) and Drury (1998)). To judge an investment, all expenses and revenues are identified, including initial cash outflow for investments. All expenses and revenues for a period are summed up, and if this sum is positive, an investment is potentially successful. In our approach, the economic value of objects is expressed in terms of monetary units (e.g. Euros or Dollars), and can be seen as expenses (objects flowing out an actor), and revenues (flowing into an actor).

It is also possible to account for the time effect of money. A Euro or Dollar (an initial expense) can be invested in the execution of an e-commerce project, but can e.g. also be invested in a savings institution. Therefore, a *discounted* cash flow calculation takes in account that money can be invested with a certain interest rate.

There are two ways to calculate with discounted cash flows. First, the net present value (NPV) can be calculated. The NPV is the expected net monetary gain or loss from a project by discounting all expected future cash inflows and outflows to the present, using some predetermined minimum desired rate of return. If the NPV is a positive value, the NPV denotes the extra amount of money received if the project is executed, compared to investing the initial expense with the predetermined interest rate. Also, the internal rate of return (IIR) can calculated. This is rate of interest at which the present value of the expected cash inflows from a project equals the present value of the expected cash outflows of the project. For a positive decision, the IRR should be higher than a predetermined interest rate of savings institutions.

To deal with the time-effect of money, multiple periods (e.g. months or years) can be distinguished during which objects of value are exchanged. Also, for each period, different figures for the number of scenario executions can be estimated. If we also know operational expenses and the needed investment, e.g. in information technology, we can estimate the Net Present Value for each actor participating in an e-commerce idea.

Consumer value theory. The way end-consumers assign value to objects has been extensively studied by Holbrook (1999). He defines consumer value as an interactive, relativistic preference experience.

With *interactive*, Holbrook means that consumer value entails an interaction between some *subject* (a consumer) and some *object* (a product or service). The value the consumer assigns to an object is determined by *both* object properties and by the subjective experience of a consumer when s/he uses/consumes the object. This is also seen in e^3-*value* : the value end-consumers assign to an object is based on properties of the object, but also by consumer specific properties.

Consumer value is *relativistic* in a way that it is *comparative*, *personal*, and *situational*. *Comparitive* means that we can state the value of an object only in reference to that of other objects, as evaluated by the *same* individual. According to Holbrook statements like ‘I like ice cream better than you like ice cream’ are illegitimate. Value is *personal* in the sense that it varies from individual to individual. People may assign different value to the same objects. Consumer value is *situational* in that it depends on the context in which the evaluative judgment is made. For instance, a same person can value a same object differently, depending on his/her physical location during valuation or the time of valuation.

Consumer value is *preferential*. This means that we assume that a unidimensional index of preference ordering exists. To come to such a preference ordering of objects, consumers may use multiple criteria of different importance. In e^3-*value* , the unidimensional index is expressed in monetary units, to allow a preferent ordering of non-money *and* money objects. The criteria to come to this unidimensional index are inspired on Holbrook’s consumer value framework. We elaborate more on this framework in chapter 7.

Finally, consumer value is an *experience*. The value is not in the object obtained, not in the brand chosen, not in the object possessed, but rather in the consumption experience derived therefrom. Actually, all products provide services because they create need-satisfying experiences.

5.6.3 Evaluate using evolutionary scenarios

Evolutionary scenarios

The profitability for each actor estimated by using profitability sheets, valuation functions, and scenario occurrences and path likelihoods, may differ substantially from reality, during execution time of an e-commerce idea. There can be various reasons for this.

First, a value model (including UCM scenario paths and valuation of objects by actors) may be *incomplete*. Expenses and revenues which occur during execution of the e-commerce idea then have not been foreseen. Moreover, if profitability sheets are only based on a value viewpoint, and not on process- and information system viewpoints or additional viewpoints, profitability numbers surely will not reflect reality. They should then be seen as a surplus of money needed to cover additional expenses.

Second, estimates such as valuation functions and the number of scenario occurrences may be subject to *uncertainties*. For instance, real consumer behavior can be different from estimations made during idea exploration. Also, future events may cause profitability numbers, which differ from estimated numbers. With respect to future events, van der Heijden (1996) distinguish the following uncertainties:

(1) risks, (2) structural uncertainties, and (3) unknowables. *Risks* are events which can be predicted, in many cases based on historical date, and a likelihood for the occurrence of such an event can be estimated. *Structural uncertainties* are not frequently occuring events, which can be thought of, but for which the likelihood of occurrence can not be estimated. Finally, *unknowables* are events which can not be foreseen.

In sum, it is very likely that identified profitability numbers will *not* be the profitability numbers once an e-commerce idea is put into operation. Moreover, profitability numbers will vary during the time span the e-commerce idea is in execution.

What is then the value of profitability sheets? First, positive numbers on profitability can contribute to an increase of stakeholders' confidence that an e-commerce idea can be successful. Also negative profitability numbers found for actors act as drivers to redo parts of value model construction process, which may lead in a better understanding of the idea. Either the value model should be changed such that each actor has positive profitability numbers, or, if such a change is not possible, the e-commerce idea seems not to be feasible. Second, the profitability sheets can be used to reason about conditions which influence profitability of an e-commerce. It can explain to stakeholders critical factors, and make stakeholders aware of strong and weak points of the e-commerce idea.

To facilitate reasoning about profitability sheets we employ *evolutionary* scenarios. In contrast to *operational* scenarios, which describe behavioral aspects, evolutionary scenarios describe events which are expected to possibly occur in the future. As such, effects of events underlying risks and structural uncertainties are analyzed, as well as effects of wrong estimations.

Elicit evolutionary scenarios

Scenario techniques to evaluate effects of expected events occuring in the future are used in a number of disciplines. From a business perspective, van der Heijden (1996) discusses scenarios as a tool for executive decision making (see also Ringland (1998)). In the realm of software engineering, properties of information systems are evaluated during the development using evolutionary scenarios (Bass, Clements & Kazman 1997, Lassing 2002).

Two extreme positions on finding scenarios exist (Carroll & Rosson 1992). On the one extreme, scenarios can be collected empirically. This is often done by interviewing stakeholders, or having workshops on scenario identification. On the other extreme, some theory of scenarios can be used. Such a theory identifies the kinds of scenarios that exist. These types of scenarios are used to organize scenarios, but also to generate scenarios.

Guideline 5.8: *Find evolutionary scenarios by asking stakeholders, while keeping various kinds of scenarios in mind.*
Explanation. In practice, we elicit scenarios by interviewing stakeholders and/or doing workshops. While doing these interviews and workshops we keep in mind various kind of scenarios which may occur: (1) scenarios which result in changed valuation functions, (2) scenarios which result in changed numbers of UCM scenario occurrences and likelihoods, and (3) scenarios which result in a changed value model structure.

Guideline 5.9: *Use a change in a valuation function to find an evolutionary scenario.*
Explanation. Enterprises may decide, during execution of an e-commerce idea to price objects differently than was estimated during idea exploration. They change then their valuation function for money objects. They are motivated to do so, if valuation functions for end-consumers are not estimated correctly. Valuation functions can also change as a result of other causes. For instance, in chapter 8, we discuss an e-commerce idea, which is largely based on price setting by a market regulator. If this same regulator changes prices after some period, profitability for some actors may decrease or even become negative.

Guideline 5.10: *Use a change in the number of expected UCM scenario occurrences and likelihoods to find an evolutionary scenario.*
Explanation. A realistic estimation on the number of UCM scenario occurrences per timeframe, as well as the likelihood of scenario paths is important but difficult. It is difficult because innovative e-commerce ideas are about new, unknown value propositions, so hardly any historical data can be used to estimate the number of UCM scenario occurrences. A realistic estimation is important, because the number of scenario occurrences directly relates to the number of value exchanges per timeframe, and affects the profitability sheet. This estimation becomes even more important if for the execution of the e-commerce idea investments are needed, which depend on the number of scenario occurrences. Consider for instance our previously discussed project on free Internet access (see chapter 3). Offering such a service requires substantial investments in computer hardware such as access servers, and capacity in telephone switches. These investments are done before the e-commerce idea is put into operation. If such a service e.g. is based on about 1000

scenario executions per hour, and in practice only 100 scenario executions per hour occur, it can be expected that the estimated profitability will decrease. Therefore, it is worthwhile to identify a number of events, such as a complete failure of the e-commerce idea, (formalized by very few scenario occurrences), and a success of the idea.

Guideline 5.11: *Use a change in a value model's structure to find an evolutionary scenario.*
Explanation. Also the structure of the model, consisting of actors, activities, exchanges, interfaces, etc. can evolve. Evolutionary scenarios can be used to study these effects. Likely changes are shifts in value activities, new actors (e.g. competitors), and disappearing actors.

5.6.4 Analyze scenario effects and feedback

If evolutionary scenarios have been identified, we analyze consequences of them on profitability sheets of actors. We produce an overview of effects of evolutionary scenarios, starting with a null-scenario. The null-scenario reflects the original value model, while other scenarios are changes to this model, including UCM scenario paths and valuation functions. We then present this overview to stakeholders, and discuss if specific evolutionary scenarios should result in a changed value model. If this is the case, we start a new cycle for value model construction (see section 5.3).

5.7 Conclusions

In this chapter we have presented a process for developing value models, as well as guidelines for doing so. The process assumes an existing, innovative e-commerce idea and focuses on the articulation of this idea, finding variations of the idea, and evaluation of the idea from an economic perspective.

Various process steps interact, therefore we propose a cyclic process for exploring the e-commerce idea. The construction of value models themselves can take two different approaches: (1) an actor oriented approach, which assumes one key actor, and builds the value model around this actor, and (2) a market oriented approach, which starts with a consortium of actors. Also, guidelines have been presented to find constructs part of the value model, such as value interfaces, offerings, ports and exchanges.

If a value model has been constructed, we start to elicit variations on it. These may stem from deconstruction of value activities, deconstruction of value objects offered and requested, and debundling of value interfaces. After deconstructing a value model, it is reconstructed again by assigning newly found value activities to performing actors.

Finally, a value model needs to be evaluated. Evaluation in the light of e^3-*value* consists of the construction of profitability sheets on a per actor basis and valuation of objects received and delivered by actors. This results in profitability numbers for actors involved in the execution of the e-commerce idea at hand. Then evolutionary scenarios are identified, which capture expected changes to the value model, and consequently an actor's profitability. Analysis of evolutionary scenario effects on profitability may lead to changed value models, and/or increased confidence in, and a better understanding of the e-commerce idea by stakeholders.

Chapter 6

Value model deconstruction and reconstruction

In the previous chapter, we have presented a way to come from an e-commerce idea to one or more value models. One way to find variations on an e-commerce idea is to deconstruct and reconstruct a value model that captures this idea. How to do so is the focus in the chapter.

To deconstruct a value model, e^3-*value* defines value model *deconstruction* operators (mainly inspired by Tapscott et al. (2000), Evans & Wurster (2000), and Timmers (1999)). These operators are part of a value model deconstruction and reconstruction process, during which we *de-assign* activities from their performing actors, try to find alternative, and/or more activities by deconstructing existing ones, and *re-assign* newly found activities to executing actors. Because we assume that activities are profitable for at least one actor, re-assignment should be possible. Essentially, to clarify discussions between stakeholders, we split the deconstruction and reconstruction process into two questions: (1) which value adding activities exist, and (2) which actors are willing to perform these activities?

Based on our e^3-*value* ontology, we discuss three generic operators for value model deconstruction: (1) the *value activity* deconstruction operator, which breaks an activity into smaller ones, but leaves the products/services offered or requested by the original activity to its environment unchanged, (2) the *value port* deconstruction operator, which breaks a service/product offered or requested by a value activity into smaller ones, and (3) the *value interface* deconstruction operator, which breaks combinations of value objects offered *and* counter-compensations requested into smaller pieces.

We illustrate value model deconstruction and reconstruction by one of the projects where we successfully applied our approach. The project at hand is about the provisioning of a value-added news service. With respect to such a service, a regular newspaper called the *Amsterdam Times* (a fictitious name, but based on an actual commercial e-commerce project) wants to offer to all his/her subscribers a service to read articles online using the Internet, but such that it will not result in any additional costs. Therefore, the idea is to finance the execution of this business idea by the telephone connection revenues, which originate from the reader who has to set up a telephone connection for Internet connectivity.

This chapter first introduces in brief value model deconstruction and reconstruction (section 6.1). In section 6.2 we present a value model, based on the aforementioned project, to be reconstructed. Sections 6.3 and 6.4 discuss both the general theory and an application of value model deconstruction and reconstruction. Based on our lightweight ontology, we can create value model variations using only a small number of deconstruction operators. Deconstruction and reconstruction is in various forms also known in the literature, in section 6.5 we present an overview. Finally, in section 6.6 we present our conclusions.

6.1 Value model deconstruction and reconstruction

Value model deconstruction and reconstruction is about finding variations on an existing value model. One of its applications is to find more, and alternative value activities. As such, value model deconstruction and reconstruction can be seen as a way to explore value activity viewpoint(s).

The process of value model deconstruction and reconstruction assumes an existing value model (see for the construction of such a model section 5.3). The value model to be deconstructed states actors who exchange objects of value. Also, it (implicitly) contains one value activity per actor, stating what an actor is doing to make profit. In some cases, the actor may have additional value activities to model the environment (see section 5.3.7).

Value model deconstruction and reconstruction consists of two steps: (1) value model deconstruction, and (2) value model reconstruction.

For value model deconstruction, we *de-assign* actors from value activities, but leave value exchanges between value activities intact. Then, we repeatedly apply one of the deconstruction operators. As we result of the deconstruction process, we find new value activities which may have a finer granularity than the activities we started with.

Value model reconstruction takes the newly identified value activities, and *re-assigns* these to actors. In many cases, alternative assignments are possible.

In the coming sections, we present value deconstruction and reconstruction in more detail, and exemplify the process using an e-commerce exploration project we have carried out.

6.2 A value model for deconstruction and reconstruction

We illustrate value model deconstruction and reconstruction using a project we have carried out in the realm of online newspapers. From a newspaper subscriber's perspective, the e-commerce idea is to offer an archive of online newspaper articles for free. Only a subscription on the paper-based version of the newspaper is required.

The financial idea behind the article online service is to use a *termination* fee to finance the service. *Termination* means that if someone tries to set up a telephone connection by dialing a telephone number, another actor must pick up the phone, that is, *terminate* the connection. If someone is willing to *cause* termination of a large quantity of telephone calls, most telecommunication operators are willing to pay such an actor for that (the *termination fee*). Because the newspaper has a large subscriber base, s/he is capable to generate a large number of terminations for an *article online* service. We have seen this mechanism also in chapter 3. Moreover, this project is further discussed in chapter 8.

The aforementioned idea is formalized by an initial value model (see figure 6.1). The model shows that the Amsterdam Times (the newspaper) funds his/her service by a termination fee offered by a telecommunication operator, who essentially is a carrier of data traffic from the reader to the Amsterdam Times and vice versa. Amsterdam Times obtains from the readers termination opportunities and offers in return online articles.

The reader also needs a telephone connection (for data transport) to access the online article archive and offers in return a telephone connection fee. The latter exchanges are between a reader and a telecommunication operator and *not* the Amsterdam Times. Note that the ports of a reader are grouped into one value interface because these objects exchanged via these ports are only of value in combination for the reader. An online article needs a telephone connection for delivery.

Also, the value model shows value activities. For each actor, initially one value activity is assumed that describes his/her value adding process at best.

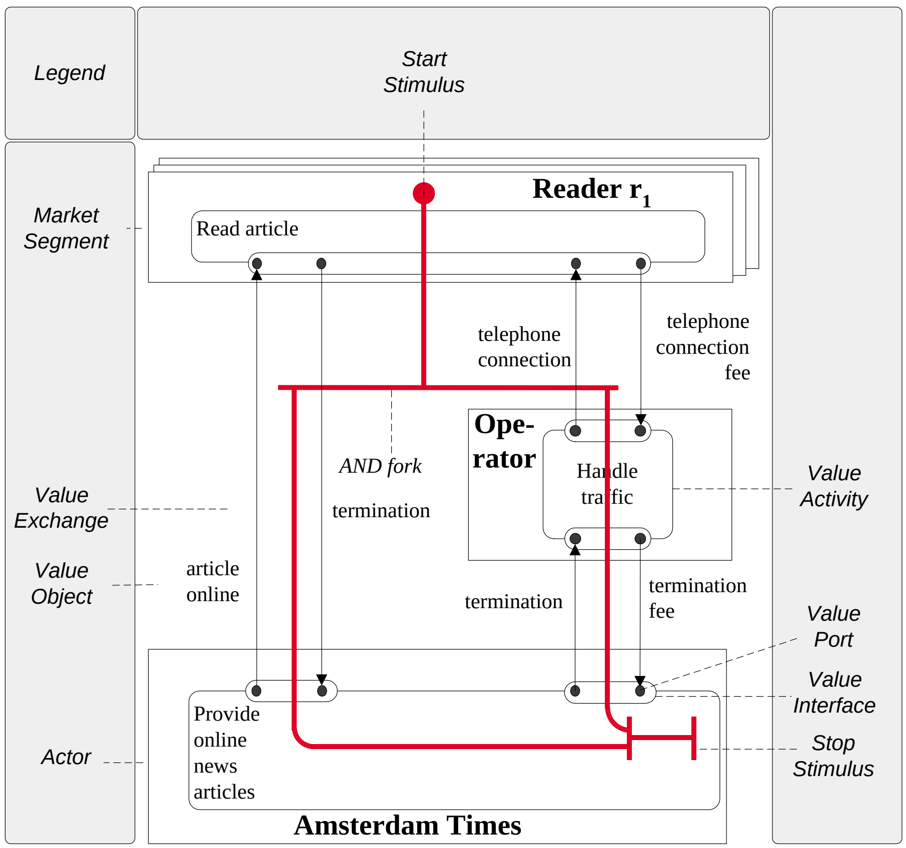


Figure 6.1: An initial value model for deconstruction and reconstruction.

6.3 Value model deconstruction

For value model deconstruction, we first de-assign value activities from their performing actors. By doing so we separate the question 'who is doing what' from 'what value activities do exist'. What remains is a value model with value activities connected by means of value exchanges, but without their performing actors.

Subsequently, we apply sequentially deconstruction operators. The *value activity* deconstruction operator splits up a value activity into smaller ones, while keeping the value interfaces of the original activity in tact. The *value port* deconstruction operator splits up ports (and thus value objects) into smaller ones. Finally, the *value interface* deconstruction operator is used to find value interfaces with a smaller number of value ports than the original one.

In the remainder of this section, we present each deconstruction operator by providing a business rationale for its existence, the focus of operator, a guideline how to apply the operator, and an example based on the online article value model.

6.3.1 Value activity deconstruction

Business rationale. Can we split a value activity, which initially is viewed as being performed as a whole by one actor, into smaller activities, together behaving as the original one, whereby each smaller activity potentially can be performed by different actors?

Focus. The value activity deconstructor focuses on the *internal* structure of a value activity while leaving its value interfaces to the environment in tact. It breaks down a value activity into smaller ones, for instance to allow specialized actors to perform one of these value activities.

Operator $VAD : a \rightarrow a_1, ..., a_n$**.**

1. Deconstruct a value activity a with value interfaces $i_1, ...i_n$ into value activities $a_1, ..., a_n$.
2. Assign each value interfaces $i_1, ...i_n$ to one or more of the deconstructed value activities.
3. Add, if necessary, extra value interfaces to the deconstructed value activities, and relate these by value exchanges. Extra value interfaces and exchanges

can be necessary to ensure that the deconstructed activities $a_1, \ldots, a_n$ are from an environment perspective equivalent to a.

4. Reconsider scenario segments, which hit the value interfaces of value activity a.

It is possible that for a value activity a multiple alternative deconstructions exist.

Example: Deconstruct the Handle traffic *value activity into two other value activities.*

Figure 6.2 deconstructs the *Handle traffic* value activity into two smaller value activities, which each can be potentially performed by a single, and different, actor. The two value interfaces of *Handle traffic* can be found at the two smaller value activities, thereby providing the same interfaces to their environment as the original value activity. In this case, additional value interfaces are needed. The value activity *Handle local traffic* offers end-to-end connectivity to a reader and gets paid for this, while it only exploits the local loop: the last miles from a local telephone switch to the reader. Consequently, this activity should 'buy' interconnection from the *Handle long distance traffic* activity, and pays for this in return. The latter activity exploits a telecommunication network between local telephone switches. Buying interconnection is shown by adding value interfaces and value exchanges between *Handle local traffic* and *Handle long distance traffic*. The scenario path is changed but hits at least the same value interfaces as was the case for the *Handle traffic* value activity.

Example: Deconstruct the Provide Online news articles *value activity into two other value activities.*

The deconstruction shown in figure 6.3 essentially separates the content creation (news) from the technical infrastructure needed to deliver content to the reader. It can be seen as *outsourcing* Internet service provisioning from a news provisioning perspective. Again we need to add value interfaces and value exchanges to represent that the *Provide news articles* value activity must acquire facilities for Internet service provisioning. Note that the scenario path for the deconstructed value activities hits at least the same value interfaces as the original value activity. However, internally, the scenario path splits to show that as a result of a termination/article online exchange, also a termination/termination fee *and* an Internet service provisioning/fee is necessary.

6.3.2 Value port deconstruction

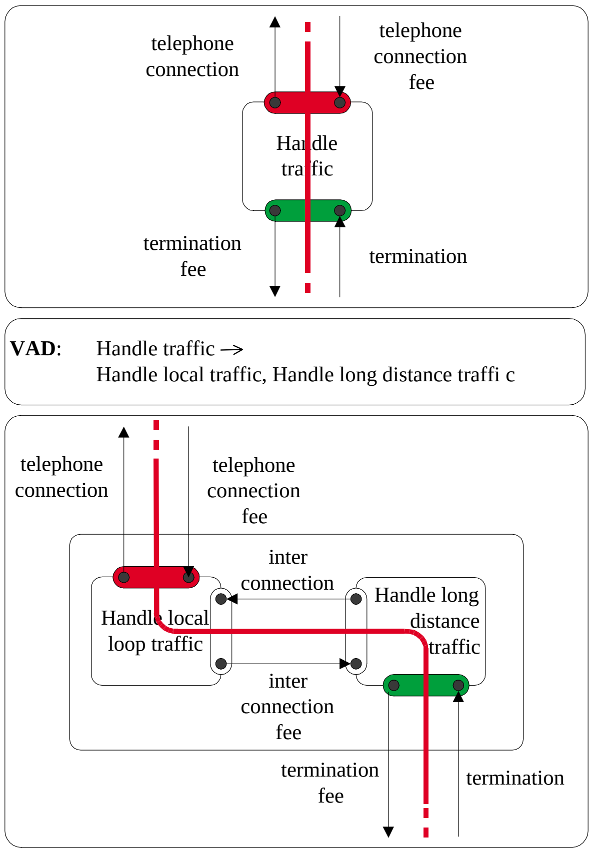


Figure 6.2: Deconstruction of the value activity *handle traffic* into two value activities *handle local loop traffic* and *handle long distance traffic*.

Business rationale. Can we split products, services or combinations into smaller products/services, which each can be delivered and consumed by individual actors?

Focus. Focus is to untangle offered or requested value objects via ports, which still are of value for actors. These objects can potentially be offered by multiple value activities rather than one, and thus by multiple actors. Because we change the value port, we change the value interface of a value activity to the environment.

Operator $VPD : p \longrightarrow p_1, \ldots, p_n$**.**

1. For each value port p in a value interface:

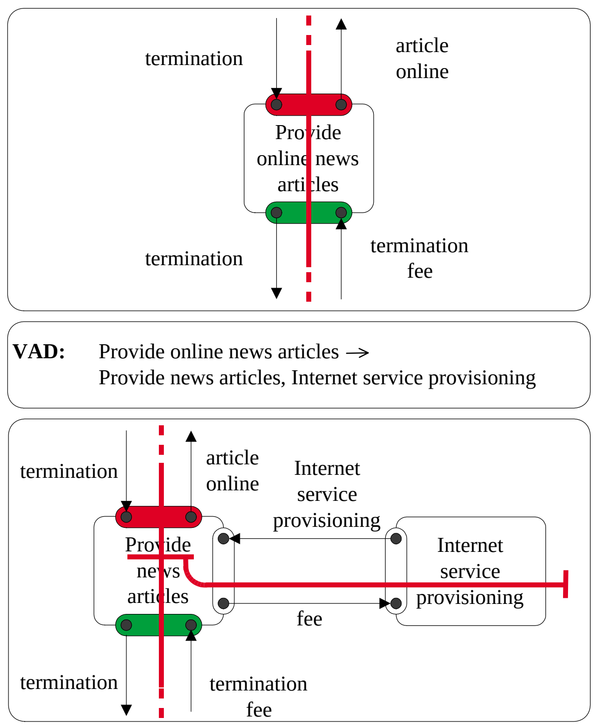


Figure 6.3: Deconstruction of value activity *provide online news articles* into two value activities *provide news articles* and *internet service provisioning*.

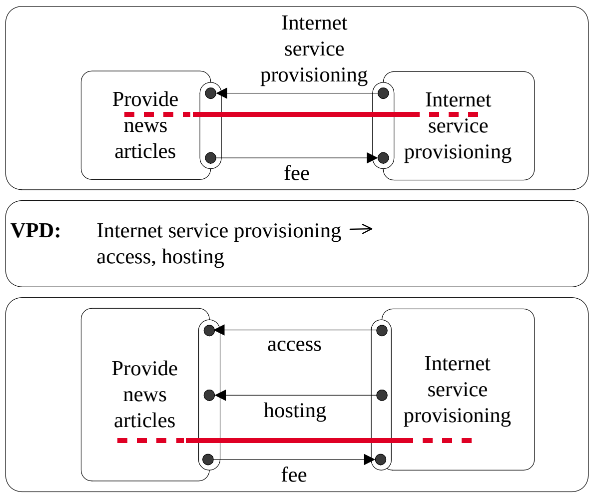


Figure 6.4: Deconstruction of the value port exchanging the value object *Internet service provisioning* into two value ports *access* and *hosting*.

2. Consider deconstruction of value port p with value object o into value ports $p_1, ..., p_n$ with value objects $o_1, ...o_n$.

3. If deconstruction is possible, deconstruct also the peer-ports of p. *Peer* ports are connected by value exchanges to value port p. Note that a value port p can be connected to multiple peer ports, representing that a value activity containing port p can exchange objects with multiple other value activities. For each peer port p_p:

   (a) Disconnect value exchanges connecting value port p and peer port p_p.

   (b) Deconstruct value port p_p into ports $p_{1_p}, ..., p_{n_p}$ in the same way as p was deconstructed.

   (c) Reconnect ports $p_1, ..., p_n$ using value exchanges with ports $p_{1_p}, ..., p_{n_p}$.

Example: Deconstruct the value object Internet service provisioning *into two other value objects.*

Figure 6.4 deconstructs the value port *Internet service provisioning* into two different ports/value objects: (1) Internet *hosting* provisioning, e.g. hosting a web site, and (2) Internet *access* provisioning, e.g. exploiting a modem pool to offer access to the Internet.

6.3.3 Value interface/offering deconstruction

Business rationale. A value interface models the notion of *economic reciprocity*, consisting of value offerings delivered and requested in return. A value offering groups ports of equal direction. During the deconstruction process, we can split up a value interface/offering into more interfaces/offerings, for two reasons: (1) debundling of value ports in a value offering, modeling that some objects can be obtained separately, and (2) finding smaller value activities. We then split up an interface into smaller ones, whereby each value interface can be associated to a new value activity, which in turn can be assigned to actors.

Focus. The focus is to find smaller value interfaces, that is value interfaces with a smaller number of value ports.

Operator $VID : i \rightarrow i_1, ..., i_n$**.**

1. For each value interface i with value ports $p_1, ..., p_n$ of a value activity a:
2. Find (alternative) value interfaces $i_1, ..., i_n$:
   (a) Choose one of the two value offerings o_{in} (in-going offering) or o_{out} (out-going offering) in value interface i. We call this offering the *focus* offering. The value offering not chosen we call the *remaining* offering.
   (b) Find new value offerings $o_1, ..., o_n$ by grouping value ports $p_1, ..., p_n$ of the focus value offering into new offerings. The cardinality (the number of ports) of each newly found offering must be smaller than the cardinality of the focus offering. Also, each port of the focus offering must be part of exactly one new offering (the ports of the focus offering must be partitioned over new offerings). In case a port of the focus offering can be part of more than one new offering, the value interface can be deconstructed in multiple alternative value interfaces.
   (c) Repeat the previous step until each value port of the focus offering is grouped into one of the new offerings.
   (d) Deconstruct the *remaining* offering: For each newly found offering $o_1, ..., o_n$, find the reciprocal offerings $o_{reciprocal_1}, ..., o_{reciprocal_n}$:
   - Consider each port p_k of the new offering o_j;
   - Select one or more of the ports of the *remaining* offering, which is a compensation for the object of port p_k; Add these ports to an offering $o_{reciprocal_j}$, which is the reciprocal offering for offering o_j;

- Group the new offering o_j and $o_{reciprocal_j}$ into one new value interface.

(e) The remaining offering is deconstructed if all its ports have been grouped into new offerings.

3. Reconsider scenario segments. As a result of introducing new value interfaces, the scenario paths need to be reconstructed.

Alternative deconstructions of value interfaces can be possible. Consequently, the above discussed deconstruction process can be applied a number of times on the same value interface to find alternative deconstructions.

Example: An access and hosting value interface.

Figure 6.5 introduces two separate value interfaces for the Internet service provisioning activity: one for offering Internet access and one for offering hosting services. Creation of these interfaces takes two steps. First we have to deconstruct the *fee* port into two ports: the *access fee* and *hosting fee*. This is necessary due to the nature of value interface. A value interface models objects of value offered to the environment *and* the objects requested in return. We therefore need ports who receive the objects requested in return for offering *access* and *hosting* value objects. Second, we create two value interfaces, representing *hosting* and *access* services.

Note we do *not* split the value interface of the *Provide news articles* value activity. This value interface models that, for offering articles online, we need *both* hosting and access for each scenario occurrence.

Example: Access and hosting via value activity *deconstruction.*

It also possible to split up the *Internet service provisioning* value activity into *Internet access provisioning* and *Internet hosting provisioning* (see figure 6.6), but there is an important difference compared to the previous example. Figure 6.6 still shows a value activity called *Internet service provisioning'* (although smaller than the original one). This activity is profitable by offering a bundle of access and hosting services, but must buy-in access and hosting from another service provider. In contrast, in figure 6.5, the value activity *Provide news articles* is responsible for acquiring both access and hosting.

6.3.4 Combining deconstruction operators

The three mentioned deconstruction operators can be sequentially applied. The following three cases appear regularly:

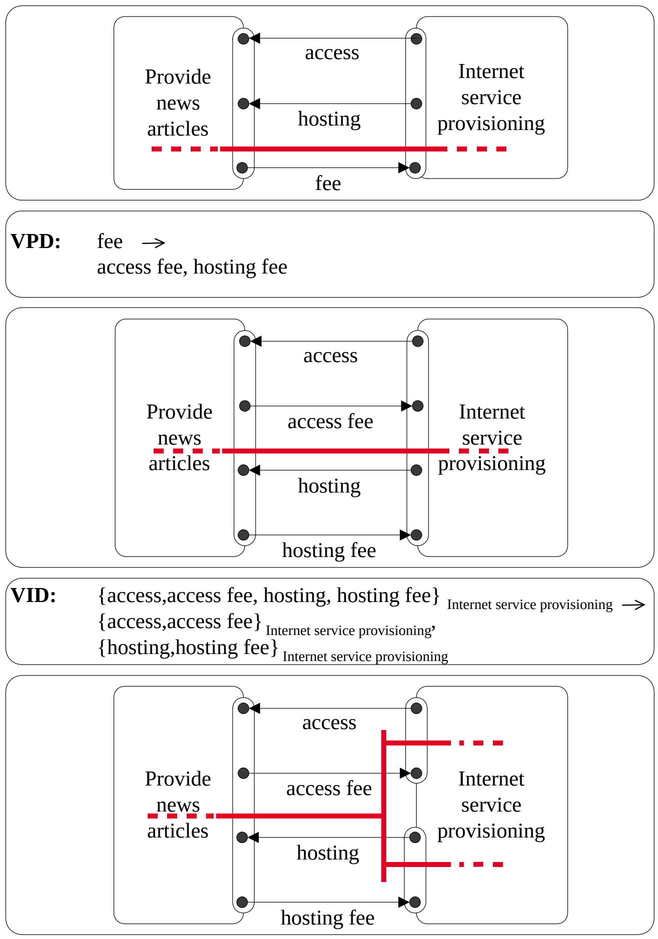


Figure 6.5: Deconstruction of the value interface with four ports into two value interfaces with each two ports.

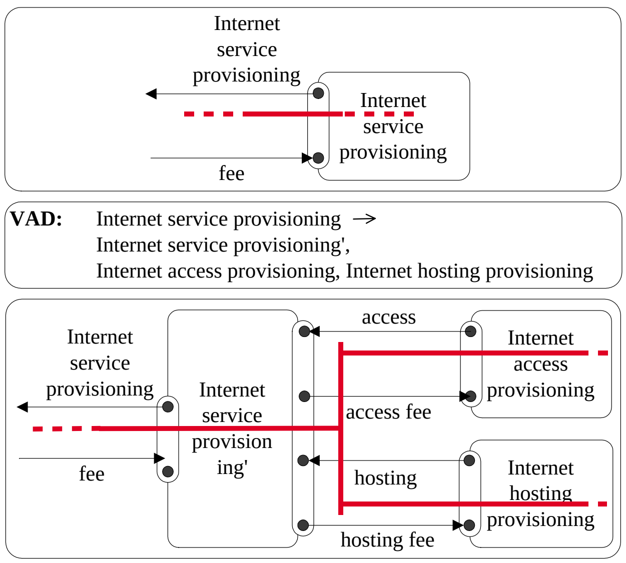


Figure 6.6: Deconstruction of the value activity *Internet service provisioning* into one for access provisioning and one for hosting provisioning. In contrast to figure 6.5, the *Internet service provisioning'* ensures that there exists still one bundle of Internet service provisioning, while in figure 6.5 an actor who wants *access* and *hosting* must compose the bundle him/herself.

- a number of sequential value activity deconstruction operations. In this case, we try to break up a value activity into (alternative) smaller ones, but do not change anything visible to the outside world.

- value port deconstructions, followed by value interface deconstructions, and finally value activity deconstructions. In this case, we try to find smaller value objects which can be offered by separate value activities, which can be performed by individual actors. Figure 6.7 is an example of this. First we deconstruct the value interface of *Internet service provisioning* into two smaller ones for access and hosting (see figure 6.5), and then we deconstruct the value activity into two smaller ones.

- debundling: a number of value port deconstructions, followed by value interface deconstructions. Figure 6.5 can be seen as a case of debundling: we allow that the services *hosting* and *access* are sold separately rather than as a whole. Note that a value interface means that if a value object is exchanged via one of its ports, value objects on all its other ports must be exchanged too, so after debundling, access and hosting can be obtained as separate services rather than as a whole.

6.4 Value model reconstruction

Deconstruction of a value model means de-assigning value activities and actors, and generating new value activities. During value model reconstruction, we study the re-assignment of value activities to performing actors.

Generate value activity configurations. First, we generate value activities *configurations*. These are connected value activities, by means of value exchanges, which represent a value model, *without* their performing actors. Because in this project we did not consider alternative deconstructions, we only have one such a configuration (essentially figure 6.8 with omitted actors).

Re-identify actors. Second, we re-identify actors, who are potentially interested in executing one or more value activities. Actors are potentially interested, if they expect to make a profit, or to increase utility by performing the value activity. Re-identification means that we consider new actors, which were not identified during development of the initial value model. It is reasonable to expect that by finding

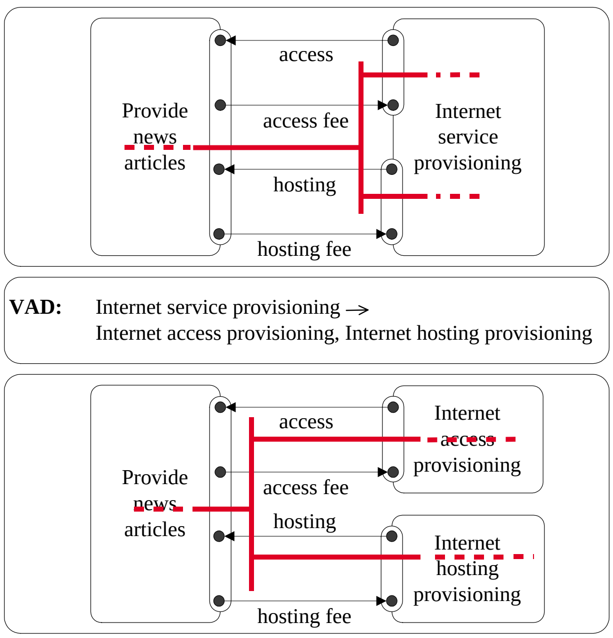


Figure 6.7: Deconstruction of the value activity *Internet service provisioning* into one for access provisioning and one for hosting provisioning, using the value interfaces deconstructed in figure 6.5.

Table 6.1: Actor - value activity matrix showing which actors can potentially perform which value activity, *and* thereby creating profit, or increasing utility by doing an activity.

Value activity	*Actor*				
	Reader	Last Mile	Data Runner	Hoster	Amsterdam Times
Read article	x				
Handle local loop traffic		x	x		
Handle long distance traffic		x	x		
Provide internet access		x	x	x	x
Hosting		x	x	x	x
Provide news articles					x

new, more specialized value activities, other actors than the ones already found are interested to perform these.

Re-assign actors. Third, we make an *actor-value activity assignment* matrix (see table 6.1). This matrix shows actors, which are potentially interested in performing value activities of a specific configuration.

Finally, using the actor-value activity assignment matrix, alternative value models can be extracted and represented using our graphical technique. Figure 6.8 shows one possible value model. Other models are possible by choosing other assignments of value activities to actors.

6.5 Deconstruction and reconstruction in the literature

6.5.1 Business science perspective

Deconstruction and reconstruction has been proposed by multiple authors in the field of business science. In this section, we present the vision of Timmers, Tap-

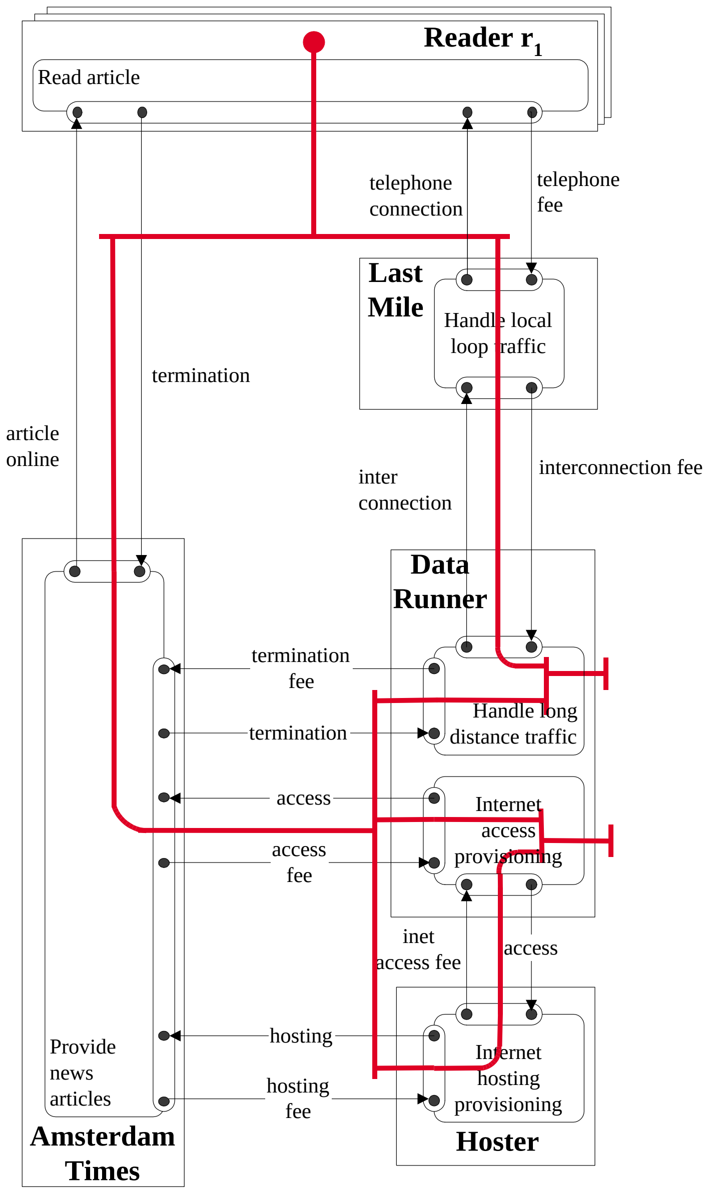


Figure 6.8: A reconstructed value model by assigning newly discovered value activities to actors.

scott, Ticoll, & Lowy, and Evans & Wurster, and relate their interpretation of deconstruction and reconstruction to the e^3-*value* approach.

Deconstruction and reconstruction according to Timmers

Based upon the value chain approach, Timmers (1999) introduces value chain deconstruction and reconstruction to find business models. Timmer's business models are similar to requirements expressed on our three viewpoints (see chapter 2). The aim of value chain deconstruction and reconstruction according to Timmers is to identify possible ways of integrating information along the value chain. To do so, Timmers proposes a framework consisting of: (1) value chain deconstruction, (2) interaction patterns, and (3) value chain reconstruction.

Value chain deconstruction. Value chain deconstruction is about identifying the elements D_i of a value chain, in terms of the kind of activities discussed by Porter (1985), such as inbound/outbound logistics, operations, marketing and slaes, service (the primary activities), and technology development, procurement, human resource management, and corporate infrastructure (the supporting activities). To have more detail, Timmers proposes to use business processes rather than value chain elements.

Interaction patterns. Timmers distinguishes four interaction patterns I: 1:1, 1:N: N:1, and N:M. For instance, an 1:1 interaction pattern represents that two value chain elements are integrated or combined with each other. Similarly, an 1:N interaction patters means integration or combination of N actors with one other actor.

Value chain reconstruction. Value chain reconstruction focuses on integration of information across a number of steps of the value chain. It is represented as $V(\{a\},\{b\})$, which means that value chain elements of party a and party b participate in integration or combination of information. Possible business models are then constructed by combining interaction patterns I_n with value chain reconstructions V_m. For instance, an electronic shop is about a single actor to a single actor (a 1:1 interaction pattern), with marketing and sales elements. On the other hand, an electronic mall built around a common brand offers many to one (N:1 interaction) marketing and sales.

Timmers and *e*^{3}*-value* . Compared to our approach, Timmers focuses on value chain elements (being the value activity elements as defined by Porter (1985)), while we focus on *e*^{3}*-value* value activities. The differences between both activities have been addressed in chapter 4: Porter's value activity needs not to be profitable, while an *e*^{3}*-value* value activity is supposed to be profitable. Moreover, Timmers does not discuss *how* to deconstruct and reconstruct a value chain but provides only a framework for doing so. In contrast, we provide a limited set of deconstruction operators, with a guideline how to apply them, as well as a guideline how to reconstruct a value model.

Deconstruction and reconstruction according to Tapscott, Ticoll, & Lowy

Tapscott et al. (2000) introduce the notion of *business web* (shortly called *b-web*) to discuss new value propositions for companies as a result of using internet technology. In terms of deconstruction and reconstruction, Tapscott et al. *disaggregate* and *reaggregate* a value proposition.

Disaggregation. To aggregate, Tapscott et al. first disaggregate a value creating system. Disaggregation entails: (1) identifying the key participants, (2) describing what each participant contributes to the system, and (3) pinpointing the weaknesses and opportunities for improvement in the current arrangement. Based on these disaggregated elements, then new b-webs are envisioned. As we do, Tapscott et al. acknowledge that this is a creative step, which requires that developers step outside their day-to-day mental models. The result should be one or more changed/new value propositions.

Reaggregation. Reaggregation entails reidentifiying value contributors and assigning value contributions to these. Contributors can be taken from the b-web started with, but is more likely that new contributors are needed, for instance for infrastructural propositions (e.g. hosting, access), commerce supporting functions like security and privacy services, transaction management, and logistics and delivery.

Tapscott et al. and *e*^{3}*-value* . Disaggregation is about decoupling the actors involved in a value proposition from how these actors add value. To do so, key participants are identified *as well as* their contribution to a value system. Reaggregation is about repopulating the categories of value contributors and assigning value contributions to these. In our approach we decouple value activities from performing actors, try to find variations by finding new value activities, interfaces and ports, and reassign newly found activities to actors. Decoupling actors from what they are doing is inspired on the work of Tapscott et al. However, the steps to do disaggregation are very loosely defined by Tapscott et al., in contrast to our approach, which defines a limited set of operators to do deconstruction.

Deconstruction and reconstruction according to Evans & Wurster

Deconstruction and reconstruction of business structures is mainly caused, according to Evans & Wurster (2000), by two factors: (1) separation of information and things, and (2) the blow-up of the tradeoff between richness and reach of information. Because the causes are fundamental for Evans & Wurster's opinion on deconstruction of business structures, we review these causes briefly below.

Separation of information and things. Many product and services sold are now a bundle of physical thing(s) and information. However, both information and physical things can each be of economic value for actors, and consequently may be sold/obtained separately also. Moreover, unbundling information and physical things can release extra economic value because bundling often is a comprise between the economics of information and economics of things. As an example, a shelf space in shop is a bundle of information and a product inventory: it gives *information* regarding the products it stores, but also serves as an *inventory*. Debundling these (e.g. by an electronic catalogue and a warehouse), may result in a maximalization of value for each function (e.g. an electronic catalogue may present more detailed information than can be done on a stock shelve).

Blow-up of the tradeoff between richness and reach information. If information is bundled with a thing, the reach and richness of information is to a certain extent determined by its carrier (the thing). A physical shelf space for instance reaches a limited audience, and so does the bundled information. By unbundling information from its physical carrier, the richness and reach trade-off can blow up (see figure 6.9). Richness means the quality of information, while reach is about the number of people participating in sharing the information.

Deconstruction. Evans & Wurster (2000) first debundle a value proposition into information and physical things. Also, additional (new) information products and services are created. Then, for the information-based products and services, it is investigated how the richness/reach trade-off can be blown up. As a result, the

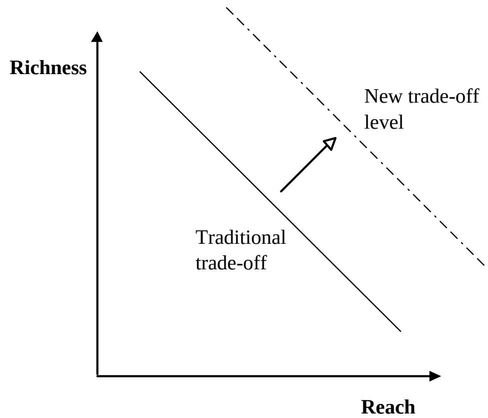


Figure 6.9: The blowup of the richness/reach trade-off.

role actors play (e.g. the kind of the activities they perform) may change and/or shift. Also, new actors may arise and others may appear as well relationships between actors themselves may evolve. Moreover, Evans & Wurstner present how deconstruction works in a number of cases, e.g. in value chains (disintermediation), in supply chains, and in organizations themselves.

Evans & Wurster and *e*^{3}-*value* . The idea of debundling valuable information from physical things can be compared to our value port deconstruction operator, followed by our value interface deconstruction operator. However, Evans and Wurster specifically focus on debundling of information and physical things as valuable objects. Furthermore, Evans and Wurster do not provide an ontology or at least a structure to describe value models, and consequently do not employ reconstruction operators. Therefore, it gives more direction in strategic thinking rather than that is usable in practical situations such as workshops on deconstructing value models with stakeholders.

6.5.2 Information technology perspective

By deconstructing value models, we aim to find basic building blocks, which we can use to configure new value models. Such a configuration like approach is also well known in the realm of information technology. We discuss two disciplines in which such a configuration oriented approach plays a domimant role: (1) knowledge engineering, and (2) patterns.

Knowledge engineering

Amongst others, knowledge engineering studies *problem solving methods*. These are represented by task models showing how an expert would tackle a particular problem. The *configuration* task is one of the knowledge intensive tasks considered by the problem solving method community. According to Schreiber et al. (2000) a configuration task consists of a number of subtasks: (1) operationalize requirements, (2) specify skeletal design, (3) propose design extension(s), (4) verify current configuration, (5) critique the current design, (6) select an action, and (7) modify the configuration. These tasks are largely based on Chandrasekaran (1990). Compared to our e^3-*value* approach, a skeletal design can be seen as a baseline value model consisting of a global and/or detailed actor viewpoint. The rest of the tasks focus on detailing (by extending) the skeletal design, verifying if the design complies with constraints, and if not, critiquing the design by indicating fixes for constraint violation. The main difference with e^3-*value* is that Schreiber et al. (2000) suppose pre-existing components, while our approach, by applying deconstruction operators, tries to find such components (e.g. new value activities and value objects).

In Motta, Stutt, Zadrahal, O'Hara & Shadbolt (1996) a configuration approach is used to find problem solving methods for configuration oriented tasks themselves. Initially, a configuration task is modeled on a coarse granularity level: a task with an input and output. Iteratively, this task is refined into more specific tasks (such as a synthesis or analysis task). Also, the inputs and outputs are specialized. For doing so, a number of rewrite operators are available. These can be compared to our deconstruction operators.

Patterns

A named pattern describes a problem which occurs over and over again in our environment, and describes one or more solutions for the identified problem as well as consequences (e.g. trade-offs) as a result of applying the pattern (see e.g. Gamma et al. (1997), Fowler (1997)). A solution mentioned in a pattern can be seen as a building block of a design for an information system (in case of a design pattern) or conceptual model (in case of an analysis pattern). An important characteristic of a pattern is that it provides one or more agreed proven solutions for a problem. Patterns are intended to facilitate the reuse of proven (design) knowledge.

Deconstructed parts of a value model can not be seen as a pattern. First, such a model fragment does not contain an explicit problem statement with solutions, and also it does not provide a decription of the consequences of applying the model fragment. Moreover, the fragment is not named, which is considered as of importance by the pattern community for the development of a shared vocabulary. Second, deconstructed model fragments are created on a per case basis, while for a pattern a solution mentioned by a pattern should be a proven solution to the stated problem. Therefore, patterns are not developed on a per case basis. Rather, they are found as a result of applying a same solution for a problem over and over in different designs, but in a similar context.

Value patterns, capturing a business problem, the context of the problem, solutions and consequences of applying it can support the design of innovative value proposition. However, finding such patterns for information technology intensive value propositions is not easy, if at all possible, at the time of writing. Formulating patterns requires *proven* solutions for problems. Such knowledge is currently hardly available for innovative e-commerce ideas.

6.6 Conclusions

Finding innovative value models is a creative task. However, finding variations on such a value model can be facilitated by value model deconstruction and reconstruction. The starting consideration for this is to separate the questions (1) which value adding activities exist from (2) which actors are performing these.

To find value model variations, we have defined three deconstruction operators, which all have a clear business rationale. The value activity deconstruction (VAD) operator helps in finding smaller value activities, which all can be profitably performed by at least one actor. We keep the value interface invariant using this operator, and only focus on the partitioning of a value activity over a number of actors rather than one actor.

A value interface models that an actor, or value activity, offers something of value to its environment, *and* wants something in return for that. The value interface deconstruction (VID) operator splits such interfaces into smaller ones. This may be done for two reasons. First, splitting can be done for unbundling reasons: the offering of value objects separately rather than as a bundle. Second, deconstructed value interfaces can be used to deconstruct a value activity associated with these interfaces into smaller activities.

Finally, the value port deconstruction (VPD) operator assists in identifying new value ports/objects, based on an initial one, which each can be delivered or requested by individual actors. Mostly, the VPD operator is followed by the VID

operator to address unbundling, and even by the *VAD* operator, to distribute the offering of the original value object over a number of actors.

Also, we have shown how these operators work out in a practical, non-trivial value modeling project. The e^3-*value* representation of value models appeared valuable in the project to illustrate complicated concepts such as call termination and interconnection to stakeholders, while the presented deconstruction and reconstruction process proved important to find new value activities, and to renegotiate assignment of these activities with the performing actors.

Chapter 7

Value modeling: the consumer value perspective

Previously, we have discussed evaluation as a necessary step to come from an e-commerce idea to a value model. Part of evaluation is the assignment of economic value to value objects. For each actor profitability sheets can be constructed, which show if there is a prospect on profit for enterprises or creation of economic value for end-consumers.

Enterprises (re)sell value objects to make profit, while end-consumers do not re-sell objects anymore, but consume, use, or possess them. Because enterprises and end-consumers have these different goals, they assign value to objects differently. Whereas for enterprise the net cash flow is of importance, end-consumers assign economic utility to a value object (Kotler 1988). However, how to calculate economic utility is not clear. In this chapter, we propose to use Holbrook's interpretive, qualitative framework on consumer value to do so. We extend Holbrook's approach with a practical approach that facilitates quantitative reasoning about consumer value and economic utility and construct profitability sheets for end-consumers. Also, we show how we employ evolutionary scenarios to reason about future events and wrong assumptions.

We exemplify consumer valuation and evaluation by a case study in the music industry. It is one of the industries heavily impacted by the increasing popularity of the Internet and digitalization of content. As a potential advantage for the industry, music can be digitally represented, and therefore can be bought *and* distributed via the Internet. However, a drawback is that the same Internet is also used to copy and obtain music illegally. During our consultancy practice, we have been involved in a project studying ways of selling and distributing music via the Internet for a consortium of large record labels. Also, for some years we have been advisor for one of the Dutch intellectual property right societies, responsible for collecting fees for using music. During these projects, it turned out that we should explain to the music industry that protecting digital content, e.g. using encryption technology is not the only way of addressing the piracy scene. Another approach is to exploit the mechanism of consumer value: if legal music has a higher consumer value than illegal music, consumers will likely prefer the legal option.

In this chapter we show how we use the e^3-*value* methodology, enriched with consumer value insights from marketing and axiology theory, to present the option of exploiting consumer value as an additional tool to fight the piracy scene. This chapter is structured as follows. In section 7.1 we briefly review ICT-dominated ways of protecting digital content. We do not argue that protection of content is unnecessary, but rather that creation of additional consumer value *and* protection of digital content should be seamlessly applied to selling music. In section 7.2 we discuss business-oriented ways to ensure that digital content is bought rather than illegally copied. One of these is the creation of additional consumer value. In section 7.3 we analyze illegal copying of music from two perspectives: (il)legality and consumer value. Section 7.4 introduces a practical attempt to quantify the consumer value contained in digital products. It is the foundation for section 7.5, which evaluates several scenarios for two prototypical consumer segments with respect to consumer value as a way to prevent illegal use of content. We show how these business scenarios help focus executive decision making. Finally, we present our conclusions.

7.1 Protection of rights on digital assets

Protection is a way to discourage the unintended use of digital content (such as copying, unauthorized resale and more), but is, as we will show, not sufficient to prevent a piracy scene, especially if the price of legal content is high enough. Various approaches for *protecting* the intended usage of digital content exist. We distinguish: (1) protection by encryption, (2) protection by watermarking, and (3) protection by law.

7.1.1 Protection by encryption

There are a number of systems, called Digital Rights Management (DRM) systems, available which support protection of digital content using encryption schemes (see Blaze et al. (1996), *Liquid Audio* (2001), *Intertrust* (2001), *Microsoft Windows Digital Rights Management (DRM)* (2001), and *A2B Music* (1999)). Also, standardization is underway for content protection (*Secure Digital Music Initiative* 2001). Digital rights management systems based on encryption technology offer facilities to *prevent* violations of the intended usage of the music but have a number of weak spots. First, the consumer can *always* make copies by resampling the analog output. There is a small quality loss but all subsequent copies can be made without any further loss. Second, the consumer can intercept the decrypted bitstream and save this stream in a file. Third, the encrypted content itself can be attacked.

7.1.2 Protection by watermarking

As can be concluded from the previous section, protection by encryption can be attacked successfully in a number of ways. Therefore, this way of protection must only be seen as a first line of defense. A next step is to *watermark* the content. A watermark can be used in court to *prove* violations of intended usage of the content. With watermarking technology it is possible to identify the digital content, to identify the original producer of the content, and to identify the consumer who bought the rights to use the digital content (Memon & Wong 1998). This information is important to prosecute violations of intended usage of content. However, in Craver et al. (1998), a number of successful attacks on watermarks are identified. First, robustness attacks aim to diminish or remove the presence of the watermark, without harming the digital content significantly. A number of successful attacks have been reported on commercial exploited watermarking techniques. Second, presentation attacks do not remove the watermark itself but instead manipulate it such that a watermark detector cannot find it anymore. Finally, interpretation attacks try to mislead watermark detectors by making multiple interpretations of a watermark possible. A popular approach is to insert a second watermark into the content, thereby creating a deadlock in the interpretation of the watermark. So, in conclusion, protection by watermarking is not the only way to go.

7.1.3 Protection by law

The last line of defense is to *prosecute* the person who violates the intended usage of digital content. Protection of digital content by law has a number of weaknesses. First, the law differs between countries. Laws of some countries offer more handles to prosecute illegal use of content than others do. Second, if the violator is in another country than the owner of the content (the prosecutor), it is difficult to prosecute the violator. Furthermore, suing itself does not scale up very well. If a large number of small violators exists (as is actually the case in music copying and downloading), it is not feasible to sue all these violators individually.

In conclusion, if digital content is to be sold, one should bear in mind that a consumer can violate the intended usage of the content, sometimes rather easily. This remains true also when various protection schemes have been applied. Especially if the motivation of the consumer is high enough, s/he is able to obtain digital content from sources other than the legal ones. Hence, protection of digital content alone is not sufficient to address the problem of misuse of digital content.

7.2 Exploiting rights on digital assets

In contrast to prevention of illegal use of music by protecting music assets, *other ways of exploiting music* can be a solution to ensure that intellectual property rights owners are paid for their work. These are inspired on work done by the economics community (see e.g. Shapiro & Varian (1999) and Choi et al. (1997)).

7.2.1 Business strategies

A number of strategies can be thought of: (1) exploiting interactivity in content, (2) exploiting time dependence of content, (3) creating multiple versions of content, and (4) bundling content with physical products, or with services. We discuss these strategies below.

Interactivity. If products can be thought of which require interactions with a consumer, payment can be coupled with this interaction. An example of such a product is a computer game. The player determines the flow of the game from a number of possibilities by interacting with the game. To continue the game, we can think of an approach that an additional piece of software needed for continuation should be retrieved from a content supplier, of course after payment. Unfortunately, music products hardly require any interaction.

Time dependency. Some products with no interaction (e.g. news) can exploit time dependency. Such products decrease in value substantially in a few days. The incentive to copy these products illegally is low. However, music is characterized by a very slow decrease in economic value over time; in contrast, over the years some music tracks become more valuable than that they were at release time.

Versioning. Another way to exploit digital content is to create multiple versions, for example a number of remixes of a music track, or different quality levels of images. However, the number of versions a consumer can choose from is usually very limited, and therefore illegal copies of versioned content will become easily accessible as well.

Related sales and bundling. A more extreme position is to sell complementary related products which cannot easily be copied such as merchandise of artists, while the content itself is nearly for free. In such a scenario, the digital content plays only the role of attracting consumers to a site: the revenues should come from related sales. A variation on this theme is *bundling*: a consumer can only buy merchandise if s/he also buys the associated digital content. A general limitation of these business-oriented approaches is that there are many cases of digital products (e.g., 'classic' songs and movies) that maintain their value over long periods of time. People really want to obtain these assets (and do not want to buy associated products), and they want to do so for a long time (the effect of a decreasing value of the asset over time is limited). Hence, it is important to analyze the concept of consumer value contained in digital content *itself*, and not solely consider the generation of revenues from complementary products and related sales.

7.2.2 Exploiting dimensions of consumer value

We thus want to explore how to exploit the value of digital content itself, in such a way that it creates a value gap between legal and illegal providers of digital content. We suggest that recent 'interpretive' marketing research on consumer value gives some useful initial handles on this topic. In particular, we use Holbrook's value framework (Holbrook 1999) that investigates different aspects or dimensions of value resulting from the consumption experience of a product.

Extrinsic and intrinsic value. In his framework, Holbrook makes a distinction between the *extrinsic* and *intrinsic* value of a product. A product has an extrinsic value component if a consumer uses the product to accomplish some goal that is outside the consumption of the product itself. For example, a consumer values a hammer mainly because it can be used to drive in a nail, rather than that s/he values the hammer in its own right. In contrast, something is valued intrinsically if the consumption experience is valued for its own sake. For example, music has an important intrinsic value component because listening to music, the experience, is of value by itself. In fact, the digital content considered in this chapter relates

Table 7.1: Value types in Holbrook's framework.

Value dimension	*Extrinsic*	*Intrinsic*
Active	EFFICIENCY (I/O function, convenience)	PLAY (fun)
Reactive	EXCELLENCE (quality)	ESTHETICS (beauty)

to the *right* to have a, hopefully appreciated, experience. The bits are only the representation of the music that enables the experience.

Active and reactive value. Another dimension introduced by Holbrook is that value may have an *active* or *reactive* component. A product with an active value component requires that a consumer actively does something with the product (for example, using a music track for karaoke singing) as part of the consumption experience. Consumer value is called reactive if the product itself accomplishes something to or with a consumer as a result of a consumption experience, such as listening to music passively.

Putting together the 2×2 combinations from these two dimensions of consumer value yields four types of value, as shown in table 7.1. Below, we show how such a value typology can be used as an aid in uncovering which different e-commerce parameters influence consumer value. In addition, we will quantify these value parameters, and analyze their effect on e-commerce value model design through a collection of realistic business-consumer interaction scenarios.

7.3 Legality versus value creation

We do not suggest that protection of digital content is irrelevant. On the contrary, such a first barrier prevents a number of consumers from committing an illegal act, and makes them aware that unintended use of the digital content is prohibited. However, we do claim that rethinking and redesigning the value to the consumer of a digital content service (e.g. the right to listen to a music track once) can contribute to reducing the illegal ways of consumption. We can exploit the fact that a digital product has valuable aspects in addition to the actual content itself, cf. the Holbrook value typology. For example, convenience in selecting and ordering, receiving the content without delays, enhancing fun by different options to interact with the digital content may all be of value to the consumer. In section 7.4 and 7.5, we analyze the multiple aspects of value created by digital content in more depth.

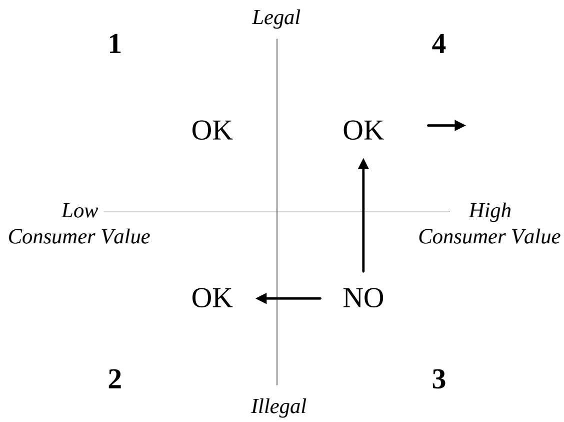


Figure 7.1: Positioning value models: (il)legal content versus created consumer value.

Figure 7.1 shows how various e-commerce value model options may be positioned in a design space spanned by the degree of legality and the degree to which consumer value is created. The first quadrant, digital content that is legal but offered with a low consumer value, is not interesting from a business point of view. Illegal content with a low consumer value, the second quadrant, is not likely to be very popular with consumers either. If for example the convenience is low, consumers will not be attracted to obtain the illegal content. Thus, offerings in this quadrant can be left alone (note also that from a business point of view, technical or legal protection measures are not really needed here).

The third quadrant, illegal content with a high consumer value, is highly unwanted, however. As indicated in figure 7.1, there are ways to make the e-commerce value 'models' positioned in this quadrant less attractive. Illegal content with high consumer value requires high visibility and accessibility in a market. If not, it takes too much effort for consumers to find and select the product. Furthermore, it must be easy and convenient to obtain and consume the content. However, visibility and a high-quality fulfillment infrastructure enable content owners to take corrective action, for example to prosecute suppliers of illegal content or to ask legally operating Internet Service Providers to remove or block illegal content. Such measures do not remove illegal offerings entirely, but result in illegal content with lower consumer value, thus moving illegal offerings from the undesirable quadrant 3 to the uninteresting quadrant 2 (in other words, these measures generate utility destruction). Alternatively, suppliers of such illegal content may decide to set up a legal operation and move up to quadrant 4. This quadrant represents the desired

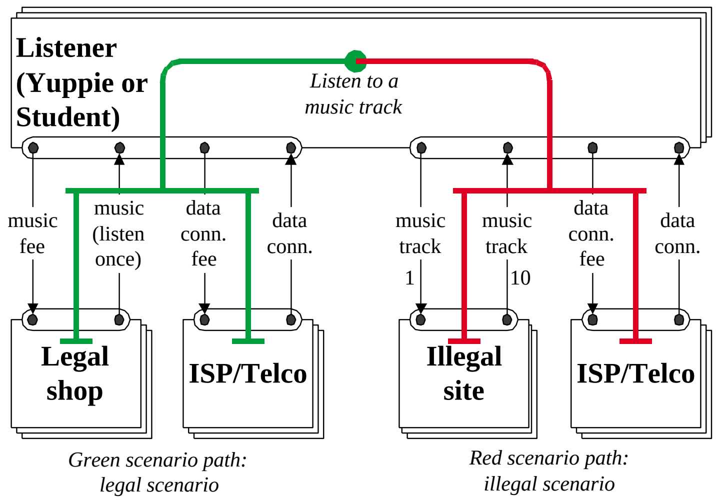


Figure 7.2: Two scenario paths for satisfying a need: a legal path (green) and an illegal path (red).

situation: providing legal content with a high consumer value. Finally, Figure 7.1 shows that legal content providers already in this quadrant may strive to increase the consumer value created by the digital products they offer.

In sum, the approach is *to increase the value gap* between legal and illegal objects.

7.4 An e-commerce idea for selling music

Music can come in various ways, which may differently perceived and valued by an end-consumer. For instance, listening to a music track broadcast by a radio station differs from listening to a downloaded music track. In the first case, the listener has only a limited influence on the music track s/he listens to, while in the second case, the listener selects a track s/he wants to listen to, downloads it, and plays it. The value proposition we use in the remainder of this chapter, is the right to listen *once* to a selectable, downloadable music track. We assume a non-streaming service: the track has first to be downloaded completely before the consumer can listen to it. Figure 7.2 presents for this proposition a value model including operational scenarios in the form of a use case map.

The value model shows two kinds of end-consumers: (1) a yuppie: a consumer characterized by enough monetary resources but with a lack of time, and (2) a student: a consumer who has scarce monetary resources but enough time. Both these kinds of end-consumers want to listen to a music track, and to do so, they can decide to obtain music from a legal shop, *or* from some illegal website containing music tracks. In case music is bought from a shop, a fee has to be paid; if an illegal site is used for need satisfaction, for ten tracks obtained, one other track should be offered. These are so-called *ratio* sites. Regardless of a decision for a legal or illegal track, end-consumers always need data communication facilities for transferring the music track. Therefore they pay a data connection fee. Data communication facilities are offered by some Internet Service Provider (ISP)/Telecommunication actors, which we do not consider in detail.

We distinguish two scenario paths for this value model: (1) a legal scenario path (colored green), modeling that a listener buys a track, and (2) an illegal scenario path (colored red), modeling that a listener obtains a track from an illegal website. Both paths can be executed by a yuppie and a student.

7.5 Evaluation of an e-commerce idea

As discussed in section 5.6 evaluation of an e-commerce idea consists of: (1) creation of a profitability sheet, (2) assignment of economic value to value objects, and (3) evaluation using evolutionary scenarios. We will discuss all these steps in the coming sections to argue that in some cases legal obtainment of music is prefered over illegal copying.

7.5.1 Creation of a profitability sheet for listeners

Value objects, and receipts and sacrifices

We will now analyze which factors play a role in the various types of consumer value creation, how they can be quantified, and how they can be used as 'control parameters', so to speak, to design an optimally positioned e-commerce value model.

In marketing literature (e.g., Holbrook (1999), Heskett et al. (1997), and Zeithaml (1988)), consumer value is often stated in terms of a *value equation*:

$$Consumer\ Value = \frac{\sum_{i=1}^{n} Receipt_i}{\sum_{j=1}^{m} Sacrifice_j}.$$

Sacrifices comprise all expenses a consumer has to make to consume the product; receipts represent the sum total of the benefits s/he experiences from consuming the product. The consumer will only buy the product if the consumer value ratio is greater than one; otherwise, a consumer decides not to buy the product at all, because the sacrifices outweight the receipts.

Receipts and sacrifices are not necessary the same as value objects flowing into or out an actor. Exchanging a value object may *cause* receipts and/or sacrifices for an end-consumer. Below, we discuss receipts and sacrifices as a result of exchanging (1) value objects representing a *fee*, and (2) all *other* value objects, being end-consumer experiences, which are valued using value types of Holbrook's framework.

Sacrifices based on value objects representing fees

If a consumer wants to listen to a track s/he has to pay money directly to others, called fees. These fees are part of the sacrifices mentioned in the value equation. Here, we distinguish (1) the *data connection fees* to be paid to a telecommunication company and/or Internet Service Provider, and (2) the *music fee* to be paid for the right to listen to the music track itself. The fees can be seen also in figure 7.2 as value objects.

Data connection fee. We consider (1) connection fees for selecting content, (2) for downloading content, and (3) for uploading content. The latter fees may appear if a consumer obtains the content illegally. Many illegal sites use a *ratio* scheme. Such a scheme requires that a consumer first *uploads* a music-track, after which s/he can *download* tracks of his/her choice.

Music fee. If the consumer buys the music legally, a fee is paid for the right to listen to the track. We assume this price is known and set by the supplier. In the illegal case, the price is Euro 0.00.

Receipts and sacrifices based on other value objects

There is an one-to-one relation between a fee to be paid by an end-consumer, and a sacrifice for an end-consumer. Obtaining a value object not representing a fee can both cause sacrifices *and* receipts for an end-consumer. For the case at hand, the end-consumer obtains two value objects s/he must value: (1) a music track (either legal or illegal) that is played once, and (2) a data connection which is used to up/download a music track. Both objects are needed to have a consumer experience: listening to a music track (once) chosen by an end-consumer. Therefore, we

Table 7.2: Value parameters for a *listen-once* experience.

Value dimension	*Extrinsic*	*Intrinsic*
Active	selection time upload time download time	interactive track play
Reactive	presentation quality	track beauty

value this experience using Holbrook's framework, rather than the objects themselves.

Table 7.2 shows factors structured according to the discussed Holbrook consumer value framework, for the experience *listen-once to a selected track of music*. Such parameters can contribute to either sacrifices or receipts, depending on the valuation by the consumer.

Selection time. Selection time is the time it takes for a consumer to search for and select a track of music s/he wants to listen to. We assume that a consumer already *knows* the title of the music track as well as the performing artist before selection; the selection-time only indicates the time necessary to find a supplier offering the *downloadable* track. The track should be downloadable because in the piracy scene, it does happen in practice that a site indicates that a particular track is available, but the track itself has disappeared. In such a case, the consumer has to spend additional time to find a new site that offers the track, which increases the total selection time for the track.

The selection time is an important instrument to fight piracy. If the selection time is low for music tracks of illegal content providers, such providers have high visibility and reliability. This enables legal providers, content owners and right organizations to fight such pirates.

Upload time. On an illegal ratio-based site, a music track must be uploaded first before one can obtain one or more tracks. The upload time is the time necessary to complete the upload and to gain the rights for one or more downloads. This includes the time necessary to obtain the track for upload from another medium such as a CD. We assume that uploading and downloading occurs sequentially, and consumes all bandwidth available.

Download time. The download time period starts when a consumer decides to listen to a previously selected track (legal scenario), or when an upload is complete (illegal scenario), and ends when the track is ready for play at the consumer site. The download and upload time depend on factors such as the available bandwidth. A legal provider can influence this factor positively, for instance by co-locating its content server with the Internet access points of its consumers.

Presentation quality. The presentation quality of music is determined by the *bitrate* of the music track. For consumers, perceived quality may be expressed in terms such as CD quality, near-CD quality, radio quality, and telephony quality. A legal provider can influence this parameter by consumer-selectable presentation quality options.

Interactive track play. The aspect of play, as identified in Holbrook's framework, refers to possibilities for the consumer to actively interact with the product. This interactivity should be of value for its own sake. For the *listen-once to a selected track of music* product we define the play aspect as the presence of functionality to turn on and off music instruments and vocals, allowing consumers to produce the vocals themselves (as in karaoke), or select alternative instruments and vocals so as to create their own version of a song. Such functionality is for example offered by the website of David Bowie (see Bowie (2000)). Legal providers can exploit this creative play and fun element, initially because they can obtain access to alternative instrument and vocal recordings, and subsequently by providing different versions of these. This boils down to a interactive versioning approach as discussed in section 7.2.

Track beauty. Finally, there is the aspect of beauty, implying that the music itself is valued as a consumption experience for its own sake by the consumer.

We note that we have introduced several independent parameters relevant to value creation. For example, a consumer may like the interactive play element of a David Bowie song, because it gives you the possibility of acting as a creative designer making a new instrumentation of a song, but s/he may not actually like David Bowie's music.

Calculating receipts and expenses

For evaluation purposes on a consumer value basis, it is necessary to calculate fees (which can be done in a rather objective fashion) as well as the different Holbrook value type aspects (of which the valuation is more subjective with respect to the consumer). As an example, consider the valuation of a short download time by the consumer. One part of this stems from the objective expected download time, which depends on the size of the track in bits and the available bandwidth. Another part may be formulated as an *inconvenience fee* in Euro/second incurred by the consumer, expressing that the utility of absorbing consumer time also has to be taken into account (as a more subjective, and consumer segment-dependent opportunity cost or nuisance value component). This utility quantification of the various objective and subjective factors is presented in table 7.3. To calculate consumer value, we use the following measurable quantities:

- the *bitrate* (bits/second) used to represent the content in a digital way;
- the *duration* of a track in seconds;
- the *bandwidth* (bits/second) available to stream content to the consumer;
- the fee for an ongoing data connection or *ticks* (Euro/second) and a *connection setup fee* (Euro/connection setup), to be paid by the consumer for a connection to the Internet;
- in the case of an illegal provider, the *ratio* between uploaded and downloaded tracks. The ratio is the number of tracks which need to be uploaded before a consumer can download one track of his choice.

7.5.2 Assignment of economic value to sacrifices and receipts

Assumptions

For the evaluation of the yuppie and student scenarios, we assume values for the consumer utility parameters as summarized in table 7.4. Values for some parameters differ between the legal and the illegal case. The rationale for this is that by carefully influencing or controlling such parameters, a legal provider has an opportunity to create additional consumer value. This especially holds for the available bandwidth, selection time, and price. By fighting piracy effectively, the search time for illegal providers can be increased, resulting in a lower consumer value of illegal content. However, some parameters cannot be easily influenced by content providers such as the costs for telecommunication.

The values of the consumer utility parameters are, where possible, based on realistic empirical estimates. We assume that an illegal site offers only 50% of the

Table 7.3: Calculation of receipts and sacrifices.

Fees	*Calculation*
Data connection fees:	
selection time	$selection\text{-}time \times ticks + setup\text{-}fee$
download time	$\frac{bitrate \times duration}{bandwidth} \times ticks$
upload time	$\frac{bitrate \times duration}{bandwidth} \times ticks \times ratio$
Music fee	*determined price by supplier*
Listen once experience	*Calculation*
Inconvenience:	
selection time	$selection\text{-}time \times inconvenience\text{-}fee_{consumer}$
download time	$\frac{bitrate \times duration}{bandwidth} \times inconvenience\text{-}fee_{consumer}$
upload time	$\frac{bitrate \times duration}{bandwidth} \times inconvenience\text{-}fee_{consumer} \times ratio$
presentation quality	$f_{consumer}(bitrate)$ assume near CD for legal and illegal scenarios
interactive track play	$f_{consumer}(availability)$ assume available for legal scenario, not available for illegal scenario
track beauty	$f_{consumer}(content)$ assume equal for legal and illegal scenarios

Table 7.4: Parameter values for the yuppie and student scenarios.

Consumer utility parameter	*Illegal case*	*Legal case*
selection time	60 s	30 s
bit-rate	128 kb/s	equal
mean duration of track	240 s	equal
bandwidth	30 kb/s	60 kb/s
ticks	Euro 0.01/minute	equal
connection setup fee	Euro 0.05/setup	equal
ratio	0.1	0

bandwidth a legal site offers to its consumers. This bandwidth is measured end-to-end: from music supplier to consumer. The bandwidth is therefore constrained by the bandwidth offered by Internet Service Providers to their end users. We take, for the legal case, a value of 60 kbit/s, which is possible using ISDN. A content provider can fully exploit this bandwidth if its content servers are co-located with the access servers of the ISP. The values for *ticks* and *connection setup fee* are taken from the current standard tariffs of a large Dutch telecommmunication company. We assume that Internet access itself is for free, as is the case in the Netherlands. For the *ratio* we assume a value of 1:10, which is often seen on illegal sites. For *bitrate* we assume a value which is currently typical for MP3 tracks on the Internet.

A profitability sheet for a yuppie

Tables 7.5 (legal case) and 7.6 (illegal case) illustrate a valuation of the experience *listen-once to a selected track of music* by yuppies. We have chosen hypothetical but reasonable values, using the following approach. First, a consumer values the presentation quality for the legal and illegal case equally, because for both cases a consumer values the same track of music. The same holds for the track beauty aspect. Second, we assume that the consumer ranks the value of Holbrook's aspects in the following order (from high to low): (1) the beauty aspect (the first priority is to listen to a particular track of a selected artist), (2) the presentation quality aspect, and (3) the interactive play capability. Further, we have assumed that the yuppie's inconvenience fee is Euro 1.-/hour. Of course, this is an example for which it is difficult to get accurate numbers. However, an important point to note is that these numbers are not intended for exact value calculations *per se*. Instead, we are interested in the much more modest goal of *relative* statements, drawn from a comparative analysis and a sensitivity analysis of relevant business scenarios. As we will see, it is indeed possible to come to strategically relevant conclusions from a quantified analysis based on rough, order-of-magnitude, numbers. This is all we aim for in this chapter. From tables 7.5 and 7.6 can be seen that a yuppie would choose for legal music (consumer value is 1.03) rather than illegal music (consumer value is 0.61).

A profitability sheet for a student

The student profitability sheets (see tables 7.7 and 7.8) assume that the student's inconvenience fee is Euro 0.10/hour (one order of magnitude lower than the yuppie's inconvenience fee). We keep all other values the same. The consumer value of the illegal case now becomes 1.25, while the value of the legal case is 1.57.

Table 7.5: Yuppie consumer profitability sheet, legal case.

Actor	Yuppie			
Scenario	Listen to a track once			
	Receipts		*Sacrifices*	
Scenario path	Legal			
Value object:			Music fee:	0.10
Value object:			data connection fee:	
			- selection:	0.055
			- download:	0.085
			- upload:	-
Value object:	Music + data connection			
			Inconvenience:	
			- selection time:	0.0083
			- download time:	0.14
			- upload time:	-
	Presentation quality:	0.15		
	Interactive play:	0.050		
	Track beauty:	0.20		
Consumer value equation results				
Total Receipts and Sacrifices		0.40		0.39
Ratio Receipts / Sacrifices	**1.03**			

Table 7.6: Yuppie consumer profitability sheet, illegal case.

Actor	Yuppie			
Scenario	Listen to a track once			
	Receipts		*Sacrifices*	
Scenario path	Illegal			
Value object:			Music track:	-
Value object:			data connection fee: - selection: - download: - upload:	 0.060 0.17 0.017
Value object:	Music + data connection			
	Presentation quality: Interactive play: Track beauty:	0.15 - 0.20	Inconvenience: - selection time: - download time: - upload time:	 0.017 0.28 0.028
Consumer value equation results				
Total Receipts and Sacrifices		0.35		0.57
Ratio Receipts / Sacrifices	**0.61**			

Table 7.7: Student consumer profitability sheet, legal case.

Actor	Student			
Scenario	Listen to a track once			
	Receipts		*Sacrifices*	
Scenario path	Legal			
Value object:			Music fee:	0.10
Value object:			data connection fee:	
			- selection:	0.055
			- download:	0.085
			- upload:	-
Value object:	Music + data connection			
			Inconvenience:	
			- selection time:	0.00080
			- download time:	0.014
			- upload time:	-
	Presentation quality:	0.15		
	Interactive play:	0.050		
	Track beauty:	0.20		
Consumer value equation results				
Total Receipts and Sacrifices		0.40		0.26
Ratio Receipts / Sacrifices	**1.57**			

Consequently, for consumer segments that incur a low inconvenience fee (that is, they are willing to spend their own time) illegal offerings become relatively more attractive.

7.5.3 Evolutionary scenarios

Evolutionary scenarios for the yuppie

Several variations on the profitability sheet for the yuppie (see tables 7.5 and 7.6) are interesting to analyze; they are motivated by expected changes that are likely to occur in the future or wrong estimations: (1) nearly equal end-to-end bandwidth for the illegal and illegal case, (2) an increase of the overall bandwidth without

Table 7.8: Student consumer profitability sheet, illegal case.

Actor	Student			
Scenario	Listen to a track once			
	Receipts		*Sacrifices*	
Scenario path	Illegal			
Value object:			Music track:	-
Value object:			data connection fee: - selection: - download: - upload:	 0.060 0.17 0.017
Value object:	Music + data connection			
	Presentation quality: Interactive play: Track beauty:	0.15 - 0.20	Inconvenience: - selection time: - download time: - upload time:	 0.0017 0.028 0.0028
Consumer value equation results				
Total Receipts and Sacrifices		0.35		0.28
Ratio Receipts / Sacrifices	**1.25**			

changing costs, (3) changes in the play factor of the product, (4) changes in the consumer's inconvenience fee, and (5) a service extension such as repeated listenings to the same track.

Scenario A1: The bandwidths of the legal and illegal sites become nearly the same. It is possible that the music industry is not sufficiently capable of fighting the illegal scene, as previously discussed in section 7.3. Then, a consequence may be that illegal sites are offering music with nearly the same bandwidth as legal sites. If we assume for the illegal site a bandwidth of 50 kbit/s, the consumer value equation ratio for the illegal case becomes 0.93 instead of 0.61, i.e., close to the value for the legal case. If bandwidths are equal (60 kbit/s) the illegal offering is even favored over the legal one in terms of consumer value. Thus, the bandwidth difference is an important parameter to create a value gap between the legal and illegal cases.

Scenario A2: The bandwidth increases. In the near feature, it is reasonable to expect an increase of available bandwidth nearly without any change in costs. Developments such as xDSL, which offer a high bandwidth connection (order 1 Mbps) over the local loop of a telecoms operator, are now commercially available. A bandwidth increase will heavily cut down both the out-of-pocket and inconvenience sacrifices, especially those related to download times. Compared to the valuation in tables 7.5 and 7.6, a bandwidth increase above about a factor of 5 (both for illegal and legal bandwidth) will start to favor the illegal site over the legal site. Therefore, a differentiation in bandwidth only (scenario A1) is not sufficient in the long run as a means to fight piracy. Because this scenario is very likely to happen in the near future, we analyze the following scenarios in conjunction with this scenario.

Scenario A3: The selection time for the illegal case increases substantially. If the music industry is successful in fighting piracy, the selection time for illegal tracks increases. For instance, if it takes 600 seconds to find a downloadable illegal track, the consumer value of the illegal scenario is 0.43 instead of 0.61, whereas the legal case remains the same at a value of 1.03. Moreover, if we additionally suppose that scenario A2 occurs, the consumer value of the illegal case becomes 0.82, while the legal scenario results in a consumer value of 2.46. Consequently, differentiation in selection time is a powerful instrument to have consumers favor the legal offering.

Table 7.9: Yuppie valuation of subsequent listenings.

Consumer Value	*Illegal case*	*Legal case*	*Total revenue*
1 listening	0.61	1.02	0.10
2 listenings	1.21	1.62	0.20
4 listenings	2.42	2.32	0.40
10 listenings	6.06	3.10	1.00

Scenario A4: The inconvenience fee is nonlinear. In our model, we assume that the yuppie uses a flat rate for his inconvenience fee. However, it might be more appropriate to assume that the costs associated with waiting for a music track grow more than linearly with time. In this way, we model the likely situation that a consumer wants to have the music fast, and if it takes too long, s/he is not interested anymore. If the inconvenience fee during the first 5 minutes is Euro 1.- per hour, during the second 5 minutes is Euro 5.- per hour, and is Euro 25.- per hour beyond that, the consumer value for the legal case is 0.64, but for the illegal case 0.086. If we analyze scenarios A2 and A4 in combination, the consumer values of the illegal and legal cases are about equal (1.89 vs. 1.88). If also scenario A3 occurs (selection time differentiation), the consumer will however prefer the legal case (0.46 vs. 1.88).

Scenario A5: Repeated listenings of the same track. Our valuation in tables 7.5 and 7.6 is based on a *pay-per-listen* product. However, for content such as music and video, *repeated* consumption occurs frequently. A consumer then listens to the same track of music a number of times. If in such a case the consumer stored the music-track locally after the first initial download, connection costs are zero for the subsequent listenings and so are Holbrook's inconvenience factors.

In our valuation in tables 7.5 and 7.6, a supplier of legal content differentiates him/herself from an illegal supplier by offering a fast download service (more bandwidth), so that the legal consumer saves data connection expenses. However, for subsequent listenings, no downloads are necessary if the content is stored locally, and the advantage of a fast download service becomes less significant. Moreover, selection does not introduce additional sacrifices and there are no extra inconvience costs. Table 7.9 presents the effect of subsequent listenings on consumer value.

From this table it can be concluded that if a yuppie expects to listen to a track four times or more, it becomes attractive to obtain the track illegally. A way to deal with this issue is to use a nonlinear pricing scheme. In table 7.10, the price of *n*

Table 7.10: Yuppie valuation of subsequent listenings using a nonlinear pricing scheme.

Consumer Value	*Discount Factor*	*Illegal case*	*Legal case*	*Total revenue*
1 listening	1.00	0.61	1.02	0.10
2 listenings	0.90	1.21	1.69	0.18
4 listenings	0.50	2.42	3.26	0.20
10 listenings	0.21	6.06	8.07	0.21

subsequent listenings is calculated as follows:

$$price_{n\text{-}subs.\text{-}list.} = discount\text{-}factor_{n\text{-}subs.\text{-}list.} \times price \times number\text{-}of\text{-}subs.\text{-}list.$$

This scheme assumes that the price per listening depends on the number of times the end-consumer listens to a track: it is per listening cheaper to listen to track a number of times, than only once. Using such a nonlinear pricing scheme, the yuppie will be encouraged to buy the music legally. The drawback of such a scheme is that, after two subsequent listenings, hardly any marginal revenues are generated. If we assume that scenario A2 also applies, the illegal offering becomes attractive. However, if scenario A3 occurs in addition, the legal offering has a higher consumer value. Application of scenario A4 strengthens this conclusion.

In sum, nonlinear pricing is a useful mechanism to stimulate legal use of music. Bandwidth differences only help in the short run. Selection time differences turn out to be a key to create a significant value gap between legal and illegal offerings.

Evolutionary scenarios for the student

The same set of evulationary scenarios can also be applied to the student's case. These scenarios have the following effects.

Scenario B1: The bandwidth of the legal and illegal site is nearly the same. A lower inconvenience cost results in a lower fee for waiting on a download. Therefore, the difference of bandwidths between the illegal and legal case is of less importance compared to the yuppie scenario. If the bandwidth of the illegal provider is 41 kbit/s and the bandwidth of the legal provider remains 60 kbit/s, the consumer will already opt for the illegal provider, while in the yuppie scenario bandwidths should be nearly equal.

Scenario B2: The bandwidth increases. Because of a lower inconvenience fee, an increase of available bandwidth by a factor of about 2 is already sufficient to favor the illegal case over the legal one. Therefore, bandwidth cannot be exploited very successfully in the student scenario to create additional consumer value.

Scenario B3: The selection time for the illegal case increases substantially. A selection time of 600 seconds for the illegal case makes that the sacrifices outweight the receipts, favoring the legal offering. This is also the case if we assume both scenario B2 and B3.

Scenario B4: The inconvenience fee is nonlinear. If the inconvenience fee during the first 5 minutes is Euro 0.1/hour, the second 5 minutes is Euro 0.5/hour, and beyond that is Euro 2.50/hour, the consumer value for the legal case (1.43) is higher than the consumer value for the illegal case (0.56). If bandwidth is no issue (scenario B2, with a 5 times increase of bandwidth), the illegal case will be chosen by the consumer.

Scenario B5: Repeated listenings of the same track. In case of repeated listenings, we find that for two listenings and more, the student chooses to obtain the music illegally. A nonlinear pricing scheme as discussed previously ensures that a student obtains music legally if the discount factor as presented in table 7.10 is 1 (1 listening), 0.82 (2 listenings), 0.41 (4 listenings), and 0.17 (10 listenings). A nonlinear pricing scheme plus scenario B2 results in a preference for the illegal case, but scenarios B2, B3 and B4 together favor the legal case.

In sum, our scenario analysis shows that for both consumer segments, selection time differences are a key parameter that must be controlled in order to create a significant value gap between legal and illegal offerings. Nonlinear pricing also is a useful ‘control parameter’ to make legal offerings attractive to the consumer. The difference between the student and yuppie consumer segments is that for the former, illegal offerings become attractive more quickly due to the lower inconvenience fee. Bandwidth differences only have short-term relevance, because the bandwidth itself is likely to increase strongly in the near future.

7.6 Lessons learned

To discuss lessons learned, we take two perspectives. First we present lessons learned which relate directly to the e-commerce idea *selling a right to listen once* *to a selected music track* to stimulate legal use of music. Second, lessons are discussed which focus on the use of consumer value theory for evaluating an e-commerce idea.

7.6.1 e-Commerce idea perspective

Lesson 1: *Exploiting end-to-end bandwidth to stimulate legal use of music has only an effect on the short term.*
Exploiting sufficient end-to-end bandwidth as a way to minimize download time, and thereby inconvenience to stimulate legal use of music, only has a short-term effect. This is caused by expectations that end-to-end bandwidth will grow substantially, both for legal and illegal services. If the bandwidth passes a certain threshold, it is not a good way anymore to create additional value for legal providers.

This lesson was a result of evaluating scenarios A/B2 on future overall bandwidth increase. At the time this study was carried out (November 1999 - February 2000), Internet access for end-consumers (in The Netherlands) was mainly based on the analogue (POTS) lines and ISDN (with a maximum of 2×64 kbits/s). Nowadays (November 2001), we see more and more Internet access via xDSL or cable technology allowing for higher bandwidths. So, the evolutionary scenarios A/B2 on bandwidth increase come true.

Lesson 2: *Exploiting a short search time for legal content (compared to illegal content) and thereby a low inconvenience fee has an effect on the long term.*
If our estimates on the inconvenience fee or ‘nuisance value’ of long waiting times are order-of-magnitude correct, this promises to be an effective barrier to inhibit consumers from obtaining illegal content. However, to make more reliable estimates on the effect of search time, more knowledge is needed about the inconvenience fee of actors of different market segments

This lesson is based on scenario A/B3. Indeed, nowadays, one of strategies followed by the music industry is to submit to illegal sites music tracks of famous artists, but which contain noise. Such actions increase the search time for illegal sites substantially, because end-consumers discover after downloading that they have a useless track. However, as a result of peer-to-peer networking (see e.g. *Gnutella* (2001), *Morpheus* (2001), Clarke et al. (2000), and Oram (2001)), a huge amount of music tracks is available. Moreover, the content of this ‘music-library’ is very dynamic and distributed: sites are appearing and disappearing within a few days, and distributed over a large number of privately hosted sites. By using an efficient search mechanism, it is still possible to locate music quickly. This dynamic behavior makes it very difficult for the music industry to fight piracy. Also, the large number of private sites hosting content makes prosecution difficult. In sum, exploiting an inconvenience factor to decrease consumer value of illegal music may be worthwhile, but is still hard to implement.

Lesson 3: *Subsequent listenings should be priced in a non-linear way.*
In the specific case of *listening only once* it is possible that end-consumers will optimize their decisions for sub-sequent listenings (listening to a same track twice or more). To be competitive with illegally obtained music, a legal variant should price subsequent listenings non-linearly. More specific, if a listener obtains music a second time or more, the price to be paid should decrease. After a number of subsequent listenings (e.g. 10), it is meaningless to ask a fee for more subsequent listenings. This brings a model for *listening once* closer to a model for *listening a (unlimited) number times*.

Lesson 4: *The price for listening music once should be low.*
Although price calculations are not intended to be exact in this chapter, we feel pricing should be more in regions of 5 to 20 Euro *cents*, than 2 to 3 Euros.

This study was carried out in the period November 1999 - February 2000. At the time of writing this chapter (November 2001), there are hardly any online shops known to us who sell *legal* music. An exception is *e-Music* (2001), who sell the right to listen to a music track a unlimited number of times. Customers pay a monthly subscription fee (Euro 9.-, assuming 1 Dollar = 1 Euro) and obtain the right to download a unlimited number of tracks. Compared to our price (20 Euro cents per track), a consumer should then download 45 tracks a month. Similarly, *Liquid Audio* (2001) sell the right to listen to track a unlimited number of times for prices in the range of Euro 1,- to Euro 2,-.

7.6.2 Consumer value perspective

Lesson 5: *Identifying multiple end-consumer segments is useful to reason about prototypical valuations of end-consumers.*
According to marketing theory and axiology, end-consumers each assign economic value to consumer experiences differently. However, for value modeling, this is not a very useful starting point. End-consumers generally come in thousands, and consequently it is not possible to model each consumer and his/her behavior. However, partitioning these consumers into a few market segments is useful. In the case of selling music, these segments allowed us to shade discussions. Using segments,

we were for instance able to say: 'in case of a yuppie actor it is likely the case that ...' rather than an often heard phrase in practice: '(all) consumers do ...'.

Lesson 6: *Holbrook's consumer value framework can be practically used to identify the valuable aspects of a product or service from the viewpoint of an end-consumer.*
Holbrook's theory on consumer value gives a framework for different types of consumer value. These types are useful in finding specific valuable aspects of a product, service or experience, which can be seen as instances of Holbrook's classes. In contrast to Holbrook's interpretive framework for qualitative marketing, we used these aspects in a meaningful way to quantify and reason about consumer value.

Lesson 7: *Evolutionary scenarios are a tool to enhance transparency of reasoning about valuation by end-consumers.*
Using evolutionary scenarios we can reason about the effects of future events or wrong assumptions. For each scenario, we assess consequences for profitability sheets for each actor or market segment involved. This makes reasoning more transparent. As an example, we thought that the effect of optimizing bandwidth between end-consumer and music store was a way to address piracy, but by assessing an evolutionary scenario it turned out that is only a solution on the short term.

Finally, we have used our consumer value-based argumentation on selling music also in discussions with the SENA (Stichting Exploitatie Naburige Rechten), one of the Dutch intellectual property rights societies. Initially, discussions focused on *protection* of digital assets, and the possibility of finding and prosecuting violators. Developing the consumer value perspective helped SENA and us in articulating factors which can be important if someone wants to sell music, and thereby exploiting the value in the (delivery of the) music itself to fight piracy.

Chapter 8

Value modeling: the profit perspective

This chapter presents the exploration of an e-commerce idea from an enterprise perspective. We do so by using an e-commerce exploration project we carried in the realm of the news paper industry. In short, the e-commerce idea was to offer a free online news article archive to subscribers on the newspaper (see section 8.1).

During the construction of a value model for this idea, it turned out that at least two principally different value models are possible. We present both models, and show that that our e^3-*value* ontology and UCM scenario mechanism can be used to pinpoint the essential differences between both models (section 8.2). Moreover, additional characteristics can be seen such as customer ownership and power elements of actors (e.g. price setting and supplier selection, see section 8.3).

In chapter 5, we have argued that for evaluation of value models we can take two perspectives: (1) a consumer value perspective; evaluation then assesses whether an e-commerce idea potentially generates sufficient value for an end-consumer, and (2) an enterprise perspective; we then focuses on potential profit generation for enterprises. In section 8.4 we exemplify evaluation of an e-commerce idea from an enterprise perspective.

The e-commerce at hand was explored, modeled, evaluated and put into operation some time ago. This allows us to retrospect on the models developed as well as on the evaluation carried out. These lessons are presented in section 8.5.

8.1 An e-commerce idea for online news articles

The e-commerce exploration project to be discussed in this chapter has already been introduced in section 6.2. To summarize, the e-commerce idea is to offer regular newspaper subscribers online news articles (in the form of an article archive). Additionally, the idea is to offer subscribers web services, such as surfing on the Internet, email and alike. In this chapter we focus on the idea to offer subscribers an online news article archive only.

From a *financial* perspective, the idea is to use a *termination* fee to finance the online article service. *Termination* means that if someone tries to set up a telephone connection by dialing a telephone number, another actor must pick up the phone, that is, *terminate* the connection. If someone is willing to *cause* termination of a large quantity of telephone calls, most telecommunication operators are willing to pay such an actor for that (the *termination fee*). Because the newspaper has a large subscriber base, s/he is capable of generating a large number of terminations for an *online article* service. We have seen this mechanism also in chapter 3.

8.2 Two alternative value models

During the construction of a value model for the aforementioned e-commerce idea, it turned out that that at least two different value models are possible: a *terminating* value model and an *originating* value model. Our experience during exploration of this idea was that many features and implications of these value models were not easy to discover during the project without the help of our model representations. Moreover, in this specific project our value models were used by stakeholders to explain to each other the consequences of choosing for a call termination or call origination model. As we will discuss in more detail, we are capable to represent the heart-beat of these value models with just a few pictures, which can easily be communicated to stakeholders and which have a clear meaning.

The following sections (sections 8.2.1 and 8.2.2) outline the differences between the models from a value ontology perspective. Section 8.3 analyzes differences more in depth. By doing so, we exemplify some other characteristics than only value creation, distribution and consumption, which also can be seen from a value model.

8.2.1 The terminating value model

The global actor viewpoint

Figure 8.1 shows the terminating value model. By following the scenario path, we see which actors have to exchange value objects in reaction to a start stimulus. Below, we follow the scenario path to introduce the terminating value model.

Readers. A start stimulus is caused by a reader if s/he wants to read an online article. Readers are subscribers on a newspaper, the *Amsterdam Times*, and come in thousands. Because of this, and for the assumption that readers value online articles equally, readers are grouped into a market segment. What makes this model special is that a reader has to exchange value objects with *two* actors to read an online article: (1) the *Amsterdam Times*, and (2) the *Last Mile*.

Amsterdam Times. The reader receives an article from the *Amsterdam Times*, and offers a *termination possibility* in return. The latter is key to this value model. By aggregating these possibilities, and because of his/her large subscriber base, *Amsterdam Times* has the potential to generate a large number of terminations.

Last Mile. The reader pays the local operator *Last Mile* a fee for a telephone connection. A local operator is a telecommunication operator who exploits the local loop: the last mile of copper or fiber between a telephone switch and a reader's house. By doing so, the local operator owns part of the infrastructure needed to offer a reader a telephone connection. This telephone connection is needed by the reader as a physical connection to access the online article archive using the Internet Protocol (IP). At the time this exploration track was carried out, only one local operator existed in the Netherlands, so only one such actor has been modeled.

Telecommunication consortium. As a result of the aforementioned exchanges both the *Amsterdam Times* and the *Last Mile* need to exchange value objects with a telecommunication consortium to deliver the online article experience to the reader, as can be seen by following the remaining part of the scenario path. These exchanges are about: (1) interconnecting traffic, (2) internet service provisioning, and (3) terminating traffic.

Interconnecting traffic. The *Last Mile*, as the name says, exploits only a part of the telephone infrastructure needed to offer the reader a telephone connection: the last mile between the reader's house and the nearest telephone switch. To make this telephone connection usable, it should be between the reader *and* a party exploiting IP access servers. These access servers offer IP connectivity and allow the reader, in conjunction with the underlying telephone connection as a physical carrier, to retrieve articles from server(s) hosting the article archive. The reader and these IP access servers can be located hundreds of miles away from each other. Now note that the *Last Mile* offers the reader a connection to an acccess server, but in reality only operates the last mile copper needed for such a connection. So, *Last Mile* needs to buy him/herself connectivity to bridge the remaining miles. In this case, another party, called a telecommunication consortium, offers this kind of interconnection. *Last Mile* pays the telecommunication consortium for doing so; this fee is called the interconnection fee. It is a fraction of the telephone connection fee paid by the reader.

Internet service provisioning. The core business of the *Amsterdam Times* is to produce news articles and newspapers. They are not so much interested in all technical activities, such IP access provisioning and content hosting, which are needed to make articles online available from a technical perspective. Therefore, they outsource these activities to the aforementioned telecommunication consortium.

Terminating traffic. For each scenario occurrence, the *Amsterdam Times* obtains a termination fee. This is paid by the telecommunication consortium, because the *Amsterdam Times* generates huge amounts of data traffic, thereby utilizing the infrastructure of the telecommunication consortium.

The detailed actor / value activity viewpoint

Figure 8.2 shows the detailed actor viewpoint, as well as the value activities performed by actors. We first discuss the elements of the telecommunication consortia and second review the value activities performed by actors.

The telecommunication consortium: a partnership. Figure 8.2 shows two partnerships, being two telecommunication consortia. *Amsterdam Times* can choose from one of these for service delivery. Each partnership has a number of actors which share a common value interface to their environment.

Consider the topmost partnership, consisting of the actors *Data Runner*, a telecommunication company and *Hoster*, an Internet service provider. Both these companies decide to offer telecommunication facilities for long distance traffic, hosting and IP access jointly as a bundle, under certain special conditions. A special condition can be the price, which might be cheaper for *Amsterdam Times* than an alter-

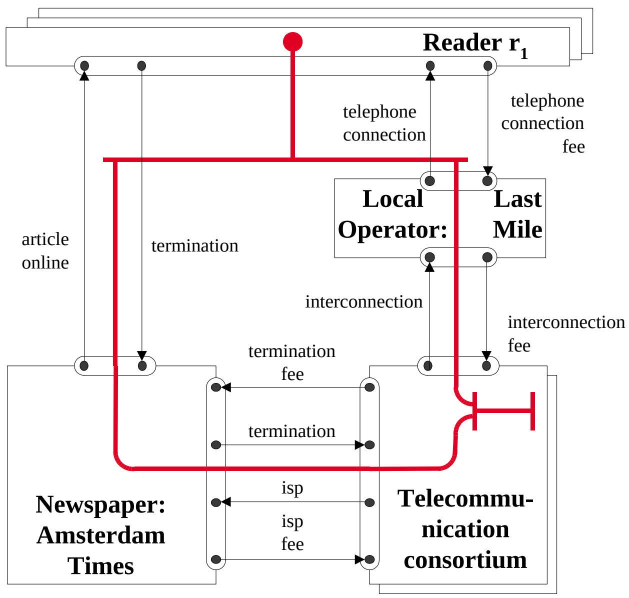


Figure 8.1: The *terminating* value model from a global perspective: the online article service offered by *Amsterdam Times* is funded by termination fees to be paid by the telecommunication consortium.

native, such as obtaining the objects of value from other actors separately. In this specific case, *Data Runner* and *Hoster* can offer services jointly cheaper, because they co-locate technical equipment such as a telephone switch, IP access servers, and web servers at one physical site, thus saving costly wide area connections to interconnect all these components.

The two partnerships shown are *equivalent* in a way that they offer comparable objects of value to their environment, but they are in no sense a market segment. They may value objects exchanged differently, and may use different pricing models. For instance, the topmost partnership may offer the same services (access, hosting) for lower prices than the other partnership. The other partnership is shown as a gray box, to prevent unnecessary cluttering of the diagram, but an additional detailed actor viewpoint for such a partnership can be modeled.

Value activities. Figure 8.2 also contains the value activities performed by actors. *Data Runner* performs long distance traffic and IP access provisioning, while his/her partner *Hoster* operates a web hosting facility. The *Amsterdam Times* has an operation of online news provisioning, which mainly boils down to (1) managing the telecommunication consortia from a service level perspective, and (2) using already written articles, obtained from the *publishing* activity to offer these online. This publishing activity is an *environmental* value activity already performed by the *Amsterdam Times*: it is needed to let the value model work (online articles must come from somewhere), but is not of further interest for this value model.

8.2.2 The originating value model

The global actor viewpoint

In contrast to the terminating model, the originating model assumes that the *Amsterdam Times* offers his/her online article service *directly* without any intermediate partners to his/her readers. From the reader's *perception* no other party than the *Amsterdam Times* is involved, while by using the terminating model the reader sees the *Last Mile* also.

Readers. To satisfy his/her needs, a reader obtains from the *Amsterdam Times* an online article, and in return pays the *Amsterdam* Times a fee for this. Note that this fee is *not* a telephone connection fee, but a fee for reading an article.

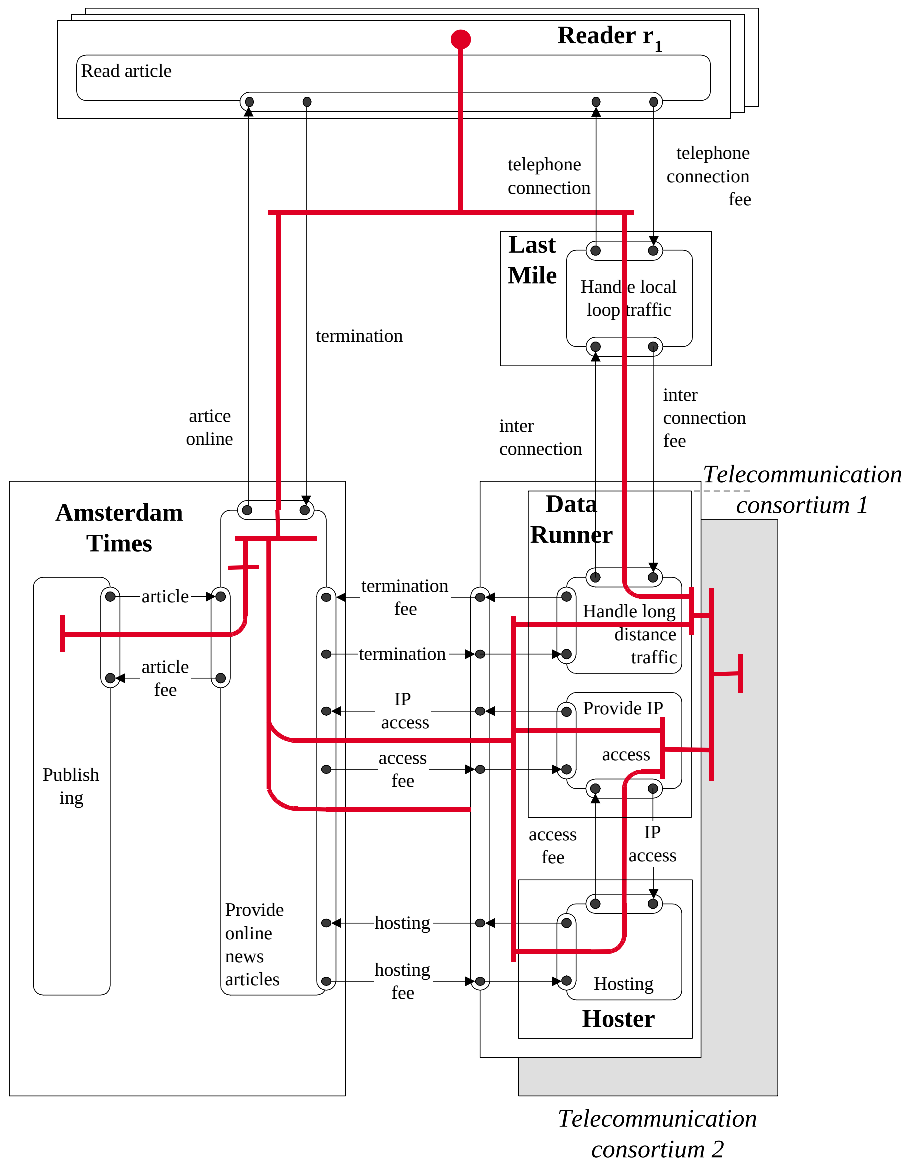


Figure 8.2: The *terminating* value model in a more detailed version: *Data Runner* and *Hoster* form a consortium to offer long distance traffic and internet service provisioning to *Amsterdam Times*. There is another consortium doing the same. *Amsterdam Times* selects, on a per scenario occurrence basis, which consortium handles the data traffic.

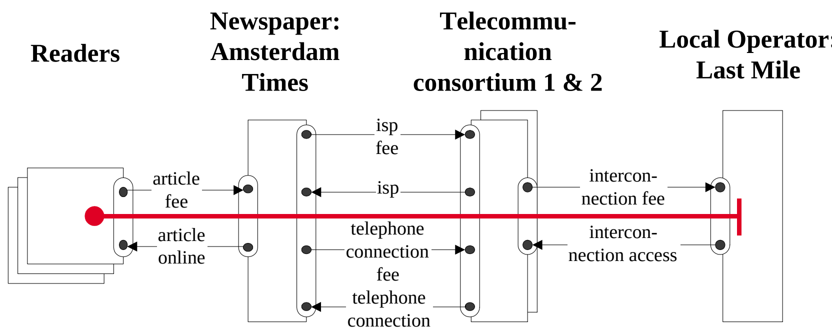


Figure 8.3: The *originating* value model from a global perspective. The reader pays the newspaper directly, who in turn pays the telecommunication consortium. This consortium pays the local operator a fee for handling the last mile traffic.

Amsterdam Times. As a consequence of value exchanges between the *Amsterdam Times* and readers, the *Amsterdam Times* needs to obtain ISP services *and* telephone connections. Note that the reader does not need to obtain a telephone connection anymore (and consequently does not see fees on his/her telephone bill for reading articles). This is the responsibility of the *Amsterdam Times*, who is offering an online article consisting of the article itself, but also the required telephone connection and IP access.

Telecommunication consortium. Activities, such as provisioning of a telephone connection and IP connectivity, as well as content hosting are outsourced to a telecommunication consortium, consisting of the same actors as was the case for the terminating value model. Only this consortium now gets a fee for services offered to the *Amsterdam Times*, which is a fraction of the fee received by the *Amsterdam Times* for providing online articles.

Last Mile. Finally, *Last Mile* receives a fee for handling the last mile of physical connection. This interconnection fee is a fraction of the telephone connection fee obtained by the telecommunication consortium. In short, the originating value model reverses the causality of revenue streams, compared to the terminating value model.

The detailed actor / value activity viewpoint

The detailed actor viewpoint and the value activity viewpoint are similar to the corresponding viewpoints for the terminating model (see figure 8.4). Differences between both models are in the value interfaces shown. The terminating model shows value interfaces for exchanging termination value objects and termination fees. These have been replaced in the originating model by value interfaces offering telephone connectivity and the corresponding fee. Also, the *Last Mile* now needs only one value interface rather than two.

8.3 Customer ownership and power

The previous section discussed the creation, distribution, and consumption of objects of value by actors for both the terminating and the originating value model. As we will show in this section, we can see other e-commerce idea's properties also by examining a value model: *customer ownership* and *power* relations between actors.

Customer ownership. In the situation that a customer buys a specific product type from only one seller regularly, such a seller starts to build up a relation with that customer, and 'owns' the customer with respect to that product type. Owning a customer is important, because it allows an actor to build a profile of a customer, which can used to offer the customer new products or services in the future. Whether an actor *solely* owns a customer can be seen by examining the value interfaces of the customer in conjunction with the connected value exchanges. If a value interface of a consumer is connected by value exchanges to only one other actor (a seller), the seller 'owns' the customer with respect to that value interface. However, if a customer's value interface is connected to more than one seller, customer ownership will be partitioned over these sellers.

Example: Customer ownership in the terminating model. Originally, the reader was a full customer of *Amsterdam Times*, because the reader is part of *Amsterdam Times*' regular subscriber database. However, for the online article service, as can be seen from figure 8.2, the reader now has to exchange values with *Amsterdam Times, and Last Mile*. Moreover, the latter is the party that receives the only payment for the delivered service. This can be seen as a shift in customer ownership from *Amsterdam Times* to *Last Mile*, which is an undesirable situation from *Amsterdam Times*' point of view.

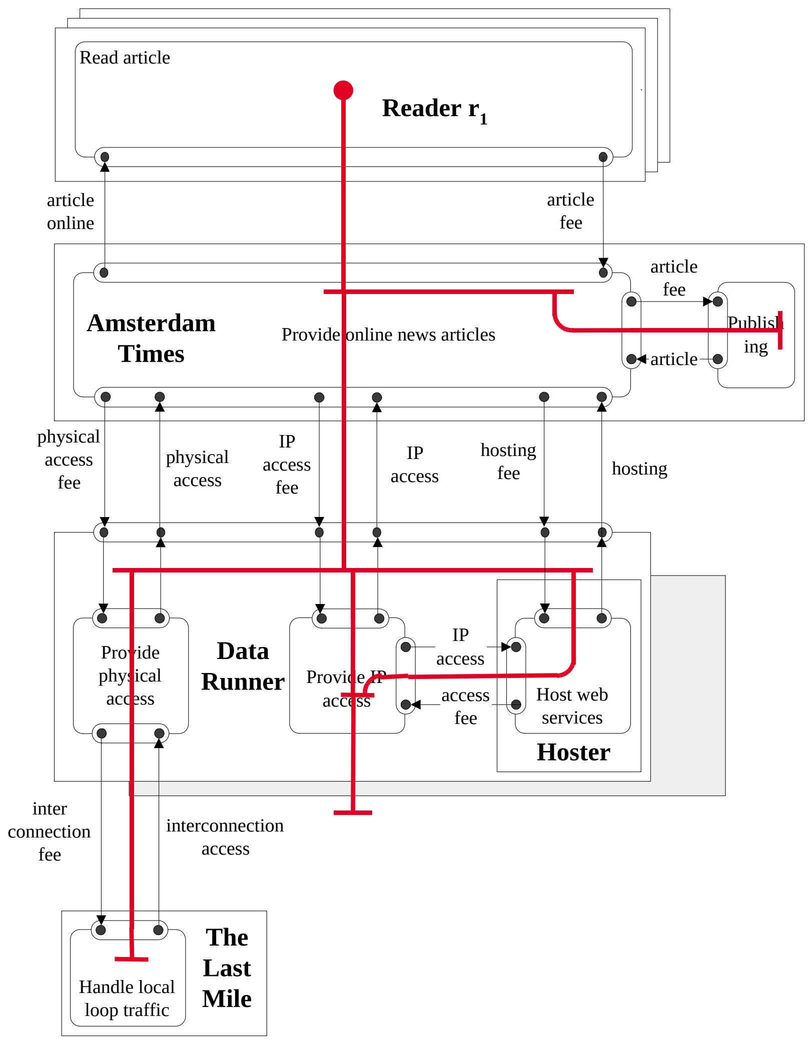


Figure 8.4: The *originating* value model in a more detailed version.

Example: Customer ownership in the originating model. In figure 8.4, the reader only 'sees' the *Amsterdam Times*, and not the *Last Mile* anymore for reading an online article. Also, the reader pays the *Amsterdam Times* directly for everything needed to read an online article. Because the reader pays to the same party that delivers the service, there is no shift of customer ownership.

Power: ability to set prices. An important aspect of business power is the ability to determine the price for products or services to be delivered. By examining value interfaces and value exchanges, at least actors playing a role in pricing can be seen. If we assign valuation functions of value objects representing money to specific actors who do these valuations, price setting can be seen in more detail.

Example: Price setting in the terminating model. In the terminating model, the reader pays for the *entire* telephone connection to the *Last Mile*, while *Last Mile* only operates a limited part of this connection. Unfortunately, no one, except *Last Mile* and perhaps a market regulation authority, can influence the pricing. Consequently, the success of the value model depends largely on *Last Mile*.

Example: Price setting in the originating model. In the originating model, the *Amsterdam Times* controls the price of the *online article* service his/herself. S/he can even decide for the, unlikely, case to *pay* the reader for reading articles online, a situation which is impossible for the terminating value model.

Power: ability to select a seller. If a buyer is able to make a selection of a larger set of potential sellers, his /her selection power increases, due to competition. This can be seen by the number of sellers a buyer is connected to.

Example: The reader can not choose his/her local loop provider in the terminating model. According to figure 8.2, the reader *must* use the *Last Mile* for local loop access. At the time the project was carried out, there was only one actor available controlling the local loop to subscribers. This can be concluded from figure 8.2, because only one actor for local loop access is drawn.

Example: Data Runner *can not choose his/her local loop provider in the originating model.* Similarly to the previous example, *Data Runner* can not choose an alternative local loop provider. This makes both value models very critical to the behavior of *Last Mile*.

Example: The Amsterdam Times *selects a telecommunication consortium per scenario occurrence.* This e-commerce idea has a special 'trick' to enlarge the power of *Amsterdam Times* with respect to the two telecommunication consortia. The *Amsterdam Times* can choose from these different consortia to actually offer the online article (from an access and hosting perspective), and this selection can be done on a per scenario occurrence base. The reason for this is that the *Amsterdam Times* does not want to be dependent on one telecommunication consortium for access and hosting. By distributing the amount of traffic over these two consortia, the *Amsterdam Times* controls the distribution of revenues for the two consortia, and motivates both to deliver a high level quality of service. This is graphically shown using an OR-fork in the scenario path, which models the supplier selection by *Amsterdam Times*.

In sum, a value model does not only represent value creation, distribution and consumption of value objects in a multi-actor network, but can also show other important business properties of the e-commerce idea at hand. For this specific case can be seen that for the terminating value model, customer ownerships shifts, while for the originating model customer ownership stays at *Amsterdam Times*. Also, from the value models can be concluded that for the terminating value model, *Last Mile* sets prices, while for the originating model the content owner, *Amsterdam Times* determines prices. The dominant role of *Last Mile* is emphasized by examining supplier selection; no one except *Last Mile* can deliver last mile connectivity to readers.

8.4 Evaluation of an e-commerce idea

The next step is to evaluate the economic feasibility of an e-commerce idea in quantitative terms, based on an assessment of the value of objects for all actors involved. As presented in section 5.6 evaluation of an e-commerce idea consists of a number of steps: (1) creation of a profitability sheet, (2) assignment of economic value to value objects, and (3) evaluation using evolutionary scenarios.

8.4.1 Creation of a profitability sheet for enterprise actors

While running the exploration track, it became clear that telecommunication actors were only able to offer a terminating value model. Therefore, we evaluate in this section only this value model.

Tables 8.1, 8.2, 8.3, and 8.4 show profitability sheets for the scenario *read online article*, for the *Last Mile*, the *Amsterdam Times*, and the telecommunication consortium 1 respectively. We suppose that the profitability sheet for telecommunication consortium 2 is the same as for telecommunication consortium 1. As the value model in figure 8.2 contains two different scenario paths, the sheets show two scenario paths also. These paths are named *Telecommunication consortium 1* and *2*, representing the different paths as a result of the OR-fork superimposed on the *Amsterdam Times* to denote the before discussed supplier selection.

We have created these sheets by following the scenario paths, starting at the start stimulus, and each time the path crosses a value interface of an actor, the sheet is updated with value objects flowing in and out of that actor.

8.4.2 Assignment of economic value to value objects

A second step in evaluation is the assignment of economic value to value objects. Since we only consider the enterprise perspective in this chapter, assignment takes two steps: (1) determination of valuation functions for value objects representing money, and (2) reduction of non-money value objects (see section 5.6.2 for a detailed discussion).

Determination of valuation functions for value objects representing money

For enterprises, we only consider cash in and -out flows. Consequently, below we only give valuations for value objects representing money.

Telephone connection fee. The telephone connection fee per scenario occurrence is based on a start tariff and a connection-time dependent tariff. To calculate the total monthly fees, the telephone connection fee is multiplied with the realized number of scenario occurrences.

Interconnection fee. The interconnection fee per scenario occurrence (here only shown for actors in *telecommunication consortium 1*) is based on a fraction (the interconnection factor, a number between 0 and 1) of the telephone connection fee, and on a percentage of the physical distance *Data Runner* bridges.

Termination fee. The termination fee *Amsterdam Times* receives, in this case from *telecommunication consortium 1*, is calculated analogously to the interconnection fee, only now we use a revenue sharing factor rather than an interconnection factor. Typically, the revenue sharing factor is smaller than the interconnection factor times the percentage of the physical distance bridged by an operator. Note that by valuing this way, we are capable of analyzing the effects of a decreasing interconnection factor (e.g. influenced by a market regulator), while the revenue

Table 8.1: Profitability sheet for the *Last Mile* actor per scenario occurrence

Actor	Last Mile	
Scenario	Read online article	
	Value Object In	*Value Object Out*
Scenario path	Telecommunication consortium 1	
Likelihood	50%	
Exchanges with readers:	$\textit{telephone connection fee} = \textit{start tariff} + \textit{connection tariff} \times \textit{duration}$	$(\textit{telephone connection})$
Exchanges with telecommunication consortium 1:	$(\textit{interconn.}_{\textit{telco cons. 1}})$	$\textit{interconn. fee}_{\textit{telco cons. 1}} = \textit{telephone connection fee} \times \textit{interconn. factor}_{\textit{telco cons. 1}} \times \textit{distance factor}_{\textit{telco cons. 1}}$
Scenario path	Telecommunication consortium 2	
Likelihood	50%	
Exchanges with readers:	$\textit{telephone connection fee} = \textit{start tariff} + \textit{connection tariff} \times \textit{duration}$	$(\textit{telephone connection})$
Exchanges with telecommunication consortium 2:	$(\textit{interconn.}_{\textit{telco cons. 2}})$	$\textit{interconn. fee}_{\textit{telco cons. 2}} = \textit{telephone connection fee} \times \textit{interconn. factor}_{\textit{telco cons. 2}} \times \textit{distance factor}_{\textit{telco cons. 2}}$

Table 8.2: Profitability sheet for the *Amsterdam Times* actor per scenario occurrence

Actor	Amsterdam Times	
Scenario	Read online article	
	Value Object In	*Value Object Out*
Scenario path	Telecommunication consortium 1	
Likelihood	50%	
Exchanges with readers:	$(termination)$	$(online\ article)$
Exchanges with telecommunication consortium 1:	$term.fee_{telco\ cons.\ 1} =$ *see Data Runner*	$(termination_{telco\ cons.\ 1})$
	$(IP\ access_{telco\ cons.\ 1})$	$IP\ access\ fee_{telco\ cons.\ 1} =$ *see Data Runner*
	$(hosting_{telco\ cons.\ 1})$	$hosting\ fee_{telco\ cons.\ 1} =$ *see Hoster*
Scenario path	Telecommunication consortium 2: see telecommunication consortium 1 path	
Likelihood	50%	
Exchanges with readers:	$(termination)$	$(online\ article)$
Exchanges with telecommunication consortium 2:	...	...

Table 8.3: Profitability sheet for the telecommunication consortium actor/Data Runner per scenario occurrence

Composite actor	Telecommunication consortium 1	
Actor	Data Runner	
Scenario	Read online article	
	Value Object In	*Value Object Out*
Scenario path	Telecommunication consortium 1	
Likelihood	50%	
Exchanges with *Amsterdam Times*:	$(termination)$	*termination fee* $=$ *telephone connection fee* $\times$ *revenue sharing factor*
	IP access $fee_{Amsterdam\ Times} =$ *IP access fee* $\times$ *duration* $\times$ *AT-forecast-formula*	$(IP\ access_{Amsterdam\ Times})$
Exchanges with *Hoster*:	*IP access* $fee_{Hoster} =$ *Hoster-forecast-formula*	$(IP\ access_{Hoster})$
Exchanges with *Last Mile*:	*interconnection fee* $=$ *see Last Mile*	$(interconnection)$

Table 8.4: Profitability sheet for the telecommunication consortium actor/Hoster per scenario occurrence

Composite actor	Telecommunication consortium 1	
Actor	Hoster	
Scenario	Read online article	
Scenario path	Telecommunication consortium 1	
Likelihood	50%	
Exchanges with *Amsterdam Times*:	*hosting fee* = *Hosting-forecast-formula*	(*hosting*)
Exchanges with *Data Runner*	(*IP access*)	*access fee* = *see Data Runner*

sharing factor remains the same. This models a situation where *Data Runner* takes the risk of a decreasing interconnection factor.

IP access fee - *Amsterdam Times*. *Data Runner* charges *Amsterdam Times* an IP access fee in return for giving readers access. This fee is based on an IP access tariff per second. We want to account for the situation that IP access equipment is a very scarce resource; *Data Runner* wants to have the opportunity to assign unused IP access ports to others. Therefore, *Amsterdam Times* is asked to forecast the number of scenario occurrences on a monthly basis, including the average duration. *Data Runner* then allocates access ports on this forecast, and can allocate remaining ports to others. To motivate *Amsterdam Times* to do good forecasting, the following valuation is used: If the number of realized scenario occurrences drops below an inaccuracy factor (e.g. 75 %) of the forecast occurrences, we use 75 % of the forecast occurrences for the calculation of the monthly IP access fee. Otherwise, we use the realized number of scenario occurrences (see also formula 8.1).

IP access fee - *Hoster*. The IP access fee to be paid by *Hoster* is based on the forecast number of maximum concurrent scenario occurrences. These occurrences

```
AT-forecast-formula(realized-occurrences, forecast-occurrences, inaccuracy-factor)
{
if realized-occurrences < forecast-occurrences × inaccuracy-factor then
    return (forecast-occurrences × inaccurancy-factor)/realized-occurrences;
else
    return 1;
endif
}
```

Formula 8.1: Forecast formula for the use of IP access by the *Amsterdam Times*.

require IP connectivity between *Hoster* and *Data Runner* with a predetermined bandwidth to be in place, which is adjusted on a monthly basis, using the forecast. Based on the required bandwidth, we calculate a fee for IP access.

Hosting fee. The hosting fee is calculated in a similar way as the IP access fee for *Hoster*. *Hoster* uses a forecast of *Amsterdam Times* of the number of concurrent page views, which in turn is based on an average number of page views per forecast scenario occurrence. This results in a fixed fee per month for hosting.

Reduction of non-money value objects

All objects which do not represent money objects are removed from the profitability sheets. While doing so, we check if each non-money object, which enters an actor, also leaves the same actor. We assume that each non-money value object that flows into an actor, also flows out such an actor (see also section 5.6.2, guideline 5.5).

Reduced value objects are in tables 8.1, 8.2, 8.3, and 8.4 shown between parentheses. For example, we remove telephone connection and interconnection from the actor *Last Mile*, because the telephone connection is an enriched interconnection. *Last Mile* enriches the *interconnection* by exploiting a district telephone switch and a last mile of copper or fiber optics. Note that the IP access value object in the profitability sheet of *Data Runner* cannot be reduced. Therefore, the IP access fee must be sufficient to finance the exploitation of IP access servers.

Table 8.5: Basic assumptions used to evaluate the terminating value model.

Property	*Value*	*Property*	*Value*
scenario occurrences per month:	1,500,000	supplier selection ratio:	0.5
concurrent scen. occ.:	10,000	inter-connection factor:	1
concurrent page views:	10,000	revenue sharing factor:	0.55
average scenario duration:	480 s	distance factor:	0.8
prices for bandwidth & hosting:	a ladder	valuation composite 1:	equal to composite 2
bandwidth per user:	1024 bps	forecast:	realized
start tariff:	Euro 0.05	conn. tariff:	Euro 0.01 per minute
IP access fee:	0.0015 Euro per minute		

8.4.3 Evolutionary scenarios

Using the valuation in tables 8.1, 8.2, 8.3, and 8.4 we evaluate several evolutionary scenarios, which model expected changes in the future regarding valuation. Important assumptions are shown in table 8.5 [1]. As an example, table 8.6 shows the consequences of the occurrence of identified evolutionary scenarios for profitability sheets.

Scenario 1: Null scenario. The *null* scenario refers to a situation for which we use the numbers in table 8.5 for calculation of profits. Observe that *all* actors make a profit.

[1] These assumptions are for a hypothetical case, to respect confidential project data.

Table 8.6: Different valuation scenarios. The null-scenario uses the valuation in table 8.5. A second scenario assumes that *Amsterdam Times* forecasts inaccurately. A decrease in the interconnection is expected to occur, especially of competition between telecommunication actors increases (see the third case). The fourth scenario supposes a drop in the revenue sharing factor between *Data Runner* and *Amsterdam Times*.

		Profit (Euro)			
Scenarios		*Amsterdam Times*	*Last Mile*	*Data Runner*	*Hoster*
1.	*Null-scenario, Forecast = Realized*	55,800	39,000	46,100	4,000
2.	*Forecast (1,500,000) >> Realized (150,000)*	-16,920	3,900	12,260	4,000
3.	*Decrease in interconn. factor (1.0 to 0.4)*	55,800	132,600	-700	4,000
4.	*Decrease in revenue sharing factor (0.55 to 0.1)*	-14,400	39,000	81,200	4,000

Scenario 2: *Amsterdam Times* is a bad forecaster. What happens if the *Amsterdam Times* is not a good forecaster of scenario occurrences [2]. It can be seen that *Amsterdam Times* will not make a profit. For *Last Mile* and *Data Runner* there is still a profit to cover the costs. *Hoster* is insensitive to bad forecasts, because it does not depend on the number of realized scenario occurrences.

Scenario 3: Interconnection factor decreases. It is reasonable to expect a decrease in the interconnection factor after some months, because presently this factor is high to stimulate competition between telecommunication operators. As soon as this competition works, this factor will decrease. *Amsterdam Times* does not feel such a decrease, but *Data Runner* will.

Scenario 4: Revenue factor decreases. *Data Runner* may decide to decrease his/her revenue sharing factor. As can be seen, this will harm *Amsterdam Times*.

In conclusion, by valuing the objects for each actor, and by making reasonable assumptions about the number of (forecast) scenario occurrences, we can perform a sensitivity analysis for the business idea hand. This sensitivity analysis is in many cases of more business interest than the numbers of the valuation itself.

8.5 Lessons learned

E-commerce idea exploration, as well as its implementation for the project discussed in this chapter took place during December 1999 - February 2000. The project was carried out for a publisher called *PCM*, a publisher of daily newspapers in the Netherlands. The driving actor was PCM Interactive Media (PIM), a subsidiary of PCM. In September 2001, PCM publicly announced to stop most of its Internet related activities, of which the service outlined in this chapter is part of (*PCM Bezuinigt op Internet* 2001). It is likely that the online article service explored in this chapter will be phased out the coming years. Because of this, we revisited PCM in November 2001. The goal of this visit was first to understand PCM's decision, but more importantly to assess whether we reasonably could have foreseen a failure during exploration of the online article e-commerce idea. If so, we can learn from it and improve our e^3-*value* approach.

[2]For scenarios 2, 3, and 4, we assume that both telecommunication consortia are equally effected. So, for scenario 2, *Amsterdam Times* has to pay both consortia a fee for bad forecasting, for scenario 3, a decrease in the interconnection factor harms both telecommunication consortia, and for scenario 4, a decrease in the revenue factor will benefit both consortia.

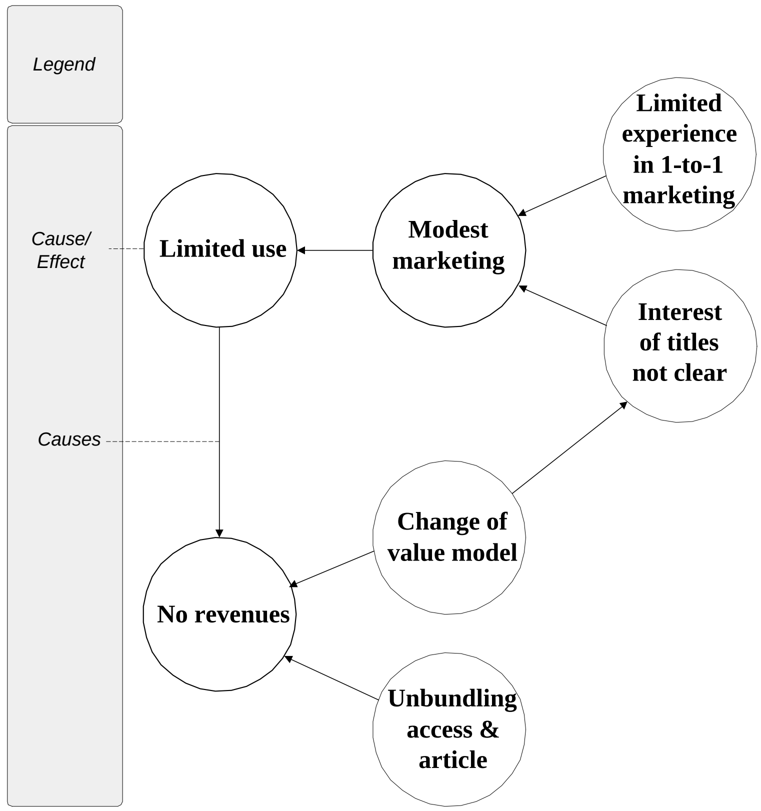


Figure 8.5: Cause effect graph showing why the online article service is likely to be terminated.

PCM has a number of reasons for stopping the *online article* service, but some of these are not directly related to this service. Figure 8.5 presents causes and effects which motivated PCM to stop the service at hand, and which are directly related to the online article e-commerce idea outlined in this chapter. The following sections discuss these causes and effects.

8.5.1 Limited use of the service

The use of the online article service is modest. The number of scenario occurrences per timeframe is not as many as hoped for. Disappointing numbers on scenario occurrences was one of the evolutionary scenarios we have identified (scenario 2). Figure 8.5 shows an explanation for limited use of the online article service: modest marketing.

Modest marketing

A cause for a limited use of the service is the *modest marketing*. For instance, it is not very easy for potential readers to find the service or to subscribe themselves on the service. Also, not many efforts have been done to attract regular subscribers on a newspaper to the online alternative.

Lesson 1: *Development of a marketing perspective is needed during e-commerce idea exploration.*
The way a service is marketed is currently not part of the e^3-*value* approach. In contrast, Timmers (1999) distinguishes explicitly a marketing perspective in addition to the notion of *business model*. We learned from this study that the way an e-commerce idea (once it can be articulated) is marketed is important for its acceptance and success. Therefore, we identify in section 11.3 the development of an explicit marketing perspective as future research.

We have identified two causes for modest marketing (see the coming sections): (1) the interest of brand owners in the service is not clear, and (2) there is only limited experience with one-to-one marketing, compared to the mass-marketing a newspaper is used to.

The interest of brand owners in the service is not clear

PCM is publisher of a number of newspapers called titles. Titles have specific *brand owners* (e.g. *De Volkrant*, *Algemeen Dagblad*, *NRC*, and *Trouw*). These brand owners were however not explicitly modeled in our value models (see figures 8.2 and 8.4). We only have identified an actor called *Amsterdam Times*, denoting PCM and all her brand owners. This actor publishes regular newspaper articles (one of the activities done by the brand owners), and offers online articles (a joint activity of brand owners and PIM). Not distinguishing PCM's internal structure has the following drawbacks:

- *commercial (selling) responsibility for the online article service is unclear*: the value model does not show in detail who is responsible for value exchanges (e.g. the online articles) between readers and PCM;

- *interests of PCM's business units (brand owners and PIM) in the e-commerce idea is unclear*: the model does not show how brand owners and PIM as independent profit centers earn money with the online article service.

To address the mentioned drawbacks, consider figure 8.6, which illustrates a possible value model for PIM and the PCM brand owners. Brand owners now are responsible for offering online articles to their subscribers. To stimulate selling, brand owners receive a modest fee (a fraction of the termination fee PIM receives from a telecommunication consortium), which directly relates to the use of the online article service.

Lesson 2: *If multiple business units of one enterprise participate in an e-commerce idea, model explicitly which units, rather than the enterprise as a whole, exchange value objects with their external customers.*
Profit and loss responsible actors (such as brand owners) being part of a conglomerate (e.g. PCM) should be modeled explicitly, as well as their interaction with customers outside the conglomerate. If such actors commit themselves to a value model, arguments on who is responsible for marketing and selling value objects (such as online articles) afterwards can then be avoided.

Lesson 3: *If multiple business units of one enterprise participate in an e-commerce idea, model explicitly these units as actors, and the objects of value they exchange.*
In addition to the aforementioned lesson, a value model should also illustrate how actors of a larger conglemerate account each other for a specific e-commerce idea. For each actor it should be clear how s/he creates value for other actors part of the conglomerate (or for external customers).

Limited experience in one-to-one marketing

PCM and its brand owners have only limited experience with one-to-one marketing. Such marketing is needed to sell the new proposition to each individual subscriber on a title. In contrast, brand owners are very experienced in mass-marketing.

As discussed before, the e^3-*value* methodology should be extended with a marketing perspective, also to address a shortcoming in marketing plan implementation capabilities.

8.5.2 No revenues

After a certain time of execution, an e-commerce idea should contribute to profit for the participating enterprises. This is not the case for the service at hand. One of

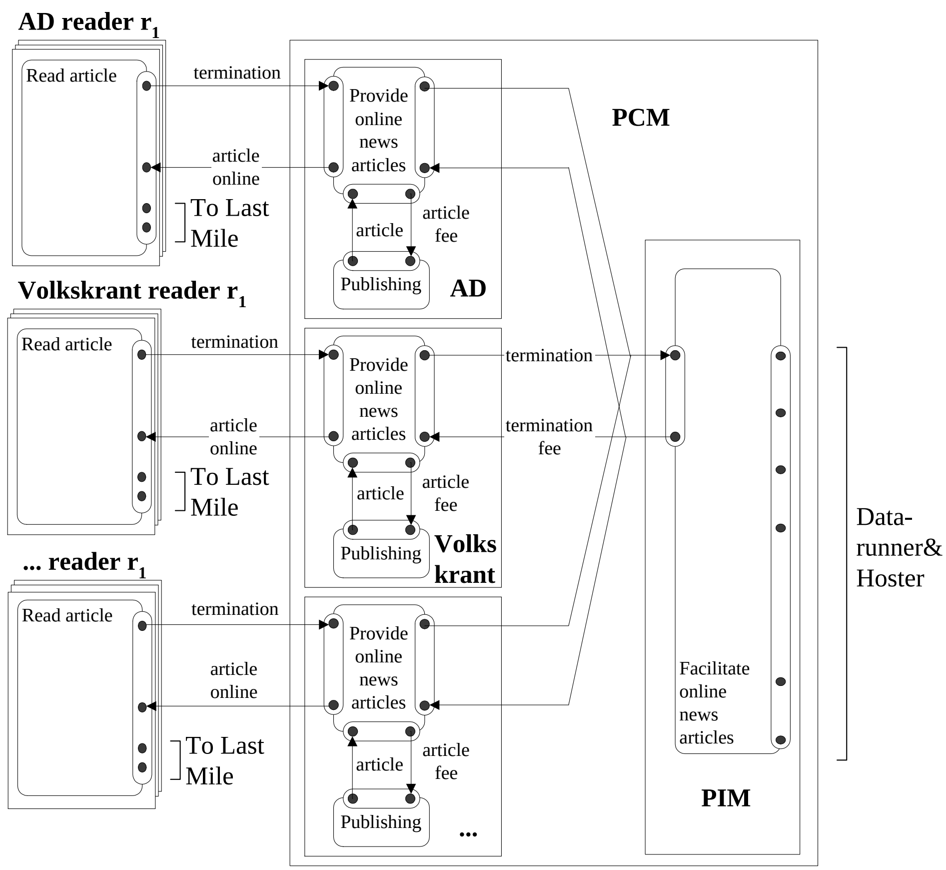


Figure 8.6: A fraction of a value model emphasizing that (1) individual brand owners have the end-consumer relations and are responsible for ‘selling’ the online articles, and (2) brand owners receive fees based on the amount of traffic they generate with their content.

the causes for this is a modest use of the service, but two other reasons have been identified: (1) a change in the proposed value model, and (2) unbundling of articles online and IP access.

8.5.3 Change of value model

At the time we left the project, the consortium decided to choose the terminating value model. A main reason for doing so was that, at the time of implementation, it was not possible to roll-out the originating value model for technical reasons. However, after we left the project, contract negotiations between PCM and the telecommunication consortia continued. They felt that the designed value model was too complex, and so they decided to choose for a model presented by figure 8.7. The difference with the original model (see figure 8.2) is that PCM pays a very modest fee to the telecommunication consortium for hosting and access. Moreover, this consortium must finance his/her operations related to the e-commerce idea by termination fees. So, in the new model PCM is not paid, but rather must pay a modest fee itsself.

Such a new value model only works if there are revenues for PCM from other sources, e.g. from subscribers, or an increase in customer loyalty/branding, which can be translated into revenues. However, it was decided not to choose for such a solution as can be seen from figure 8.7: fees are only *paid* by PCM and not *received*. It also not clear how the business units (brand owners and PIM) themselves are funded for this service. This is one of the main reasons why the online article service can not survive.

8.5.4 Unbundling access and online articles

The original value model (see figure 8.2) assumes that the *only* way to access an online article is to set up a telephone connection with a *selected* telecommunication actor. With such a telecommunication actor, PCM has an agreement on termination fees. In other words, access is bundled with online articles. This can be concluded from the actors shown, their value interfaces and exchanges, as well as the way scenario paths are drawn. Bundling of access and articles ensures that an interconnection fee and termination fee is paid to the telecommunication consortium and PCM.

Some brand owners have chosen not to bundle access and the online article (see figure 8.8). Readers of a specific label can choose an Internet Service Provider (ISP) themselves to access the online articles. To do so, the online article archive

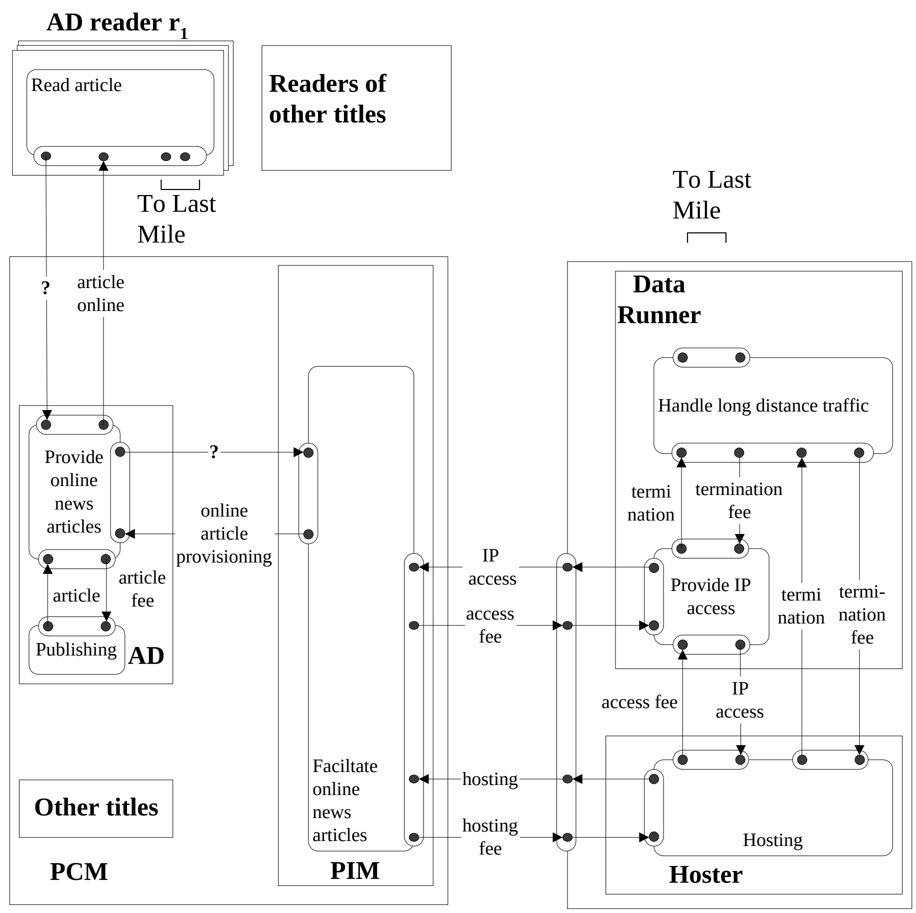


Figure 8.7: A fraction of the value model implemented, which differs from the proposed model (see figure 8.2): Data Runner and Hoster must finance their operation entirely with termination fees, plus with a small fee received from PCM/PIM. PCM has no additional funding.

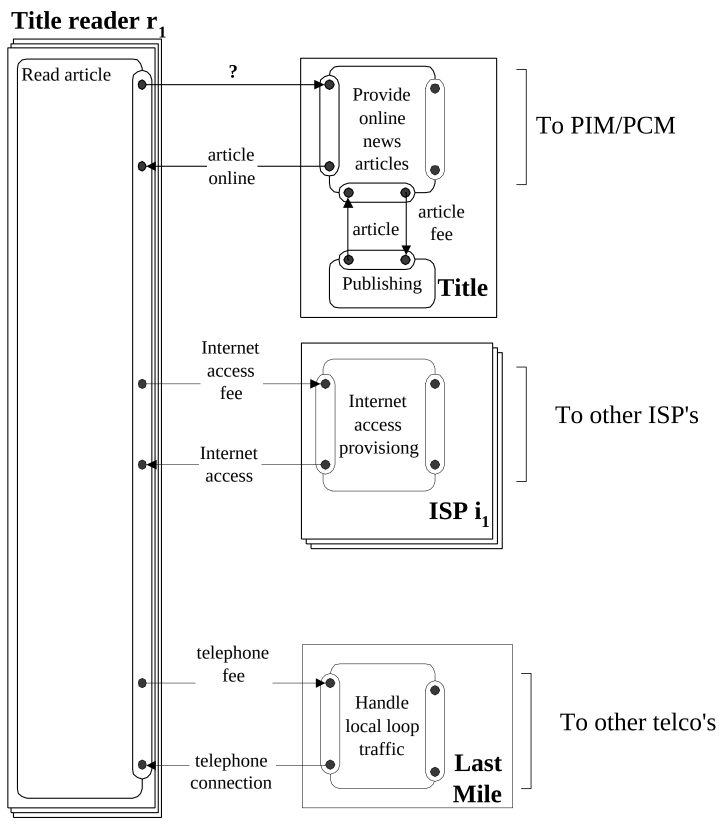


Figure 8.8: A fraction of a value model implemented by one of the brand owners: debundling of access and online articles.

is connected to the Internet. As a result, no interconnection fee is paid to the telecommunication consortium the e-commerce idea was designed for, and consequently PCM does not receive a termination fee. This disrupts the designed value model presented in figure 8.2, but also shakes up the implemented value model in figure 8.7. In the latter case, the telecommunication consortium does not receive fees anymore to finance his/her hosting service offered. As a result, this actor may charge an additional fee for hosting, e.g. to the title responsible for unbundling. It is questionable (denoted by the question mark in figure 8.8), how the reader is charged for this service. The consequence of unbundling is that the online article service must be financed by sources elsewhere (e.g. by the reader), but is not clear how this happens.

Lesson 4: *An e-commerce idea continuously evolves over its lifetime. A value model should therefore be maintained and evaluated for each major change.*
As shown, an idea evolves during its lifetime (e.g. during design and execution). Ideally, a value model should capture these changes. Moreover, each major change should be evaluated for profitability consequences. For the specific case at hand, the consequences of removing the termination fee value exchange between the telecommunication consortium and PCM, as well as unbundling the online article and access, can be shown using our e^3-*value* modeling technique. Also, figure 8.7 illustrates that PCM has no income after changing the value model, and by constructing new scenario paths and profitability sheets, it can seen from figure 8.8 that the telecommunication actor misses revenues as a result of unbundling

Lesson 5: *Find evolutionary scenarios by using various kinds of scenarios: (1) scenarios caused by changes in valuation, (2) scenarios caused by a change in the number of scenario occurrences, and (3) scenarios caused by a change in a value model's structure.*
During idea exploration, we have only focused on evolutionary scenarios, which capture changes in valuation by actors, e.g. as a consequence of market deregulation, and scenarios presenting wrong assumptions on the use of the service (wrong numbers on scenario occurrences). By revisiting PCM, we have learned that evolutionary scenarios can also be classified as changes in the structure of the value model itself (e.g. removing value exchanges and ports, and debundling). See also lessons 5.9-5.11 as discussed in section 5.6.3.

Chapter 9

The e-commerce model: viewpoints + scenarios

In sections 2.3.3 and 2.4, we have argued that multiple viewpoints are important for the exploration of an e-commerce idea. One of the reason for doing so is to have the right discussions with the appropriate stakeholder group. For instance, not all stakeholders have a say in business value decisions. Also, CxO's should hardly be involved in information system related issues.

A potential drawback of using these relatively self-contained viewpoints is that they each may diverge from the original e-commerce idea, because each stakeholder group responsible for the exploration of a particular viewpoint takes his/her own decisions without consulting other stakeholder groups to much. In short, we need a mechanism that helps stakeholders to explore *different* viewpoints while at same time helps to keep focused on the *same* e-commerce idea.

In this chapter, we present an approach to achieve this. We propose our previously introduced operational scenario method (based on use case maps (UCMs), see section 3.4.1) to relate and integrate different stakeholder viewpoints. For each viewpoint we develop the *same* set of operational scenarios, expressed by *different* use case maps tied to that particular viewpoint. By developing the same operational scenarios for each viewpoint, different requirement viewpoint models emerge as a single integrated set of requirements. By doing so, we see an e-commerce model as a model consisting of specifications on our three identified viewpoints, plus integrating operational scenarios.

There is also another use of the aforementioned operational scenarios when exploring multiple viewpoints. In section 5.1 we have stated that we use information on revenues and expenses from all viewpoints to create a profitability sheet for each actor. We use operational scenarios as a way to relate revenue and expense numbers on each viewpoint and thus to create profitability sheets for the overall e-commerce idea rather than for a specific viewpoint. The major objective of these sheets is to justify stakeholder confidence in the commercial feasibility of an e-commerce idea, and not so much to obtain precise estimates of expected benefits. In fact, in the early requirements stages of innovative e-commerce projects, the former is much more important (and realistic) than the latter.

The remaining part of this chapter is structured as follows. Section 9.1 defines the concept of *e-commerce model* from a viewpoint and integrating scenario perspective. Section 9.2 presents in an informal way an e-commerce idea for Internet contact ads, which we use to show how integrating scenarios work, and how we construct profitability sheets for this e-commerce idea. Section 9.3 shows for this idea the operational scenarios. The idea is conceptualized in section 9.4 (value viewpoint), section 9.5 (business process viewpoint), and section 9.6 (two alternative information system viewpoints). Section 9.7 illustrates how we use operational scenarios to integrate profitability sheets of various viewpoints, and section 9.8 presents our conclusions.

9.1 The e-commerce model

9.1.1 The e^3-*value* viewpoints revisited

In section 2.4 we have introduced three viewpoints which are important for innovative e-commerce idea exploration: (1) the value viewpoint, (2) the business process viewpoint, and (3) the information system viewpoint. In this section, we briefly review these viewpoints.

The value viewpoint. The value viewpoint is the focus of this thesis (see chapter 3 for an extensive discussion on this viewpoint as well as concepts used to express it). The value viewpoint shows how a multi-actor network creates, distributes and consumes economic value. This viewpoint contributes to insight in revenues *and* expenses, caused by the exchange of valuable objects between actors.

The business process viewpoint. The business process viewpoint shows how an e-commerce idea described by a value viewpoint can be put into operation. It does so by outlining operational activities, their sequence of performance, the in-and outputs for these activities, and their operating actors. The business process viewpoint should provide understanding of how an e-commerce is carried out, and should after a first exploration cycle provide a starting point for identification of large operational expenses, which are necessary to put the e-commerce idea into operation. These can influence the economic feasibility of an e-commerce idea.

The information system viewpoint. The information system viewpoint outlines constituting components of an information system to be developed at a course granularity. From an exploration point of view, this viewpoint should reveal expensive system components (expensive relative to other expenses, e.g. for performing business processes), both from an operational expense perspective and a capital expense perspective.

9.1.2 The e-commerce model: viewpoints plus integrating operational scenarios

The before mentioned viewpoints are based on similar foci of stakeholders who play a role during e-commerce idea exploration. A main reason for using these separate viewpoints is to have the right discussions with the right stakeholder group. In section 2.4 we mention assumptions and criteria we have used to identify these viewpoints. One of these criteria is minimum overlap: viewpoints should be relatively self-contained to allow stakeholders to make decisions without consulting stakeholders focusing on other viewpoints too much. A potential danger of using these self-contained viewpoints is that they become unrelated. Because groups of stakeholders decide relatively independent on requirements expressed by the viewpoints they focus on, they can easily overlook consequences of choices made by others.

To address the potential danger of unrelated viewpoints, we use operational scenarios for all the three viewpoints. These scenarios are from a conceptual perspective the *same* for each viewpoint, and are expressed by use case maps (see section 3.4.1). These use case maps, which put a scenario into operation, *differ* for each viewpoint. As a result, viewpoints are related with each other by operational scenarios which in turn are grounded in customer needs. By doing so, we see scenarios (as well as their operationalization in viewpoint specific use case maps) and the three viewpoints related by these scenarios as a specification of an *e-commerce model*.

The idea of using operational scenarios for relating viewpoints is borrowed from Kruchten (1995). He introduces operational scenarios, expressed by scripts, to relate four software architectural viewpoints and to exemplify how these viewpoints work together in describing a software architecture.

Identifying the e-commerce model with a set of operational scenarios gluing together viewpoints also serves another goal. It allows us to create profitability sheets for an e-commerce idea as a whole, rather than to have profitability sheets only on the level of viewpoints This chapter shows, using an e-commerce exploration project, how models on various viewpoints can be related using operational scenarios, *and* how we use these scenarios to construct profitability sheets based on all viewpoints, thus resulting in an overall e-commerce model.

Other opinions on e-commerce models. In the realm of e-commerce, one often refers to the concept of e-business model. Rather than using *e-business model*, we prefer to talk about the *e-commerce model* (see also section 2.1.1 on the difference between e-commerce and e-business). In this section, however, we use *e-business model* and *e-commerce model* interchangeably.

Many definitions of *e-commerce model* recognize our value- and sometimes our business process viewpoint to a certain extent, but fail to see an information system as part of an e-commerce model. For instance, others define the concept *e-commerce/business model* as follows:

Timmers (1999): *An architecture for product, service, and information flows, including a description of the various business actors and their roles; and a description of the potential benefits for the various actors; and a description of the sources and revenues.*

Slywotzky (1996): *The totality of how a company selects his/her customers, defines and differentiates his/her offerings (or responses), defines the tasks it will perform his/herself and those it will outsource, configures his/her resources, goes to the markets, creates utility for customers and captures profits.*

Rappa (2000): *In the most basic sense, a business model is the method of doing business by which a company can sustain his/herself - that is, generate revenue. It spells-out how a company makes money by specifying where it is positioned in the value chain.*

Compared to these definitions, we explicitly take the information system perspective into account with respect to an e-commerce model, because in our experience information systems are key to success for most e-commerce tracks. Moreover these approaches do not identify explicitly how to relate requirements expressed on different viewpoints.

9.2 An e-commerce idea for digital contact ads

We illustrate operational scenarios and how they serve as an integration glue between viewpoints using an e-commerce exploration track we have carried out. The project is about an Internet enabled contact ad service, which we also used in chapter 4 to outline the differences between value models and process models.

The *Ad Association* is a company that coordinates about 200 local, world-wide located, free ad papers called FAPs. FAPs independently produce (non-electronic) papers with ads and serve a geographical region. The handling of ads is as follows. A customer submits an ad to a FAP. The FAP checks the ad (e.g. for absence of dirty language and for style) and places the ad in his/her next issue. It is possible to place an interregional ad. In this case, the FAP to which the ad was submitted distributes the ad to the *Ad Association*, who redistributes the ad to other FAPs (serving different geographical regions). These other papers publish the ad as soon as possible. In a new e-commerce idea, the *Ad Association* and FAPs want to exploit their local established brand names to develop an internationally, Internet-based, contact ad service.

The following sections show for this e-commerce idea the first iteration in exploring the aforementioned viewpoints to build confidence in commercial and technical feasibility. We construct one value model and a corresponding process model. Subsequently, we discuss two information system variants that both realize the given value and process model.

9.3 Operational scenarios

A first step after a statement of an e-commerce idea is to outline the value-added services to be offered in terms of customer grounded operational scenarios. This step can lead to multiple, alternative, sets of scenarios. A possible set of operational scenarios for the e-commerce idea at hand is:

- A contact searcher submits an ad to a FAP, and gets a possible contact in return. The latter means that an ad submission increases the chance for a contact searcher to find a contact s/he likes.
- A contact searcher queries for an ad on a website of a FAP, reads an ad, and pays a fee to the FAP.
- The *Ad Association* redistributes ads from FAPs to other FAPs, pays the originating FAP a fee, and gets paid by the FAPs who receive the ad.

We use these textual expressed operational scenarios on *each* viewpoint. Scenarios are represented using scenario paths, which *differ* in structure for each viewpoint. Moreover, the number of paths can be different for each viewpoint.

Many other sets of scenarios are possible, for instance a set where FAPs exchange ads on a bilateral basis (without the *Ad Association*). However, in this chapter we only consider the former set of scenarios. Note that this articulation of scenarios is a *result* of executing our e^3-*value* track. In an initial phase of a project scenarios often are described more vaguely.

9.4 Value viewpoint

After identification of operational scenarios, the next step is to design a value model and to put into operation the identified scenarios (see section 9.3) with use case maps. We use these maps to create profitability sheets, for now solely based on the value viewpoint.

Value model. Using the e^3-*value* ontology (see section 3.2), figure 9.1 presents a value model for the e-commerce idea introduced in section 9.2. From figure 9.1 it can be seen that:

- A contact searcher submits an ad (to be placed on a website) to a FAP and gets a *possible* contact in return.
- A contact searcher reads an ad and pays a FAP for this.
- A FAP gets a checked ad and pays for this. S/he can either do this his/herself or ask a colleague FAP to do so.
- A FAP resells a submitted and checked ad to publishing parties. These are the FAP his/herself, and a redistribution company, in this case the *Ad Association*. In return, a fee is obtained and a guarantee that the ad will be published.
- The Ad Assocation resells ads to FAPs, and gets a fee for this, plus the guarantee that the ad will be published.

Note that the value model introduces a value proposition that is expected to be commercially viable: *checking an ad*. This proposition was not present in the in the initial e-commerce idea. It was identified by stakeholders later in the project, because they were forced to think about value activities.

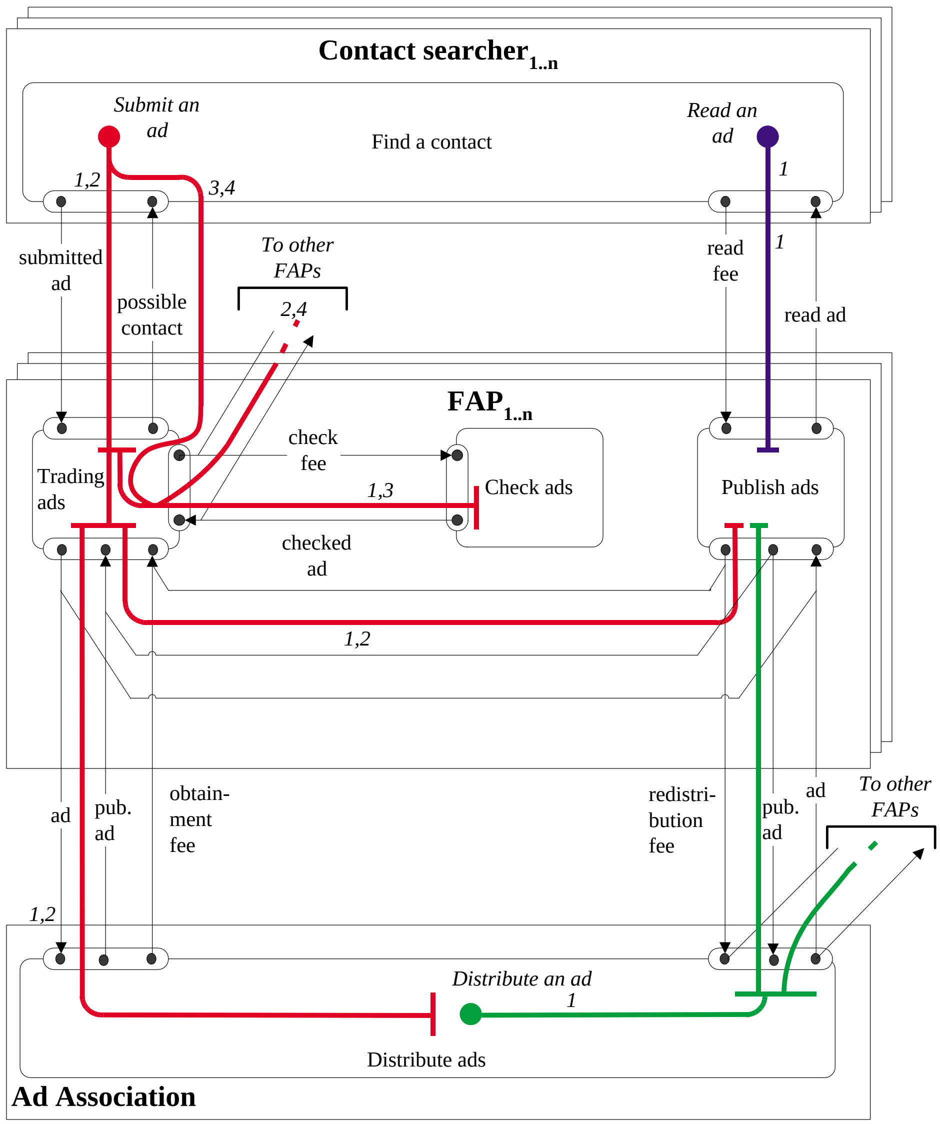


Figure 9.1: Value model with UCMs for the *Ad Association* e-commerce idea.

Value operational scenarios. Figure 9.1 shows also the UCM paths for each of the three operational scenarios identified above. The operational scenario *submit ad* has a number of scenario paths, which are labeled *1*, *2*, *3*, and *4*. Paths 1 and 2 model a submitted ad, which is *accepted* and *published* by one or more FAPs, while paths 3 and 4 represent a submitted ad which is *rejected*. Note that a rejected ad does neither result in the exchange of a *possible contact* nor in the exchange of an *ad placement* and their reciprocal value exchanges. The only value exchanges that occur is the *check ad* exchange and its reciprocal exchange.

Each scenario path touching a value interface results in a *responsibility point*. We use these points to model changes in the *profitability sheet* of an actor as a result of executing a scenario path. Changes in a profitability sheet are caused by exchanges of values between actors via their value interfaces. Therefore, scenario path elements crossing value interfaces are responsibility points by definition in a value model. If we can estimate the number of times a scenario path is executed, and we have all possible paths, we have a basic idea about the profitability of the e-commerce idea for a specific actor.

Profitability sheet: revenue and expense perspective. For the operational scenario *submit ad* we derive a profitability sheet for FAP_i (table 9.1). By following all scenario paths of an operational scenario, a list is constructed consisting of all objects of value entering or leaving the actor. As an example, table 9.1 shows all value objects entering and leaving FAP_i for the scenario *Submit ad*.

Also, table 9.1 shows *reduced* value objects between parentheses. For enterprise actors, objects representing something else than money are not of interest. We assume these objects flow into an actor, and after some time flow out. We show this in a profitability sheet by *reduced* value objects. For example, consider the first scenario path. The *submitted ad*, obtained from a contact searcher, and the *ad*, delivered to the *Ad Association* are removed from the profitability sheet, because the *submitted ad* flows into FAP_i, and after some time flows out also. All other non-money can be removed from the profitability sheet in a similar way, except the *checked ad* of scenario path 4. This ad is rejected (e.g. because it contains dirty language), but expenses have been made to discover this. The *checked ad* flows into FAP_i but never leaves FAP_i. Moreover, it is worthless because the ad can never be published. Therefore, we remove this object from the profitability sheet also.

Finally the profitability sheet shows the likelihood per scenario occurrence. We calculate the profitability per scenario occurrence, first by multiplying the profitability per scenario *path* occurrence by the likelihood of the occurrence. This results in an expected profitibality per scenario path occurrence. Second, we totalize expected profitability numbers for all scenario paths of an operational scenario. This is shown in table 9.1. Moreover, if we fill-in the fees in table 9.1, we get a first impression of the profitability of the e-commerce-idea. Also, table 9.1 can be used to perform a sensitivity analysis of the profitability, for instance during a workshop with actors about the value model. However, for a more overall view on the profitability, additional profitability sheets for the business process and information system requirements have to be developed. We use the scenario profitability and not the scenario *path* profitability of each viewpoint to calculate overall profitability, because scenario paths may differ per viewpoint, but scenarios are the same.

9.5 Business process viewpoint

The business process viewpoint illustrates processes to be carried out by actors, and messages interchanged between those actors, on a conceptual level. Because we gain more insight in *how* processes, necessary to create value, are carried out, it is possible to identify major operational expenses such as expenses caused by persons carrying out work. Responsibility points indicate such expenses.

Business process model. A number of techniques have been developed to model processes, such as UML activity diagrams with swimlanes to represent actors (Rumbaugh et al. 1999), or role-based process modeling techniques (Ould 1995). In this chapter, we choose for the latter. Ould defines a *role* as a set of activities that are carried out by an actor in an organization. An *activity* is what actors do in their roles. Between activities and thus between roles *interactions* can occur.

Figure 9.2 shows a process model which explains *how* the value model is carried out by actors. We do not show the interactions explicitly to prevent unnecessary cluttering of the diagram. Interactions are shown by the UCMs.

Business process operational scenarios. In a business process model, a UCM scenario path shows the time-sequence of messages and activities performed for a specific scenario. The same operational scenarios as in the value model are shown, however the paths now show a sequence of interactions between roles. Note the synchronization bar (with the N:1 indication) in the *distribute ad*, the *place ad* and the *publish ad* role. Such a bar ‘collects’ a number of ads, say 100, and then continues the operational scenario with one payment for all these 100 ads. This refers to the mechanism of aggregate payment (Choi et al. 1997); it is much cheaper to handle one big payment rather than a large number of small ones. We do not

Table 9.1: Profitability sheet for FAP_i for the operational scenario submit ad (value viewpoint).

Actor	FAP_i	
Viewpoint	Value viewpoint	
Scenario	Submit ad	
	Value Object In	*Value Object Out*
Scenario path	1	
Likelihood	60%	
Exchanges with contact searchers:	*(Submitted ad)*	*(Possible contact)*
Exchanges with the *Ad Association*:	*Obtainment fee* *(Placed ad)*	(*Ad*)
Scenario path	2	
Likelihood	20%	
Exchanges with contact searchers:	*(Submitted ad)*	*(Possible contact)*
Exchanges with the *Ad Association*:	*Obtainment fee* *(Placed ad)*	(*Ad*)
Exchanges with other FAP:	*(Checked ad)*	$\mathit{Check\ fee}_{FAP_{other}}$
Scenario path	3	
Likelihood	15%	
	-	-
Scenario path	4	
Likelihood	5%	
	(Checked ad)	$\mathit{Check\ fee}_{FAP_{other}}$
Expected profit	$p_{value} = 0.6 \times \mathit{distr.fee} + 0.2 \times (\mathit{distr.fee} - \mathit{check\ fee}_{FAP_{other}}) - 0.05 \times \mathit{check\ fee}_{FAP_{other}}$	

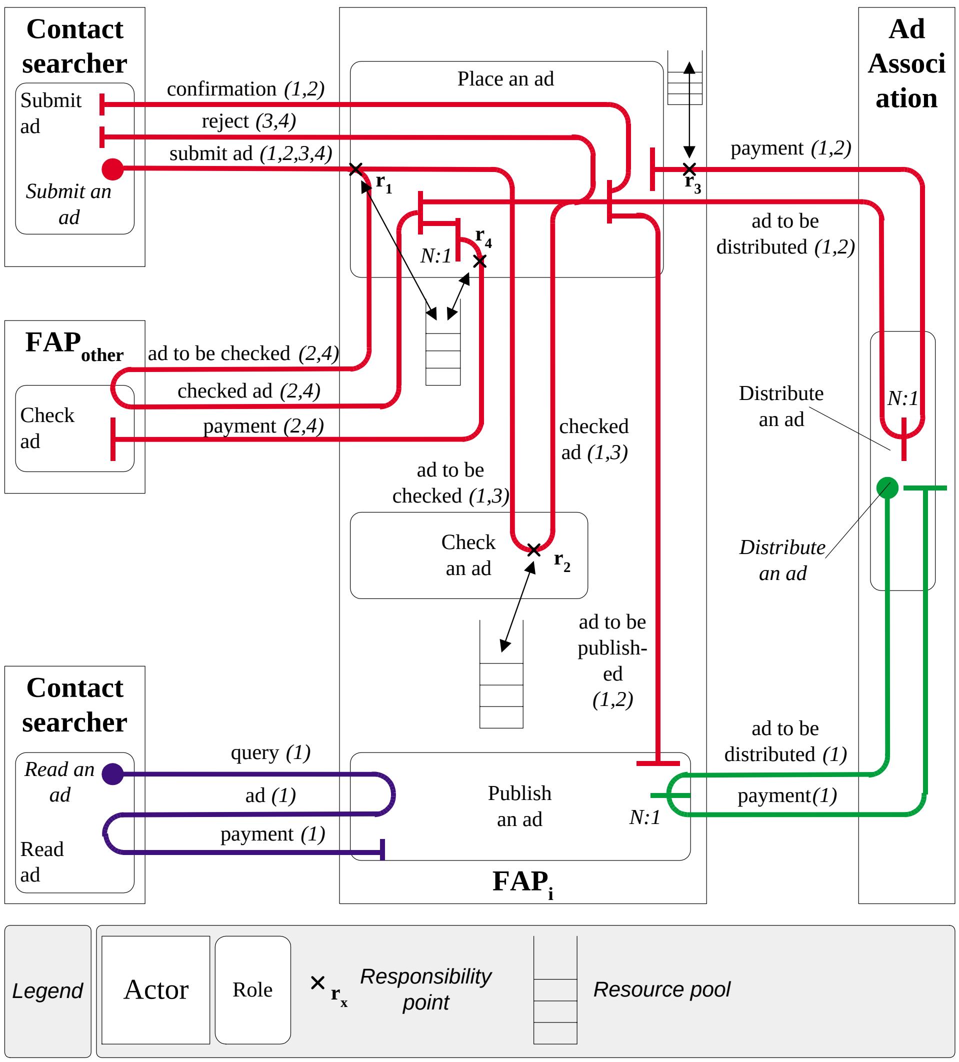


Figure 9.2: Business process model for the *Ad Association* e-commerce idea.

show aggregate payments on the value viewpoint, because on the value viewpoint we only show that something is paid for, and not how this executed in practice.

Responsibility points indicate substantial operational expenses, for instance caused by personnel. Selecting a capable checker, checking an ad itself, and administrating payments (received payments and payments done) are all tasks where humans are involved. These points are used to fill in the profitability sheet for FAP_i (table 9.2). Based on estimations of the occurrence of the scenario paths, the expected expenses for the entire operational scenario are calculated, analogue to the process described in section 9.4.

Note that, although we use the *same* operational scenarios in all our viewpoints, the scenario paths may *differ* in structure as well as in number for each viewpoint. This is caused by the different modeling perspectives of requirement viewpoints.

Profitability sheet: expense perspective. Table 9.2 extends the profitability sheet in table 9.1, but now only from an expense perspective. Revenues have been identified on the value viewpoint, the business processes only contribute expenses, in many cases in the form of personel.

The *select expenses* in table 9.2 denote that an employee must select a checker capable (e.g. possessing the right language skills) of assessing a submitted ad. Such a checker can either be employed by the FAP receiving the ad, or can be a checker of another FAP. In case an ad is checked by the FAP receiving the ad, *check expenses* are made for doing so, because an employee of the FAP must judge the ad. Note that if another FAP checks the ad, expenses for doing so have been modeled on the value viewpoint. Finally, *administrative expenses* have been identified for handling payments. These expenses are divided by N, which is the aggregation factor as a result using aggregate payments (see the business proces model).

9.6 Information system viewpoint

The information system viewpoint shows system components only on a global level. We want to see key system components which are fundamental to an information system supporting the e-commerce idea. The reason for doing so is twofold: (1) for each key component we want to estimate expenses for investments, operation and maintenance, and (2) we want to increase confidence in the technical feasibility; by exploring the information system viewpoint, technical impossibilities may come up, which influence the value model and business process

Table 9.2: Profitability sheet for FAP_i for the operational scenario submit ad (business process viewpoint).

Actor	FAP_i
Viewpoint	Business process viewpoint
Scenario	Submit ad
	Operational expenses
Scenario path	1
Likelihood	60%
	$e_1 = select\ expenses(r_1) + check\ expenses(r_2) + admin\ expenses/N(r_3)$
Scenario path	2
Likelihood	20%
	$e_2 = select\ expenses(r_1) + (2 \times admin\ expenses)/N(r_3, r_4)$
Scenario path	3
Likelihood	15%
	$e_3 = select\ expenses(r_1) + check\ expenses(r_2)$
Scenario path	4
Likelihood	5%
	$e_4 = select\ expenses(r_1) + admin\ expenses/N(r_4)$
Expected expenses	$e_{process} = 0.6 \times e_1 + 0.2 \times e_2 + 0.15 \times e_3 + 0.05 \times e_4$

model. Below we explore the information system for the contact ad idea for two information system variants.

Information system variants

Different information systems can be thought of, which support the presented value- and business process model for the contact ad e-commerce idea. We discuss two information system variants: (1) a variant with decentralized databases (figure 9.3) to be exploited by FAPs themselves, and (2) a variant with one centralized database (figure 9.4), to be exploited by the *Ad Association*. These variants both comply with the identified value- and process model, but are fundamentally different from an IT perspective.

These variants show *global* solutions for an information system supporting the contact ad e-commerce idea. We chose these variants to explore because they show decisions which directly impact the entire e-commerce model and so profitability sheets. For a decentralized variant, FAPs must invest (e.g. in equipment, software licenses and maintenance), while for the centralized variant, the *Ad Association* must invest.

Decentralized variant. System components needed for the decentralized variant are show in figure 9.3. Each FAP exploits and maintains his/her own database of ads. The database is filled by ads which are submitted via the FAPs website. These ads are also sent to the *Ad Association*. The Ad Assocation redistributes received ads via a message server (e.g. a SMTP mail server) to all other FAPs interested. In the same way, the FAP receives ads his/herself from the *Ad Association*. Finally, a reader of ads uses the local database of a FAP s/he is connected to answer his/her query.

Centralized variant. Figure 9.4 presents components needed for a centralized variant. For this alternative, the *Ad Association* maintains the database of all ads for all readers centrally. An ad which is submitted via the website of a FAP is (after checking) entered in the ad database of the *Ad Association*. If a reader wants to access an ad, s/he sends a request for an ad via the website of a local FAP, but this FAP will forward the request to the *Ad Association*. The ad is searched for in the database of the *Ad Association* and, if found, shown to the reader. Note that redistribution is not necessary anymore, since only one database consists containing all ads.

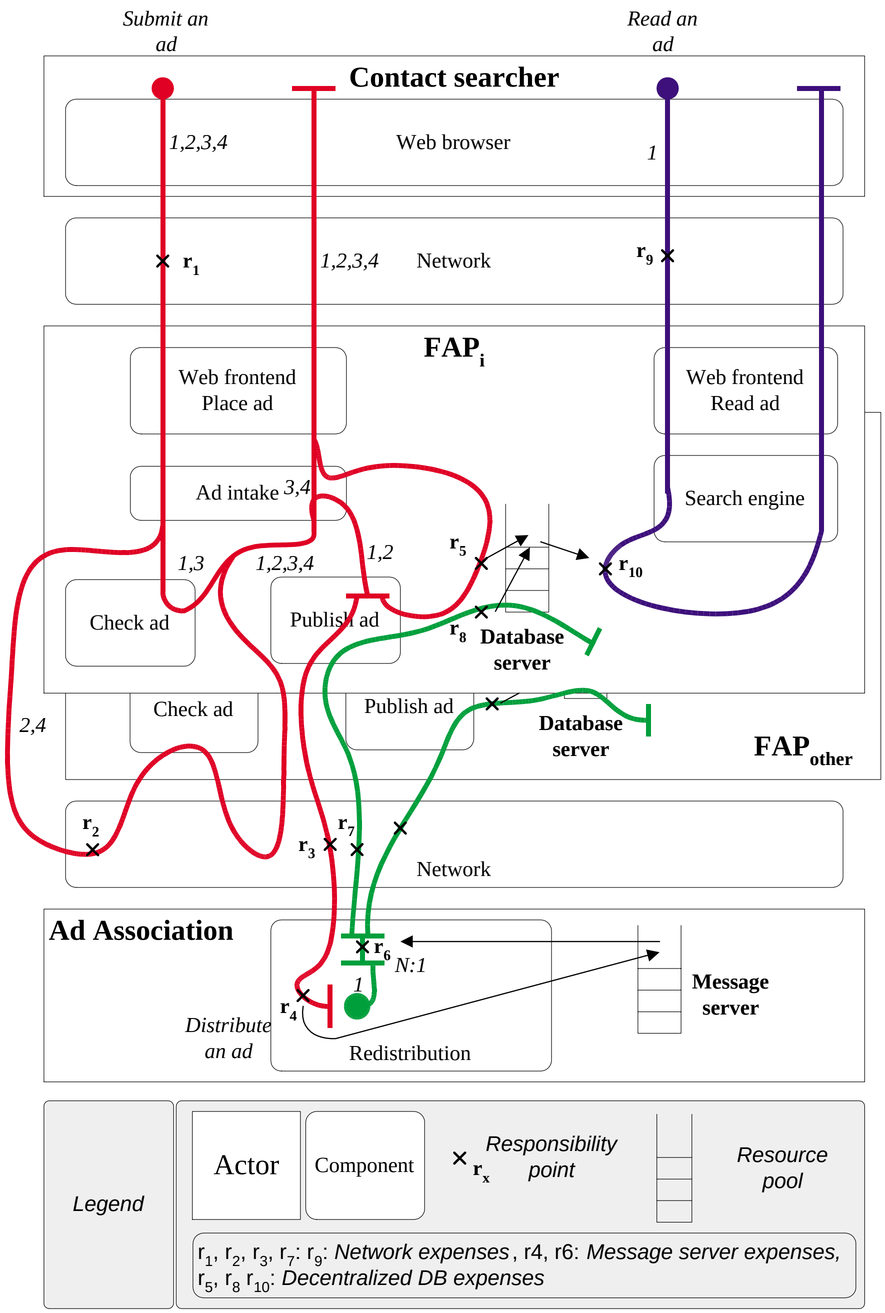


Figure 9.3: A decentralized variant. Each FAP has his/her own database. The *Ad Association* exploits a message server (e.g. SMTP-based) to redistribute ads.

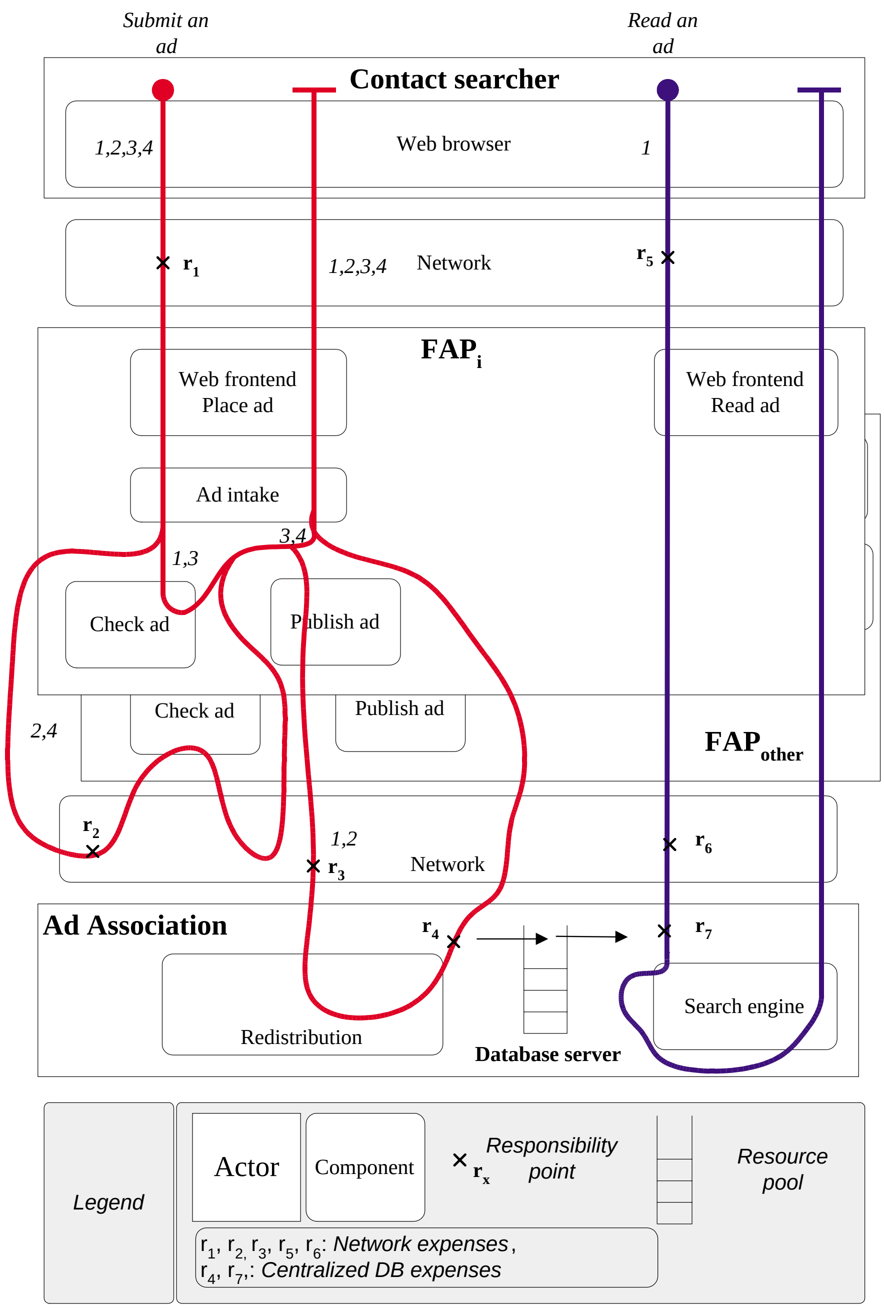


Figure 9.4: A centralized variant. FAPs rely on the central database server exploited by the *Ad Association*. Note that redistribution is not needed anymore, because there is only one database containing all ads.

Information system operational scenarios. The centralized and decentralized variants are further explained by using operational use case maps. The same scenarios as were used for the value and business process viewpoint, but with different paths, can be superimposed on the components in figures 9.3 and 9.4. Responsibility points on these global information system viewpoints indicate expenses per scenario path occurrence, e.g. expenses for updating/quering a database, for networking or for using a message server.

We also use UCM scenario paths to construct profitability sheets for the two information system variants. The expenses per scenario path, caused by using ICT components, are allocated to a scenario occurrence using the likelihood of the occurrence of a scenario path. This allows to integrate the sheets in tables 9.3 and 9.4 with the profitability sheets constructed for the business process and value viewpoints, which also account for expenses and revenues on a per scenario occurrence base.

For the variants at hand, four paths can be identified for the *submit ad* operational scenario (paths 1,2 for ads which are published and locally or remotely checked, paths 3,4 for ads which are rejected and locally or remotely checked). The other scenarios each have one path.

Profitability sheet: expense perspective

Table 9.3 and table 9.4 show expenses for the identified operational scenarios for FAPs and the *Ad Association*. Expenses for system components are paid by the actors operating these components. So FAPs pay for decentralized databases, while the *Ad Association* pays for a centralized database. Contact searchers pay networking expenses to contact FAPs. FAPs pay their own networking expenses for distributing ads and for checking ads remotely. The *Ad Association* pays networking expenses for delivering ads to FAPs.

We use the sheets in table 9.3 and table 9.4 to evaluate the two information system variants *with respect to the variation centralized versus decentralized database*. We neglect network expenses, because the current tendency is that these are much cheaper than database or message server expenses. The database server expenses comprise all expenses for having a local or central database server. For the decentralized operational scenario, we assume a message server (e.g., a SMTP server), which incurs expenses. All these expenses are accounted for on a per scenario occurrence basis. This means that no fixed expenses exist, as these are allocated to each individual execution of an operational scenario, based on the expected number of executions per time-frame. Note that the database server(s) and the message server are not part of the value and business process model, so their impact on the expenses cannot be assessed by evaluating a value or business process model in isolation. We discuss these sheets from two perspectives: (1) the *Ad Association* perspective, and (2) the FAP perspective.

***Ad Association* perspective.** If we assume that a message server is much cheaper than a database server, the expenses for the *Ad Association* in the centralized solution are greater than the expenses in the decentralized solution. This is a reasonable assumption to make, since a mail server can be implemented using a low-cost machine with nearly free software. A database server capable of a large number of queries and updates per minute represents a high investment, both in hardware and software. Also maintenance expenses are substantial, for instance for performance tuning. This increase in expenses for the centralized variant may result in a higher distribution fee, which was not identified in the business value viewpoint. Therefore, the *Ad Association* becomes a more dominant player in the centralized database solution, that is, more cash is flowing into the *Ad Association*. Moreover, because *the* database of ads will reside at the *Ad Association*, it will become a powerful player.

FAP perspective. If we consider the total database expenses for all actors involved in the centralized solution, they are less than the database expenses in the decentralized solution, under the assumption that a single database server is cheaper than all FAP-owned decentralized database servers having the same total capacity as the single server. Therefore, if FAPs would only consider direct financial effects, they are likely to choose for the centralized variant.

Discussion. We have presented both solutions at an international conference organized for the FAPs involved. It turned out that large FAPs (typically 30 to 50 employees) do not choose for a centralized variant despite the expectation that a centralized information system would be cheaper. The argument of these FAPs was that they want to stay in control of the ad database themselves, and are not willing to rely on the *Ad Association* for this. In contrast, small FAPs (typically with 5 to 10 employees) are in favor for a centralized variant because they have not the skills to operate a database management system themselves.

Table 9.3: Profitability sheet for the decentralized variant (information system viewpoint).

Actor	FAP_i	Ad Association
Viewpoint	Information system (decentralized)	
Scenario	Submit ad	
	Operational expenses	
Scenario path	1	
Likelihood	60%	
	$e_{1_{fap}} = decentr.\ dbase(r_5) + network(r_3)$	$e_{1_{aa}} = message\ server(r_4)$
Scenario path	2	
Likelihood	20%	
	$e_{2_{fap}} = network(r_2) + network(r_3) + decentr.\ dbase(r_5)$	$e_{2_{aa}} = message\ server(r_4)$
Scenario path	3	
Likelihood	15%	
Scenario path	4	
Likelihood	5%	
	$e_{3_{fap}} = network(r_2)$	
Expected expenses	$e_{fap_{inf.sys.}} = 0.6 \times e_{1_{fap}} + 0.2 \times e_{2_{fap}} + 0.05 \times e_{3_{fap}}$	$e_{aa_{inf.sys.}} = 0.6 \times e_{1_{aa}} + 0.2 \times e_{2_{aa}}$
Scenario	Distribute ad	
	Operational expenses	
Scenario path	1	
Likelihood	100%	
	$e_{4_{fap}} = decentr.\ dbase(r_8)$	$e_{3aa} = message\ server \times (M-1)(r_6) + network \times (M-1)(r_7)$
Scenario	Read ad	
	Operational expenses	
Scenario path	1	

Table 9.3: continued.

Likelihood	100%	
	$e_{5_{fap}} = decentr.\ dbase(r_{10})$	

9.7 The overall profitability sheet

In the previous sections, we have constructed profitability sheets using models and operational scenarios for each identified viewpoint. The value viewpoint contributes expenses *and* revenues, while the business process and information system viewpoint yield only expenses.

We can use these profitability sheets to construct on per actor basis a profitability sheet for the e-commerce as a whole, thus consisting of the three before discussed viewpoints plus operational scenarios. This results in a profitability number for each actor involved in the e-commerce idea. Because each viewpoint has profitability sheets which show financial effects on a scenario occurrence level, we can easily aggregate these sheets. Note that we must aggregate profits/expenses of each viewpoint on the *scenario* level and not on the *scenario path* level. Scenarios are conceptually the same for each viewpoint, but it is possible that viewpoints contain different numbers of scenario *paths* for the same conceptual scenario, or with different likelihoods of execution. Table 9.5 shows an overall profitability sheet for FAP_i, for the scenario *Submit ad*.

9.8 Conclusions

The key point of this chapter is twofold: use operational scenarios to glue viewpoints, and use the same scenarios also to assess potential overall profitability of an e-commerce idea. To keep stakeholder groups focused on the *same* e-commerce idea while exploring *different* viewpoints, we use customer grounded, operational scenarios, which are for each viewpoint the same, but are specified with different use case maps. To assess profitability for an e-commerce idea, we use revenues and expenses identified on each viewpoint to create an overall profitability sheet. We aggregate these revenues and expenses on the level of scenarios rather than on the level of scenario paths. Scenario paths may differ for different viewpoints, both in quantity as in structure, while scenarios are the same for each viewpoint.

Table 9.4: Profitability sheet for the centralized variant (information system viewpoint).

Actor	FAP_i	Ad Association
Viewpoint	Information system (centralized)	
Scenario	Submit ad	
	Operational expenses	
Scenario path	1	
Likelihood	60%	
	$e_{1_{fap}} = network(r_3)$	$e_{1_{aa}} = central\ dbase(r_4)$
Scenario path	2	
Likelihood	20%	
	$e_{2_{fap}} = network(r_2) + network(r_3)$	$e_{2_{aa}} = central\ dbase(r_4)$
Scenario path	3	
Likelihood	15%	
Scenario path	4	
Likelihood	5%	
	$e_{3_{fap}} = network(r_2)$	
Scenario	Distribute ad	
	Operational expenses	
Scenario path	1	
Likelihood	100%	
Scenario	Read ad	
	Operational expenses	
Scenario path	1	
Likelihood	100%	
	$e_{4_{fap}} = (network)(r_6)$	$e_{3_{aa}} = central\ dbase(r_7)$

Table 9.5: Overall profitability sheet for actor FAP_i for the scenario *Submit ad*.

Actor	FAP_i
Profit/scenario occurrence	$profit_{Submit\ ad} = p_{value} - e_{process} - e_{fap_{inf.sys.}}$
Scenario occurrence frequency	$f_{Submit\ ad}$ per month
Profit/month	$profit_{Submit\ ad} \times f_{Submit\ ad}$

For the case at hand, the exploration of the information system perspective resulted in two alternative viewpoints. The decentralized database variant assumes local databases for each FAP involved. It is expected that total expenses for this variant outnumber expenses for a centralized variant, for which the *Ad Association* operates a centralized database. Obtainment fees to be paid to the *Ad Association* are likely to be higher for the centralized variant than for the decentralized variant, because the *Ad Association* must exploit an expensive database management system. This consequence was not visible by only exploiting the value viewpoint. In contrast, the decentralized viewpoint results in more expenses for the FAPs themselves for having an own database management system. Also, we learned that a decision for a specific direction (decentralized vs. centralized) is not solely based on minimum overall expenses in a multi-enterprise stakeholder network. Owning content, as well as confidence in this, is also of importance.

The value, business process, and information system viewpoints plus operational scenarios are our interpretation of an *e-commerce model*. Our approach differs from other opinions on e-commerce/e-business models because we take explicitly the information system viewpoint into account. To our opinion, information systems are key to e-commerce ideas, and consequently should be accounted for in an early stage of e-commerce idea exploration.

Chapter 10

Tool support for e^3-*value*

Our experiences with e-commerce projects show that there is a need for tool-support for e^3-*value* . This is also confirmed by consultants using e^3-*value* for e-commerce idea exploration. Constructing one or more value models, creating profitability sheets and calculating profitability numbers takes too much time if done entirely manually. This hinders assessment of a substantial number of evolutionary scenarios. Especially if an evolutionary scenario captures a change in the structure of a value model, the experience is that that is too time consuming to create new profitability sheets, and to calculate profitability numbers.

At present, we only have limited tool support for e^3-*value* . Following the activities performed during e-commerce exploration as outlined in chapter 5, we support value model *construction* by a Microsoft Visio stencil, which assists in drawing e^3-*value* diagrams. To check whether a value model is e^3-*value* ontology compliant, we use a PROLOG implementation, called *EVORT*, of the e^3-*value* ontology. *EVORT* can also be used to check some business rules. Deconstruction and reconstruction is done entirely manually. Finally, evaluation is done by using an Excel spreadsheet, for which we use a standard template.

A drawback of the current tool support is that tools are not integrated on the e^3-*value* ontology level. For instance, an e^3-*value* Visio drawing conceptually consists of graphical shapes, and not of our ontology constructs. As a consequence, models made in Visio can not easily be exported to *EVORT* and to a spreadsheet for evaluation. To address these shortcomings, we will develop an integrated toolset to support e^3-*value* modeling in the near future . Development of this envisioned tool support is an activity of the EC-funded IST project OBELIX .

In this chapter, we first review the current tool support for e^3-*value* (section 10.1). Hereafter, we present envisioned tool support in section 10.2. We do so by taking the *e*^{3}*-value* exploration activities outlined in chapter 5 as a starting point. Finally, section 10.3 summarizes this chapter.

10.1 Existing tool support

There is limited tool support available for *e*^{3}*-value* . Using the activities: (1) value model construction, (2) value model deconstruction and reconstruction, and (3) value model evaluation (see chapter 5), we discuss the current tool support. Moreover, in the coming sections, we identify shortcomings experienced in the current toolset, to envision a new *e*^{3}*-value* toolset.

10.1.1 Value model construction

For value model construction, we use two tools: (1) Microsoft Visio with an *e*^{3}*-value* stencil for drawing various value viewpoints, and (2) a PROLOG implementation of the *e*^{3}*-value* ontology for value model representation and checking.

Visio stencil for *e*^{3}*-value*

Microsoft Visio is a generic schematic drawing software package. It uses *stencils* which contain prototypical shapes. We have developed an *e*^{3}*-value* stencil containing the visualizations for the constructs in our ontology such as actors, value interfaces, value exchanges, value activities, and market segments. A screenshot of the *e*^{3}*-value* stencil with a value model is shown in figure 10.1.

The advantage of Visio is that is easy to develop a stencil, and that with minimum effort some tool support can be offered. Using predefined shapes for *e*^{3}*-value* modeling constructs reduces drawing time.

There are however serious drawbacks, caused by the fact that Visio is mainly a drawing tool. First, elements of a value model are only known as shapes (graphical constructs), and not as *e*^{3}*-value* ontology constructs. As a consequence, Visio does for instance not know that a value interface belongs to an actor; that a value interface has ports, and so on. Second, viewpoints are in no way related. Visio's has a facility for making multiple graphical diagrams (being viewpoints) in a file (representing a value model) but has no way of relating these diagrams, and ensuring consistency. Third, the Visio model can not be checked automatically (e.g. for compliance with the ontology). Fourth, profitability sheets can not be generated automatically.

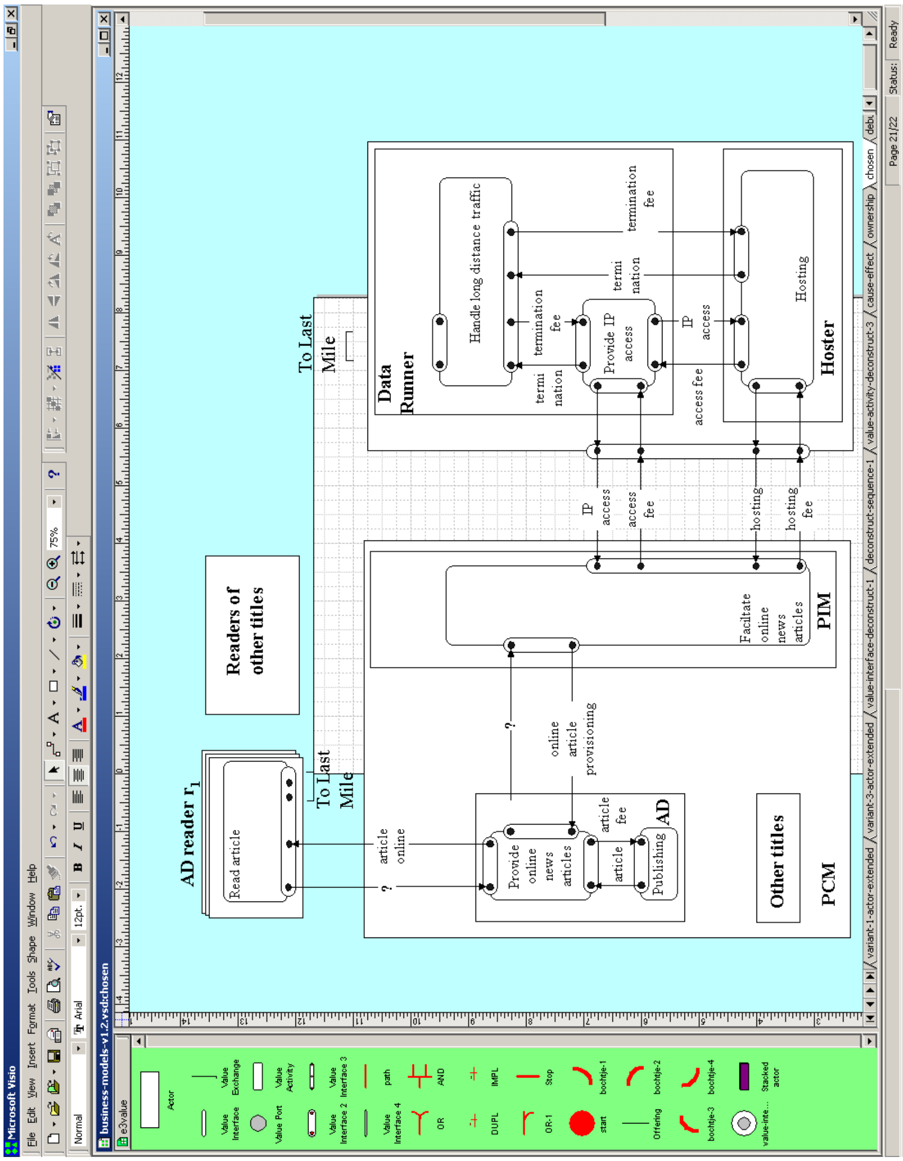


Figure 10.1: Screenshot of Visio. The left pane shows the e^3-*value* stencil, the right pane shows a presentation of a value model.

EVORT: a tool for representing and validating *e*^{3}*-value* models

While Visio stencils focus on the graphical presentation of value models, *EVORT* (*e*^{3}*-value* Ontology Representation Tool, (Schuitemaker 2000)) is a tool that knows the *e*^{3}*-value* ontology concepts and relations, and can be used to represent an *e*^{3}*-value* model on a conceptual level. *EVORT* is based on a preliminary version of the *e*^{3}*-value* ontology (see Gordijn et al. (2000*b*)), and uses object oriented PROLOG (*SICStus Prolog version 3.8 manual* 1999). *EVORT* focuses on *e*^{3}*-value* ontology concepts only, it does not provide representation facilities for additional mechanisms used on top of *e*^{3}*-value* such as operational scenarios (use case maps). This implementation facilitates representation of a value model by *specializing* and *instantiating* the *e*^{3}*-value* ontology. Also, it is possible to check whether a value model complies with the ontology constraints and rules. Finally, elementary queries can be done.

Specialization and instantiation of the *e*^{3}*-value* ontology. To construct *e*^{3}*-value* ontology-based value models, we use a two-layered approach. First, the ontology concepts and relations are *specialized* into an ontology for a specific e-commerce application domain. For example, the *e*^{3}*-value* is specialized into an ontology capable of representing free Internet access e-commerce ideas. This step is optional, but can facilitate reuse of value model fractions in a value model or in alternative models. Second, the concepts and relations of the specialized *e*^{3}*-value* ontology are *instantiated* for a specific e-commerce idea.

The idea of ontology specialization is borrowed from Borst, Akkermans & Top (1997). They use ontology specialization as an ontology projection operator that modularizes a large ontology into smaller parts. This modularization facilitates understanding of the ontology and reuse of ontology parts.

From a technical perspective, *EVORT* consists of a number of OO-PROLOG classes and a number of relations, which correspond to concepts and relations identified in the *e*^{3}*-value* ontology. In addition, methods have been defined which allow for the creation and deletion of specialized concepts and relations of the *e*^{3}*-value* ontology, as well instantiating these concepts and relations to build a value model.

Constraints and rules checking. *EVORT* can check a specialized ontology and a value model with respect to: (1) *e*^{3}*-value* ontology compliance, (2) ontology specialization and instantiation, and (3) compliance with business rules.

Ontology compliance. The *e*^{3}*-value* ontology consists of concepts and relations, but contains also constraints such as cardinality, equality, totality and exclusivity constraints. *EVORT* can check whether a specialized ontology complies with the e^3*-value* ontological constraints. Similarly, it can be checked whether the (specialized) ontology has been instantiated correctly.

Ontology specialization and instantiation. A specialized and instantiated e^3*-value* ontology should be consistent with the ontology constraints, but has an additional requirement. Specialization and instantiation occur in disjoint layers. It is not allowed to relate a specialized concept with an instance of a (specialized) concept using one of the ontology relations. We do so, because a value model should be about specific enterprises and end-consumers and thus contains always specific identifiable or non-identifiable actors (see section 5.3.2). In contrast, a specialized ontology only says that a specific kind of actors must exist, but does not refer to the instance yet. We do not want to mix up these various kinds of actors. *EVORT* can check if specialized concepts are only related with other specialized concepts of the ontology, and if instantiated concepts are only related with other instantiated concepts.

Business rules. A value model should obey certain business rules. *EVORT* can check a number of such rules. As an example, *EVORT* checks whether a value interface has at least one in- and one out port, whether *all* ports of a value interface are connected with ports of other value interfaces , and whether the right type of value exchanges are used to connect value ports.

Queries. A value model which is represented using *EVORT* can be queried for a number of elementary questions. Specifically, an *EVORT* user can ask whether all actors, all value interfaces, and all ports are connected by value exchanges. Also, the user can determine if all value objects are used, and if all value activities have exactly one performing actor. An example of a useful query which has not been implemented is the determination of value interface similarity. Value interfaces of different actors are similar if have they exchange exactly the same value objects into the same direction. Such a query can be used to find potential market segments, because a market segment groups value interfaces of actors who are supposed to value objects equally. Value interfaces of actors in such a segment should however exchange the same objects into the same direction. Finally, using a query, consumer value experienced by an end-consumer actor can be retrieved. Consumer value is then calculated using the formula as explained in section 7.5.1.

10.1.2 Value model deconstruction and reconstruction

Value model deconstruction and reconstruction is not supported in the current *e^3-value* toolset. It is now an entirely manual task. In section 10.2.4, we will discuss ways to support value model deconstruction and reconstruction.

10.1.3 Value model evaluation

To evaluate a value model we create profitability sheets for actors and use evolutionary scenarios to assess sensitivity for foreseen future events and misassumptions (see section 5.6). There is no tool support available for generating profitability sheets based on a value model and operational scenario paths. However, to facilitate the creation of profitability sheets, we use a predefined Excel spreadsheet (see figure 10.2), which we fill in following the steps outlined in section 5.6.

Figure 10.2 shows parts of the sheets we used to evaluate the *online article* e-commerce idea, as discussed in chapter 8. We use four types of sheets: (1) the actor profitability sheet, (2) the value transaction/value exchange sheet, (3) the scenario sheet, and (4) utility sheets such as ladder tables.

Actor profitability sheet. The actor profitability sheet (figure 10.2 (a)) shows for all actors the estimated profitability or consumer value on various abstraction levels. Profitability contribution is shown on the actor level, but also on value interface and scenario path level.

Value transaction/value exchange sheet. The actor profitability sheet uses the value transaction/value exchange sheet (figure 10.2 (b)) to calculate effects of value objects flowing into and out an actor as a result of scenario path execution. To do so, valuation functions are used, which in turn may use valuation properties such as observable product/service properties, or subjective properties set by an actor.

Scenario sheet. To produce a profitability sheet for actors per timeframe (e.g. a month) we need to estimate the number of scenario occurrences per timeframe and path likilihoods. By using these numbers, and the financial effects of value exchanges, we calculate the actor profitability sheet. Figure 10.2 (c) shows estimates for the actual scenario occurrences (to be realized), but shows also estimates on forecast scenario occurrences. In this specific case, pricing for some objects depends on the *forecast* number of scenario occurrences and not on the the actual occurrences.

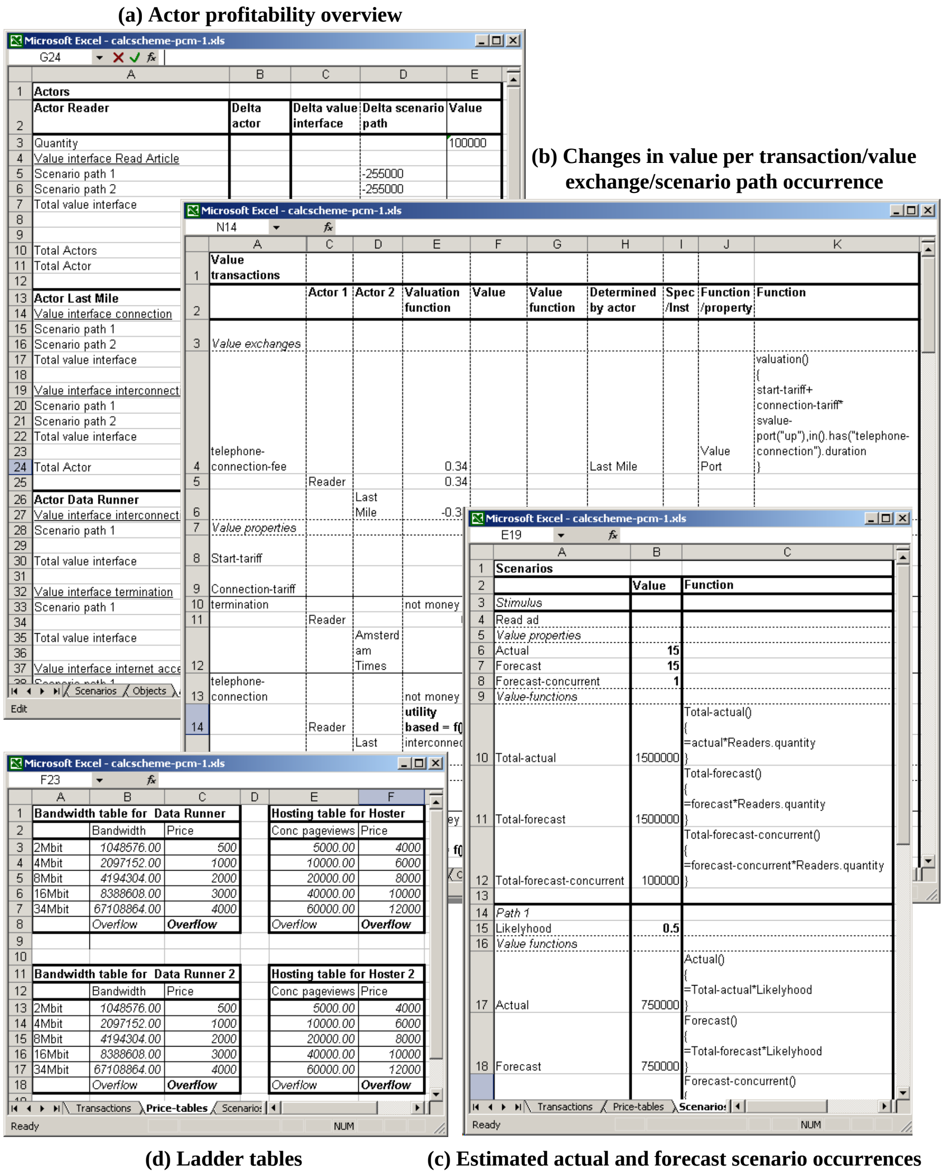


Figure 10.2: Screenshot of Excel sheets used to evaluate a value model.

Utility sheets. Sometimes, utility sheets are needed to calculate profitability numbers. Figure 10.2 (d) shows ladder tables, which are used by Internet service provisioning actors and access provisioning actors to price access-based on the bandwidth requested by the customer.

Evolutionary scenarios

Most evolutionary scenarios assume a change in the way actors value objects, or capture a change in the number or likelihood of scenario path occurrences. These scenarios are evaluated by changing values in the Excel spreadsheet. Assessing changes in the *structure* of a value model are more difficult to analyze with the current toolset. At least parts of the Excel sheets have to be reconstructed, which is time consuming. What lacks is integration between a value modeling tool and a spreadsheet for evaluation purposes.

10.1.4 Discussion

The main drawback of the current toolset is that tools are not integrated. During construction of a value model, this is experienced when multiple viewpoints are developed with overlapping information (e.g. an actor that is shown on both viewpoints). There is no way to keep such value viewpoints consistent. Also, because our drawing tool is not coupled with our *EVORT* ontology implementation, it is not possible to check a value model directly. Based on a Viso drawing, an *EVORT* representation has to be created for a specific value model manually. Moreover, evaluation of alternative models is time consuming. First, for each value model, spreadsheets have to be created manually, although these sheets are based on templates. Second, evolutionary scenarios representing changes in the structure of a value model can not be studied easily, because such a change causes major changes in the profitability sheets. Finally, deconstruction and reconstruction is done manually. In the next section, we use these experienced shortcomings to envision a new toolset for *e*^{3}*-value* .

10.2 Envisioned tool support

10.2.1 Tool development context: the OBELIX project

In the near future, we will develop an integrated toolset for supporting the *e*^{3}*-value* approach. These tools will be a deliverable of the EC-funded IST project OBELIX

(Obelix consortium 2001). OBELIX aims to develop an e-commerce ontology tool suite and library to support smart collaborative e-commerce and the realization of innovative e-commerce applications. The OBELIX tool suite consists of an integrated toolset to support the e^3-*value* approach. Moreover, facilities will be added to specify complex value objects in more depth to support the development of content management standards for complex products and services. Complex value objects are for instance value objects offered by a value constellation rather than by an individual actor. Also configuration tools for managing the production, delivering and consumption of such complex objects will be implemented, and validated using three demonstration projects in the realm of e-markets for energy trading and servicing, new digital music value constellations, and online design of events. To facilitate these tools, a generic ontology server providing facilities for editing, component brokering, ontology management, and Web language import and export, will be developed.

10.2.2 An integrated toolset for developing value models

As observed in section 10.1.4 a main drawback of our current e^3-*value* toolset is that the tools are not integrated at the level of the e^3-*value* ontology. To address this lack of integration, figure 10.3 shows the main components of the *integrated* toolset to be developed.

The *value model repository* is a database that stores value models constructed and associated information such as needed for visualization, evaluation (e.g. properties needed for valuation, and evolutionary scenarios). The value model repository is a means to achieve data integration; all other tools parts of the toolset which manipulate a value model should do so by retrieving the value model from the repository, and storing the model once finished. The value model repository will directly be based on our e^3-*value* ontology, plus facilities for storing visualization parameters, and data needed for evaluation. From a technical perspective, we will use the OBELIX ontology server to store our e^3-*value* ontology.

The *drawer*, *checker*, *constructor*, and *evaluator* offer each functionality for deconstruction and reconstructing value models, for checking such models, e.g. for compliance with the e^3-*value* ontology, and for evaluating models respectively. In the next sections we discuss per value model design activity foreseen functionality.

Finally, all tools have a common *user interface*. This means that from a user perspective, the toolset should present itself as one tool, rather than consisting of several tools.

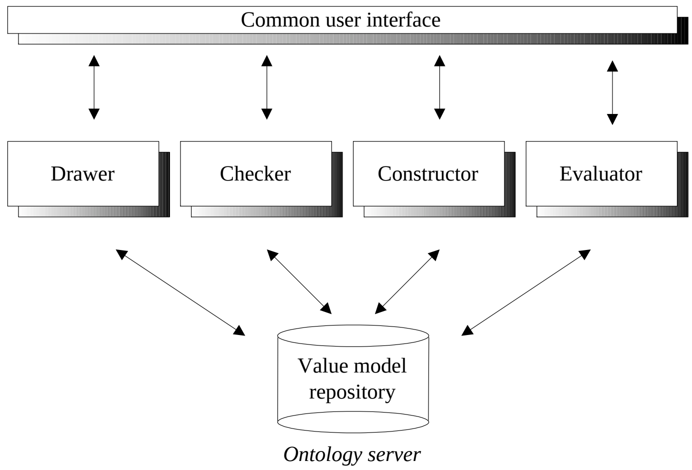


Figure 10.3: An integrated toolset for developing value models.

10.2.3 Value model construction

Drawing a value model. The *drawer* component provides functionality for drawing value viewpoints. We have experienced by using the *e^3-value* Visio stencil that *free hand editing* is a convenient way for the user to make such graphical representations. With free hand editing, it is possible to create an incomplete and inconsistent model. The advantage of free hand editing is that a user need not to ensure that his/her model is *always* correct and consistent with the *e^3-value* ontology. Moreover, the process to be followed for creating a model is not guided by ontology compliance and consistency requirements, but can be done as the modeler prefers (even ad hoc). A potential drawback of free-hand editing however, is that not much guidance is offered during drawing, e.g. regarding potential incorrect model fragments.

A usable compromise is offered by the DiaGen (Minas & Hoffmann 2001) generic graphical modeling toolset. DiaGen is a Java-based toolset, with which drawers for graphical modeling techniques can be specified and generated. A diagram technique is described by DiaGen as a hypergraph (used to specify diagram representation), and by a hypergraph grammer (used to specify diagram syntax). Additionally, Java subclasses are used for the visualization of concepts, which are used by a specific technique. DiaGen-based graphical editors support free hand editing. Users can arbitrarily create, delete and modify diagram components. However, after each editing operation, the drawer analyzes the diagram, using the predefined hypergraph grammer, and shows syntax errors (by coloring syntactical inconsistent fragments). In our drawer, we will use this out-of-the-middle approach: facilitating free hand editing, while checking diagram syntax on-the-fly.

Annotating a value model. The *drawer* is also used to annotate a value model with information needed to do evaluations and for the creation of the profitability sheets. Model elements such as actors, value interfaces, value activities, value exchanges and value objects, but also scenarios and scenario paths can have properties with a value, which can be used by a valuation function to calculate the economic value an actor assigns to obtaining or delivering a value object. Valuation functions themselves can be specified using the *drawer* component, for value exchanges and value ports. A valuation function of a *port* models that the actor owning the port determines the valuation function, while a valuation function assigned to a *value exchange* represents that valuation is done by both actors involved in that value exchange.

Checking. During model drawing, a user can validate whether his/her model complies with the e^3-*value* ontology from a syntactical point of view. The *checker* component adds additional value model validation means. These are based on checks mentioned in section 10.1.1.

We foresee that further use of e^3-*value* will reveal additional constraints and business rules. Therefore, the *checker* component should be flexible in the extension of contraints and rules to be checked. To do so, we will continue to use our PROLOG-based checking component (*EVORT*). New constraints and rules then can be codified in PROLOG, and plugged into the e^3-*value* toolset.

10.2.4 Value model deconstruction and reconstruction

In chapter 6, value model deconstruction and reconstruction is introduced as a way to find variations on an earlier found value model. Although the deconstruction and reconstruction process itself is a creative, human task, a *constructor* component can support the deconstruction and reconstruction process by: (1) deconstruction preparation, (2) facilitating deconstruction operators, and (3) reconstruction.

Deconstruction preparation. First, the *constructor* component creates a value model to be reconstructed. The component does so by taking an already existing value model (the source model), and by removing the performing actors from that source model. What remains is a set value activities, which are connected by value exchanges.

Deconstruction operator support. For each deconstruction operator, support can be offered. For the *value activity* deconstruction operator, the component facilitates splitting of value activities, and the assignment of existing value interfaces to the newly found activities. Also, it can be validated if the deconstructed model fragment is invariant to its environment, by checking if the deconstructed model fragment has at least the same value interfaces as the source model fragment. The *value port* deconstruction operator can be facilitated in a similar way as the *value activity* deconstruction operator. More specifically, a *constructor* component should support deconstruction of peer-ports, and deconstruction of value exchanges connected to these (see section 6.3.2, step 3). Finally, value interface/offering deconstruction can be supported by following the steps in section 6.3.3

Reconstruction. A *constructor* component should offer facilities for value model reconstruction. First, various value configurations can be created and stored in the repository. These configurations are deconstructed value activities connected by value exchanges. They can be seen as complete value models but without performing actors. A second step in reconstruction is re-identification of actors and their interest in performing one or more value activities. A *constructor* component should allow to make an *actor-value activity assignment matrix* as explained in section 6.4, to capture these interests. Based on this matrix and value configurations, alternative, reconstructed, value models can be generated.

10.2.5 Value model evaluation

Creation and calculation of profitability sheets. The *evaluator* assists in the creation of profitability sheets for each actor in the value model under study. A generic structure for the profitability sheets has been presented in figure 10.2. To create profitability sheets, the *evaluator* component traverses all scenario paths in a value model and updates the profitability sheet of actor, if the scenario path touches a value interface of that actor. Then, profitability numbers for each actor are calculated, by using values assigned to valuation properties, by using expressed valuation functions, and by using numbers on scenario path likelihoods and scenario occurrences.

Evolutionary scenarios. An important advantage of integrated tool support for e^3*-value* is the possibility to assess a substantial number of evolutionary scenarios. These scenarios capture variations on the structure of a value model, on the valuation of objects by actors, and on the number of estimated scenario occurrences and path likelihoods. An *evaluator* component must be capable of representing these variations, and must produce, for a given number of evolutionary scenarios, profitability consequences for selected actors in a tabular form (see table 8.6 for an example).

Net Present Value analysis. Finally, the *evaluator* component offers Net Present Value analysis (see section 5.6.2). Using the Net Present Value technique, we can assess profitability of an e-commerce idea over a number of sequential time periods. Each period may have a different value model, different valuations of objects by actors, and different numbers on estimated scenario occurrences and path likelihoods. Periods may have different numbers on scenario occurrences and path likelihoods, because many promising e-commerce ideas show an increase in the number of scenario occurrences to model an increasing sales of products or services offered. It is needed to allow different value models for time periods, because we have seen that value models change in time, e.g. because value constellations are dynamic themselves, and because of environmental changes. Tool support should be able to capture these phenomena.

If during evaluation multiple sequential time-periods are distinguished, an *evaluator* component uses the standard Net Present Value calculation (Drury 1998) method to account for the value of money over time. By doing so, we take into account interest effects of money.

10.3 Summary

To allow for a fast exploration of innovative e-commerce ideas, tool support is needed. This support should be grounded on the e^3-*value* ontology, and should offer functionality for drawing e^3-*value* models and viewpoints, for checking models (e.g. for ontology compliance and business rules), and for evaluating value models.

Nowadays, only limited tool support is available. Moreover, these tools are not integrated on the e^3-*value* ontology level. This hinders assessment of a substantial number of evolutionary scenarios, especially if such scenarios capture changes in the value model itself.

We will address this need for tool support in the IST project OBELIX. An integrated toolset will be developed, offering facilities for drawing value models, checking and deconstructing and reconstructing these, and generating profitability sheets. Also, evaluation and evolutionary scenario representation will be handled by this toolset.

Chapter 11

Conclusions and future research

This chapter presents conclusions and directions for further research on exploration of innovative e-commerce ideas. To do so, we start this chapter with the key points of this thesis (section 11.1). Thereafter, in section 11.2, we revisit the research question (see section 1.2) and present how and to which extent this question has been addressed in this thesis. Section 11.3 outlines directions for future research. Finally, section 11.4 discusses the future of value-based requirements engineering.

11.1 Key points of this thesis

This thesis is about the exploration of innovative e-commerce ideas, which utilizes principles from both requirements engineering and conceptual modeling, and focuses on the exploration of an information technology intensive value proposition. We call such an exploration track *value-based requirements engineering*.

Based on observations made during e-commerce idea exploration tracks, we motivate the need for an e-commerce *model*, rather than a vaguely described *idea* (see chapter 2). Development of such a model serves two goals: (1) enhancing agreement and a common understanding of an e-commerce idea amongst a wide group of stakeholders, and (2) enabling validation of the e-commerce idea in terms of evaluating economic feasibility. Additionally, an e-commerce model can be used as a starting point for a more detailed requirements engineering process. Based on experiences in exploring e-commerce ideas, such a model-based approach should be:

1. a *lightweight* approach to address the only limited time-span available for doing exploration tracks;

2. a *graphical conceptual modeling* approach to enhance a common, more precise understanding of the idea amongst stakeholders, and to allow for validation by evaluation of the e-commerce idea;

3. a *multi-viewpoint* approach to deal efficiently with the different interests of a multi-stakeholder group. We distinguish three viewpoints, being (1) the business value viewpoint, (2) the business process viewpoint, and (3) the information system viewpoint.

4. a *scenario* approach, which can be subdivided in an *operational* scenario approach and an *evolutionary* scenario approach. Operational scenarios are used to relate stakeholder viewpoints, and to express viewpoint specific semantics. We employ evolutionary scenarios to do a *what-if* assessment for an e-commerce idea.

5. an *economic value aware* approach, to explicitly account for the financial effects of the execution of an e-commerce idea, thereby gaining insight and confidence into the feasibility of an e-commerce idea.

Some viewpoints, such as the business process viewpoint and the information system viewpoint can be adequately specified using established techniques, but a suitable specification vehicle for the business value viewpoint lacks. To address this shortcoming, we have developed an ontology, called e^3-*value* , with on top of it a well known operational scenario specification technique called Use Case Maps (UCMs) (see chapter 3). Such a value model shows how in a multi-actor network objects of economic value are created, distributed, and consumed. To present the model in such a way that stakeholders reasonably can understand the model, we have also developed a way to visualize value models. Furthermore, the ontology is enriched with constraints and rules. This ontology and associated constructs are one of the main contributions of this thesis. In general, requirements engineering approaches neglect the value proposition of an information system. However, understanding of this value proposition is key to the development of e-commerce intensive information systems.

Our economic value-based modeling approach is sometimes confused with business process modeling approaches. Although we borrow some thoughts from the business process modeling community, value models represent different stakeholder decisions, have different modeling constructs, and show different Universe of Discourse statements. A value model is about who is creating something of value for whom, in a profitable way, while a business process model shows the activities, the sequential ordering of these, and resources needed to put a value model into practice (see chapter 4).

For practitioners, it is not only sufficient to know *how* to represent a value model, but they also need guidelines how to *construct* and *validate* such a model. Therefore we propose in addition to an ontology a way of working for inexperienced e^3-*value* users, which consists of the construction of one or more baseline value model(s), finding variations on these, and validating such models by profitability and sensitivity analysis using an evolutionary scenario approach (see chapter 5).

To find variations on a stated business value model, we use an approach called value model deconstruction and reconstruction (see chapter 6). By doing so, we study: (1) variations in the assignment of value activities to actors, (2) variations in value objects offered and requested by actors, and (3) variations in reciprocity of objects offered and requested as well as variations in bundles of objects offered and requested. Because we have a lightweight ontology consisting of a small number of concepts and relations, we have also a limited number of deconstruction operators, which act on a given value model, and which can be used to find these variations.

The e^3-*value* methodology has been used in a number of explorative e-commerce projects. Moreover, e^3-*value* has been evolved as a result of doing such projects in an Action Research like style. Examples of these projects are presented in chapters 7, 8, and 9.

First, we exemplify e^3-*value* from an enterprise, profitability oriented perspective. For an online news article service idea, we show the exploration and evaluation track. This idea has been put into operation some time ago, allowing us to do a longitudinal assessment of the idea. As a lesson, we saw that is important to model all profit-responsible parties, who have a commercial stake in the idea, also if such actors are part of a same company. Also we learned that a potential successful e-commerce idea is not sufficient; it should also be marketed well. The marketing viewpoint is currently not part of e^3-*value* .

Second, we discuss how e^3-*value* works in an end-consumer setting. We present how e^3-*value* has been used for the music industry to clarify effects of the illegal music copying scene. It shows how value objects can be quantified in monetary units utilizing an interpretive qualitative consumer value approach, grounded in axiology.

Third, we show how value model exploration interacts with exploration of other viewpoints. A potential danger of a multi-viewpoint approach is that stakeholder groups develop diverging perspectives for the e-commerce idea at hand. To address this issue, we employ operational scenarios to relate requirements on the various viewpoints (chapter 9). We use the *same* operational scenarios for each requirements viewpoint, which are put into operation by *different* use case maps for each viewpoint. A second motivation to use such scenarios is to integrate financial effects of different viewpoints into one profitability sheet per actor.

Finally, we have experienced that integrated e^3-*value* tool support is needed to work with e^3-*value* (see chapter 10). Using a way of working with e^3-*value* as outlined in chapter 5, we discuss available tools for e^3-*value* , such as a drawing tool, a tool for checking e^3-*value* models, and a spreadsheet template we use for evaluation. Based on an analysis of experienced shortcomings in the current toolset, functionality for an envisioned toolset is presented, which will be implemented by the OBELIX IST project.

11.2 Reviewing the research issues

Our research question was (see section 1.2):

- How can we precisely define an innovative e-commerce idea such that it is clear to all stakeholders and allows for profitability evaluation?

We have researched this question in an Action Research like fashion, to construct a theory, called e^3-*value* , on innovative e-commerce idea exploration. Also, e^3-*value* has been benefited from theory on requirements engineering, conceptual modeling, and organizational science and axiology. We will review the research question along the lines of these observations made in chapter 1.2:

1. *Information technology knowledge is key to many e-commerce ideas*.

   We have addressed the influence of information technology knowledge on e-commerce idea exploration, by taking explicitly an information system perspective into account during an exploration track. Exploring such a perspective should identify drivers for substantial operational and capital expenses. Chapter 9 presents how we use such a viewpoint in conjunction with other requirement viewpoints, such as the business value and business process viewpoint.

2. *A wide range of stakeholders, ranging from CxO's to information technology concerned persons is involved*.

   To address the wide range of stakeholder interests, we use at least three stakeholder groups with associated requirement viewpoints. These viewpoints are (1) the business value viewpoint, capturing objects flowing into and out actors, (2) the business process viewpoint, representing how the business value viewpoint can be put into operation, and yielding insight in operational expenses, and (3) the information system viewpoint, showing the main system components needed, and providing knowledge on major operational and capital expenses. To prevent diverging viewpoints as a result of various stakeholder interests, we keep viewpoints related by means of operational scenarios, which are the same for each viewpoint, but are formalized by different use case maps, depending on the semantics of the specific viewpoint. These operational scenarios serve also as a glue to create overall profitability sheets, based on all three viewpoints.

3. *Many e-commerce ideas are described only vaguely, thereby leaving room for multiple interpretations*.

   In this thesis, we use well-known modeling principles and techniques to develop and represent models on the three viewpoints. For the business process and information system viewpoints, we use standard techniques from the realm of requirements engineering and conceptual modeling. Our main contribution is an ontology which can be used to express models on the business value viewpoint, to find variations on these models and to evaluate value models in terms of expected profitability (for enterprises) or increase in consumer value (for end-consumers).

4. *Idea exploration may take only a limited period of time, typically a few weeks*.

   Our e^3-*value* ontology has been designed to be lightweight. Therefore, the ontology contains only a very limited number of concepts and relations between these concepts. This can be seen as a partiality in a language to express such models, but to keep the exploration really lightweight, partiality in modeling is also important. Therefore, we only develop global models, which capture the essence of an e-commerce idea. Moreover, we only focus on substantial expenses and revenues. During exploration, finding a direction the e-commerce idea may take is more important than a detailed conceptualization.

5. *A focused and unambiguous e-commerce idea should also be feasible*.

   Economic feasibility is assessed by evaluating the idea from a profitability and consumer value perspective. These profitability sheets are only best estimates; hard numbers on profitability are for innovative e-commerce ideas simply not realistic. By creating profitability sheets, we increase confidence in the sustainability of the idea, providing that such sheets have positive numbers. Confidence can be increased even further by using evolutionary scenarios to do a sensitivity analysis. By evaluating an idea this way, we reason about profitability of the idea, while making our assumptions in doing so explicit.

11.3 Future research

The e^3-*value* methodology is only the beginning of value-based requirements engineering. Many work is yet to be done to understand information technology intensive new business development. In this section we present research topics to be investigated.

From sensitivity analysis to predictions. The profitability sheets are now estimates to do a sensitivity analysis and should not be seen as predictions for profitability and consumer value. Enterprises and business developers are however interested in such predictions. Reliable estimations depend on a sound forecasting of valuation of value objects by actors, the number and likelihood of scenario path occurrences, and expenses seen from a business process and information technology perspective. Also, the structure of the value model must correspond to reality. The number of scenario occurrences and path likelihoods are hardly known in advance. Because we explore *innovative* value propositions, we can not rely on historical data. In practice, such numbers can only be found by doing market research, and even then it is difficult because it is not very well possible yet to predict whether an innovative idea will be adopted. Other factors having financial effects are the kind of business processes and information system components chosen. An approach which may lead to better predications is to use known benchmarks which indicate expenses of a particular solution on the business process and information system viewpoints, given a value model and numbers on scenario occurrences and likelihoods. For instance in the case of the online news article e-commerce idea (see chapter 8), for serving only two articles online per minute a lightweight webserver may be sufficient, while for thousands of articles per minute a heavyweight solution such as a load-balancing farm of webservers is needed.

A marketing viewpoint. As observed in chapter 8, exploration of a marketing approach should be part of an e-commerce exploration track. How to do so, preferably model-based, is a topic of future research.

Value model patterns. In the realm of information technology, analysis (Fowler 1997) and design patterns (Gamma et al. 1997) are emerging. A pattern describes a problem which occurs over and over again in an environment, and describes one or more solutions for the identified problem as well as consequences (e.g. trade-offs) as a result of applying the pattern. For value models, also such patterns may be developed, which address a particular business issue (e.g. how can I retain customer ownership), and show possible solutions how to do so. Moreover such patterns may be related to already existing business process and information system patterns, to show how particular business needs can be fulfilled with business processes and information systems.

Viewpoint relations. Our three requirement viewpoints are now loosely coupled by operational scenarios to enhance consistency between these viewpoints. How can these viewpoints be related more closely, so that requirement conflicts as a result of using multiple requirements viewpoints can be detected. Additionally, requirements expressed on the one viewpoint may influence choices to be made on another viewpoint. How to deal with this? In recent work (Baida et al. 2002), we propose the use of a feature-solution graph (de Bruin & van Vliet 2001) to do so. Viewpoints are split-up in features and solutions, which are connected by different types of relations. Some features e.g. can have multiple solutions, or can be positively influenced by a choosing a solution. On the other hand, some solutions may also be forbidden if a particular feature is of importance, or may negatively influence a feature. These relations are also possible between viewpoints themselves. For instance, many solutions chosen on the business value requirement result in requirements on the business process viewpoint, and sometimes on the information system viewpoint. By modeling these relations explicitly, we can reason about choices for a particular feature and solution on each viewpoint.

Primary and secondary value objects. The exploration of an e-commerce idea results in one or more value models, capturing the essentials of the idea. The exchange of value objects in such models often require additional facilities which can themselves be seen as business opportunities. For instance, many e-commerce value models suppose facilities like payment services. Different payment services exist, e.g. debit or credit cards, or micropayments, for which the user (e.g. the seller and/or buyer) must pay a fee. Because these services offer a valuable experience to one or more actors, these services can also be modeled as part of a value model. Such a payment service can be seen as a secondary value object, which supports a primary value object. It is in principal possible to model primary and secondary value objects in one value model, but this leads to rather complex models. Consequently, a question for further research is how to model these primary and secondary value objects (and their exchanges) in such a way that value models are still easy to understand.

More thorough validation. We have used e-commerce idea exploration projects to develop e^3-*value* . Using an Action Research like approach, we have learned from project experiences and we extended and changed e^3-*value* accordingly. A way to improve and validate e^3-*value* is to use it in a slightly different domain. So far, we have used e^3-*value* in innovative, Internet enabled e-commerce ideas, with a focus on products and services, which can be online ordered and delivered. In the near future, we will extend and validate the e^3-*value* approach by developing innovative services for the energy market in an EC-funded EESD project called BusMod. Energy services are similar to digital products and services, in a way that ordering and influencing the way of delivery can be done using an Internet like network. In addition, BusMod will focus on the representation of *dynamic* value constellations and complex value objects (e.g. objects offered by multiple parties).

Integrated tool support. Integrated tool support is needed for drawing and checking models (e.g. for compliance with the e^3-*value* ontology), as well as to evaluate value models. At the time of writing, no integrated tool support is available. We will develop such support in the IST project OBELIX.

11.4 The future of value-based requirements engineering

The work presented in this thesis is part of an envisioned, more comprehensive, requirements engineering approach for value proposition intensive information systems. e-Commerce is an important domain for studying value-based requirements engineering. Over the past few years, many text books have seen the light on e-commerce (see e.g. Choi et al. (1997), Turban et al. (2002), Chan, Lee, Dillon & Chang (2001), Saloner & Spence (2002), and Awad (2002)). Some of these books are biased towards a business perspective, while others focus more on ICT issues. What lacks is a truly multi-disciplinary approach, which is needed to develop e-commerce information systems. The e^3-*value* methodology aims to be such a multi-disciplinary approach.

But even in their own right, many of these publications fail to address e-commerce and value-based requirements engineering adequately. In the realm of economics and organizational science, only a few authors really demonstrate new value propositions which are enabled by new technological possibilities, let alone that they present guidance how to develop such a proposition. Also, businesses themselves (e.g. existing enterprises and start-ups) have clearly shown how *not* to do e-commerce, as can seen by the large number of enterprises who have gone bankrupt. In many cases, e-commerce ideas have not been profitable at all, but more importantly, no thorough approach has been used to understand these ideas well. In this thesis we have shown that requirements engineering and conceptual modeling techniques can be used to do so. However, also the realm of information technology has difficulties in articulating e-commerce issues. For instance, information technology biased text books often fail to discuss *specific e-commerce* technology issues but present common information technology practices (at worst how to write a Java program or how to set up a web server), which are not specific for e-commerce at all. They should better explain how we can add a bit more semantics to the web (Berners-Lee & Lassila 2001, Berners-Lee & Fischetti 2001, Fensel & Musen 2001), or they should discuss how to build software agents with economic behavioral knowledge, e.g for marketplaces (see e.g. Wurman (2001) and Akkermans (2001*a*)).

One of the most important steps to be taken in value-based requirements engineering is that business oriented as well as information technology oriented stakeholders have the skills to participate in a requirements engineering process. To do so, both parties must become more knowledgable on the strategic implications of information technology. A CxO should learn to articulate and specify an e-commerce value proposition more thoroughly and must know which enabling roles technology can play in value propositions. In contrast, a programmer should understand that in the end someone must be willing to buy his/her products. As such, the e^3-*value* approach outlined in this thesis is intended to bring these stakeholders somewhat closer to each other.

Appendix A

OCL constraints

This appendix gives some e^3-*value* ontology constraints expressed using UML's OCL (*OMG Unified Modeling Language Specification, Version 1.3* 1999), (Warmer & Kleppe 1999). We use a slightly different notation to enhance readibility. To navigate through the UML model, OCL uses the role name connected to the class one wants to navigate to (the destination class). Classes and roles are separated by a '.'. We use the source role name, that is the role name connected to the class one starts with, and we append this role name with the class name one navigates to. Also, we fill in spaces (e.g. in class names) with the '-' character.

A.1 Exchange related constraints

- Value exchanges may only connect ports, which exchange the same value objects

```
context value-exchange inv:
    if self.type=#type1 or self.type=#type4 then
        self.has-in.value-port.offers-requests.value-object=
        self.has-out.value-port.offers-requests.value-object
    else
        true
    endif
```

```
context value-exchange inv:
    if self.type=#type2 or self.type=#type3
        self.has-first.value-port.offers-requests.value-object=
```

```
        self.has-second.value-port.offers-requests.value-object
    else
        true
    endif
```

- A value exchange of:
  - type 1 must connect two opposite directed ports in value interfaces of different actors

```
context value-exchange inv:
    if self.type=#type1 then
        self.has-in.value-port.
        in.value-offering.in.value-interface.assigned-to-ac.actor <>
        self.has-out.value-port.
        in.value-offering.in.value-interface.assigned-to-ac.actor
        and
        self.has-in.value-port.direction=#in
        and
        self.has-out.value-port.direction=#out
    else
        true
    endif
```

  - type 2 must connect two equally directed ports in value interfaces of different actors, where the first port is in a value interface of a composite actor, and the second port is in a value interface of another actor and the latter interface must also be in the set of value interfaces grouped by the composite actor.

```
context value-exchange inv:
    if self.type=#type2 then
        /* assume that the composite actor's port is the first port */
        let composite-vp: value-port = self.has-first.value-port
        let another-vp: value-port = self.has-second.value-port
        let composite-vi: value-interface = composite-vp.in.value-offering.
        in.value-interface
        let another-vi: value-interface = another-vp.in.value-offering.
        in.value-interface
        let composite-actor: actor = composite-vi.assigned-to-ac.actor
        let another-actor: actor = another-vi.assigned-to-ac.actor
        in
```

```
        composite-actor <> another-actor
        and
        composite-vp.direction = another-vp.direction
        and
        composite-actor.oclIsTypeOf(composite-actor)
        and
        another-actor.oclIsTypeOf(actor)
        and
        composite-actor.consists-of.value-interface→
        exists (vi:value-interface | vi=another-vi))
    else
        true
    endif
```

- type 3 must connect two equally directed ports where the first port is in the value interface of an actor, and the second port is in a value interface of a value activity which is performed by that actor

```
context value-exchange inv:
    if self.type=#type3 then
        /* assume the first port is the actor’s port*/
        let actor-vp: value-port = self.has-first.value-port
        let value-activity-vp: value-port = self.has-second.value-port
        let actor-vi: value-interface = actor-vp.in.
        value-offering.in.value-interface
        let value-activity-vi: value-interface = value-activity-vp.in.
        value-offering.in.value-interface
        in
        actor-vp.direction = value-activity.vp.direction
        and
        value-activity-vi.assigned-to-va.value-activity.
        performed-by.elementary-actor=actor-vi.assigned-to-ac.actor
    else
        true
    endif
```

- type 4 must connect two opposite directed ports in value interfaces of different value activities, which are both performed by the same actor

```
context value-exchange inv:
    if self.type=#type4 then
        let first-vp : value-port = self.has-in.value-port
```

```
        let second-vp : value-port = self.has-out.value-port
        let first-activity : value-activity = first-vp.
        in.value-offering.in.value-interface.assigned-to-va.value-activity
        let second-activity : value-activity = second-vp.
        in.value-offering.in.value-interface.assigned-to-va.value-activity
        let first-actor: actor = first-activity.performed-by.
        elementary-actor
        let second-actor: actor = second-activity.performed-by.
        elementary-actor
        in
        first-vp.direction <> second-vp.direction
        and
        first-actor = second-actor
        and
        first-activity <> second-activity
    else
        true
    endif
```

- A value exchange is uniquely identified by the ports it connects.

```
context value-exchange inv:
    value-exchange.allInstances→forAll (ve1:value exchange, ve2:value exchange |
        ve1 <> ve2 implies
            ((ve1.has-in.value-port <> ve2.has-in.value-port) or
            (ve1.has-out.value-port <> ve2.has-out.value-port))
            or
            ((ve1.has-first.value-port <> ve2.has-first.value-port) or
            (ve1.has-second.value-port <> ve2.has-second.value-port))
        )
```

A.2 Offering related constraints

- A value offering contains only equally directed value ports.

```
context value-offering inv:
    self.consists-of.value-port→forAll (vp:value-port | vp.direction=#in)
    or
    self.consists-of.value-port→forAll (vp:value-port | vp.direction=#out)
```

- A value interface contains one value offering, or contains two value offerings. In the latter case, one value offering contains only ports with direction *in*, while the other offering contains only ports with direction *out*.

```
context value-interface inv:
    self.consists-of.value-offering→size=1
    or
    (self.consists-of.value-offering→size=2 and
    self.consists-of.value-offering→exists (vo:value-offering |
        vo.consists-of.value-port→forAll (vp:value-port | vp.direction=#in))
    and
    self.consists-of.value-offering→exists (vo:value-offering |
        vo.consists-of.value-port→forAll (vp:value-port | vp.direction=#out))
    )
```

A.3 Transaction related constraints

- A transaction only contains value exchanges of equal types (type 1, 2, 3 or 4).

```
context value-transaction inv:
    self.consists-of.value-exchange→
    forAll (ve:value-exchange | ve.type=#type1) or
    self.consists-of.value-exchange→
    forAll (ve:value-exchange | ve.type=#type2) or
    self.consists-of.value-exchange→
    forAll (ve:value-exchange | ve.type=#type3) or
    self.consists-of.value-exchange→
    forAll (ve:value-exchange | ve.type=#type4)
```

- A transaction relates value interfaces via ports and value exchanges. For each value interface related to such a transaction must hold that each port of such a value interface is connected to a value exchange in that transaction. Otherwise, the semantics of value interface (exchange objects via all ports, or none at all), is not obeyed.

```
/* we assume here type1 or type4 value exchanges: connecting ports with
opposite directions */
```

```
context value-transaction inv:
    if self.consists-of.value-exchange→forAll (ve:value-exchange |
    ve.type=#type1 or ve.type=#type4) then (
        self.consists-of.value-exchange→forAll (ve:value-exchange |
            /* take the ports connected to the value exchange, */
            /* and find the value interfaces connected to these ports */
            let vi1: value-interface = ve.has-in.value-port.in.
            value-offering.in.value-interface
            let vi2: value-interface = ve.has-out.value-port.in.
            value-offering.in.value-interface in
            /* for each value port in the first value interface:*/
            vi1.consists-of.value-offering→forAll (vo:value-offering |
                vo.consists-of.value-port→forAll (vp:value-port |
                self.consists-of.value-exchange→
                exists (vp.in-connects.value-exchange)
                or
                self.consists-of.value-exchange→
                exists (vp.out-connects.value-exchange)
            )) and
            /* for each value port in the second value interface:*/
            vi2.consists-of.value-offering→forAll (vo:value-offering |
                vo.consists-of.value-port→forAll (vp:value-port |
                self.consist-of.value-exchange→
                exists (vp.in-connects.value-exchange)
                or
                self.consist-of.value-exchange→
                exists (vp.out-connects.value-exchange)
            ))
        ))
        else
            true
        endif
```

```
/* we assume here type2 or type3 value exchanges connecting ports with
equal directions*/
context value-offering inv:
    if self.consists-of.value-exchange→forAll (ve:value-exchange |
    ve.type=#type2 or ve.type=#type3) then (
        self.consists-of.value-exchange→forAll (ve:value-exchange |
            /* take the ports connected to the value exchange,*/
            /* and find the value interfaces connected to these ports*/
```

```
        let vi1: value-interface = ve.has-first.value-port.in.value-interface
        let vi2: value-interface = ve.has.second.value-port.in.value-interface in
        /* for each value port in the first value interface:*/
        vi1.consists-of.value-offering→forAll (vo:value-offering|
            vo.consists-of.value-port→forAll (vp:value-port |
            self.consist-of.value-exchange→
            exists (vp.firsts-connects.value-exchange)
            or
            self.consist-of.value-exchange→
            exists (vp.second-connects.value-exchange)
        )) and
        /* for each value port in the second value interface:*/
        vi2.consists-of.value-offering→forAll (vo:value-offering|
            vo.consists-of.value-port→forAll (vp:value-port |
            self.consist-of.value-exchange→
            exists (vp.first-connects.value-exchange)
            or
            self.consist-of.value-exchange→
            exists (vp.second-connects.value-exchange)
        ))
    ))
    else
        true
    endif
```

- A port which is related to a transaction via one of its value exchanges, must only relate via that value exchange to that transaction.

```
context value-port inv:
    self.in-connects.value-exchange→forAll (ve1 : value-exchange |
        /* the transaction ve1 is in */
        let vt : value-transaction = ve1.in.value-transaction in
        /* select all value exchanges connected to the port except ve1*/
        /* check if all value exchanges in the selected set are not part of the
        transaction in which ve1 is*/
        (self.in-connects.value-exchange→
        select (ve2:value-exchange | ve1<>ve2))→
        forAll (ve3: value exchange | vt.consists-of.value-exchange→
        exists (ve3)=false)
    )
```

```
context value-port inv:
    self.out-connects.value-exchange→forAll (ve1 : value-exchange |
        /* the transaction ve1 is in */
        let vt : value-transaction = ve1.in.value-transaction in
        /* select all value exchanges connected to the port except ve1*/
        /* check if all value exchanges in the selected set are not part of the
        transaction in which ve1 is*/
        (self.out-connects.value-exchange→
        select (ve2:value-exchange | ve1<>ve2))→
        forAll (ve3: value exchange | vt.consists-of.value-exchange→
        exists (ve3)=false)
    )
```

```
context value-port inv:
    self.first-connects.value-exchange→forAll (ve1 : value-exchange |
        /* the transaction ve1 is in */
        let vt : value-transaction = ve1.in.value-transaction in
        /* select all value exchanges connected to the port except ve1*/
        /* check if all value exchanges in the selected set are not part of the
        transaction in which ve1 is*/
        (self.first-connects.value-exchange→
        select (ve2:value-exchange | ve1<>ve2))→
        forAll (ve3: value exchange | vt.consists-of.value-exchange→
        exists (ve3)=false)
    )
```

```
context value-port inv:
    self.second-connects.value-exchange→forAll (ve1 : value-exchange |
        /* the transaction ve1 is in */
        let vt : value-transaction = ve1.in.value-transaction in
        /* select all value exchanges connected to the port except ve1*/
        /* check if all value exchanges in the selected set are not part
        of the transaction in which ve1 is*/
        (self.second-connects.value-exchange→
        select (ve2:value-exchange | ve1<>ve2))→
        forAll (ve3: value exchange | vt.consists-of.value-exchange→
        exists (ve3)=false)
    )
```

Samenvatting

Informatiesystemen worden steeds meer een onderdeel van producten en diensten die bedrijven aanbieden aan hun klanten. Dit is met name zichtbaar bij *e-commerce* toepassingen waar het Internet wordt ingezet om commerciële transacties met klanten af te handelen. De invloed van het Internet is met name merkbaar bij *digitale* producten. Dit zijn producten zoals muziek, film en software. Kenmerkend is dat digitale producten niet alleen besteld kunnen worden via het Internet, maar ook via ditzelfde medium geleverd kunnen worden. Dit maakt tal van nieuwe business ideeën mogelijk die innovatieve proposities naar klanten bevatten. Een business idee dat gebruik maakt van de mogelijkheden die het Internet biedt noemen we een *e-commerce* idee.

De recente e-commerce geschiedenis heeft duidelijk gemaakt dat het succesvol uitvoeren van nieuwe e-commerce ideeën niet eenvoudig is. Veel implementaties zijn mislukt en hebben zelfs geleid tot faillissement. Een belangrijke oorzaak voor mislukking is het gebrek aan winstgevendheid. Een ander complicerend aspect met betrekking tot de uitvoering van een e-commerce idee is dat business en technologie vraagstukken sterk met elkaar verweven zijn. Ook is een e-commerce idee in eerste instantie vaak slechts vaag geformuleerd, hetgeen leidt tot een verschillende interpretatie van het idee door de verschillende partijen die betrokken zijn bij de uitvoering.

De in dit proefschrift voorgestelde e^3-*value* methode is enerzijds bedoeld om een e-commerce idee beter te beschrijven zodat een ieder het idee op dezelfde wijze interpreteert. Anderzijds staat het begrip *economische waarde* centraal zodat een oordeel kan worden gevormd over de potentiële winstgevendheid van het idee voor een ieder die erbij betrokken is. Een kenmerk van onze methode is dat we verschillende perspectieven hanteren die het belang van verschillende groeperingen reflecteren. Zo is er een gezichtspunt voor de propositie naar klanten, een gezichtspunt voor het interorganisationele bedrijfsproces dat die propositie deels implementeert, en een gezichtspunt voor het informatiesysteem dat bij het idee behoort.

Dit laatste gezichtspunt is noodzakelijk omdat het het belang van informatietechnologie aangeeft in een e-commerce idee: de technologie die benodigd is om het idee daadwerkelijk uit te voeren moet reeds in een vroeg stadium verkend worden omdat deze technologie vaak het idee (on)mogelijk maakt. We gebruiken deze gezichtspunten om discussies over de inhoud van het e-commerce idee efficiënter te laten verlopen; discussies worden gevoerd door groeperingen die daar daadwerkelijk belang bij hebben en besluiten worden genomen door personen die daar ook het mandaat voor hebben.

De informatiekunde & informatica als wetenschap heeft veel bijgedragen aan het beter, en formeler beschrijven van visies op een werkelijkheid. Zo'n beschrijving noemen we een conceptueel model en het opstellen van zo'n model is vaak het resultaat van een requirements engineering proces: het in samenspel met betrokken partijen opstellen en valideren van een *model* dat een pakket van eisen beschrijft. De informatiekunde & informatica heeft voor bedrijfsprocessen en voor informatiesystemen het construeren van dergelijke modellen veelvuldig bestudeerd. Het beschrijven van een propositie naar klanten is echter nog niet aangepakt middels een conceptuele, requirements engineering-achtige benadering. Daarom is de focus in dit proefschrift gericht op het modelmatig beschrijven van zo'n propositie, zodat partijen op een gelijke wijze de propositie interpreteren. Een modelmatige beschrijving van een propositie noemen we een *waarde*model omdat het model aangeeft welke partijen objecten van waarde creëren, distribueren en consumeren. Een tweede toepassing van een waardemodel is het kunnen evalueren van een e-commerce idee; in onze context betekent dat beoordelen onder welke condities een idee winstgevend kan zijn.

De concepten die in een waardemodel moeten voorkomen zijn uitgedrukt in een ontologie. Een ontologie is een formele specificatie van een conceptualisatie die wordt gedeeld door een aantal partijen die belang hebben bij die conceptualisatie. De concepten in onze ontologie zijn gebaseerd op een aantal projecten die tot doel hadden een e-commerce idee te concretiseren en op literatuur op het gebied van marketing en axiologie (waardetheorie), en bevatten termen als *actor*, *object van waarde* en *waardenuitwisseling*. De ontologie is lichtgewicht in de zin dat deze slechts een beperkt aantal concepten bevat. Als we het idee dan ook nog globaal beschrijven is het mogelijk een waardemodel in korte tijd op te stellen. Dit is van belang omdat, gezien onze ervaring, het verkennen van een e-commerce idee slechts een beperkte tijdsperiode in beslag mag nemen.

Om de propositie naar klanten verder te beschrijven gebruiken we *operationele scenario's*. Deze scenario's laten zien welke objecten van waarde actoren met elkaar moeten uitwisselen, als gevolg van een klant die een bepaalde behoefte heeft. Operationele scenario's kunnen gezien worden als 'verhalen' die eenvoudig uitgelegd kunnen worden aan partijen, maar dienen ook een ander doel. Zij maken het mogelijk om op basis van het aantal verwachte klantbehoeften per tijdsperiode te redeneren over de verwachte winstgevendheid per actor.

De voorgenoemde ontologie geeft aan *wat* er in een waardemodel moet voorkomen, maar geeft niet aan *hoe* een dergelijk model kan worden geconstrueerd. Daarom geven wij een stappenplan voorzien van richtlijnen dat gebruikt kan worden om voor een specifiek e-commerce idee een waardemodel op stellen. Eén zo'n richtlijn is de 'stopcontacten' regel: als een actor aan een andere actor een object van waarde aanbiedt, verwacht de aanbiedende actor een ander object van waarde terug ter compensatie.

Een onderdeel van het opstellen van een waardemodel is deconstructie en reconstructie van een model. Een opgesteld waardemodel wordt dan eerst afgebroken in fragmenten (de deconstructie stap). De reconstructiestap bouwt nieuwe waardemodellen op, met gebruikmaking van de gevonden fragmenten. Dit is weliswaar geen recept om nieuwe e-commerce ideeën te vinden, maar wel een manier om variaties op een reeds geformuleerd idee te ontdekken. Wij hebben een aantal operatoren gedefinieerd die op algoritmische wijze aangeven hoe een waardemodel gedeconstrueerd en gereconstrueerd kan worden.

Voor een aantal innovatieve e-commerce ideeën worden mogelijke waardemodellen besproken en geëvalueerd. Evaluatie van waardemodellen richt zich op een analyse van de winstgevendheid. Harde getallen over winstgevendheid zijn nauwelijks te produceren omdat deze afhankelijk zijn van onzekere factoren, zoals het aantal klanten dat diensten en producten afneemt en de wijze van economisch waarderen van deze diensten en producten door betrokken partijen. Daarom richten we ons tijdens de evaluatie meer op mogelijke toekomstige variaties, bijvoorbeeld een variatie in het aantal verkochte producten en diensten per tijdsperiode. Zulke verwachte toekomstige variaties in het oorspronkelijke e-commerce idee noemen we *evoluationaire* scenario's.

Hoewel ons onderzoek zich hoofdzakelijk richt op het verkennen van de propositie naar de klant, onderkennen we ook dat andere perspectieven moeten worden geëxploreerd om een goed beeld van een e-commerce idee te krijgen. Twee andere perspectieven die het interorganisationele bedrijfsproces en het benodigde informatiesysteem verkennen voor een contactadvertentie e-commerce idee worden geïllustreerd, tezamen met het bijbehorende waardeperspectief. Operationale scenario's krijgen dan nog een andere rol. De verschillende perspectieven vertegenwoordigen belangen van verschillende groeperingen. Zo zal een directielid vaak uitspraken willen doen over het waardemodel, terwijl mensen die verantwoordelijk zijn voor de benodigde informatietechnologie een zware stem hebben in beslissingen die tot uitdrukking worden gebracht op het informatiesysteem perspectief. Een potentieel gevaar van een dergelijke aanpak is dat de verschillende perspectieven divergeren ten opzichte van het oorspronkelijke idee. We gebruiken de operationale scenario's, die gegrondvest zijn op een klantbehoefte, om te zorgen dat de drie perspectieven hetzelfde e-commerce idee belichten. Voor ieder perspectief wordt *hetzelfde* operationele scenario voor ieder perspectief *verschillend* geformaliseerd.

Tot slot blijkt dat computerondersteuning onontbeerlijk is voor beschrijven van de verschillende perspectieven en operationale- en evolutionaire scenario's. Er is elementaire ondersteuning aanwezig voor e^3-*value* , maar geïntegreerde ondersteuning zal worden ontwikkeld in het EC-IST project OBELIX.

Bibliography

A2B Music (1999).
URL: *http://www.a2bmusic.com/*

Akkermans, J. M., ed. (2001*a*), *IEEE Intelligent Systems - Intelligent e-Business*, Vol. 16, IEEE Computer Society Publications, Los Alamitos, CA.

Akkermans, J. M. (2001*b*), 'Intelligent e-Business: From technology to value', *IEEE Intelligent Systems - Intelligent e-Business* **16**(4), 8–10.

Amyot, D. & Mussbacher, G. (2000), On the extension of UML with use case maps concepts, *in* A. Evans, S. Kent & B. Selic, eds, 'UML 2000 - The Unified Modeling Language. Advancing the Standard.', Vol. 1939 of *LNCS*, Springer Verlag, Berlin, D, pp. 16–31.

Antón, A. I. & Potts, C. (1998), 'A representational framework for scenarios of system use', *Requirements Engineering* **3**(3/4), 219–241.

Avison, D., Lau, F., Myers, M. & Nielsen, P. A. (1999), 'Action research', *Communications of the ACM* **42**(1), 94–97.

Awad, E. M. (2002), *Electronic Commerce: From Vision to Fulfillment*, Prentice Hall, Upper Saddle River, NJ.

Baida, Z., de Bruin, H. & Gordijn, J. (2002), 'Business cases assessment: From business value to system feasibility', *submitted* .

Bass, L., Clements, P. & Kazman, R. (1997), *Software Architectures in Practice*, Addison-Wesley, Reading, MA.

Berners-Lee, T. & Fischetti, M. (2001), *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*, Harper, San Francisco, CA.

Berners-Lee, T. & Lassila, O. (2001), 'The semantic web', *Scientific American* (May), 29–37.

Blaha, M. & Premerlani, W. (1998), *Object-Oriented Modeling and Design for Database Applications*, Prentice Hall, Upper Saddle River, NJ.

Blaze, M., Feigenbaum, J. & Lacy, J. (1996), Decentralized trust management, *in* 'Proceedings of the IEEE Symposium on Security and Privacy', pp. 164–173.

Bollier, D. (1996), *The Future of e-Commerce*, The Aspen Institute, Washington, DC.
URL: *http://www.aspeninstitute.org/*

Borst, W. N., Akkermans, J. M. & Top, J. L. (1997), 'Engineering ontologies', *International Journal of Human-Computer Studies* **46**, 365–406.

Borst, P. (1997), Construction of Engineering Ontologies for Knowledge Sharing and Reuse, PhD thesis, Universiteit Twente, Enschede, NL.

Bowie, D. (2000), *Fame*.
URL: *http://www.davidbowie.com/fame/*

de Bruin, H. & van Vliet, J. C. (2001), Scenario based generation and evaluation of software architectures, *in* J. Bosch, ed., 'Proceedings of the Third International Conference on Generative and Component-Based Software Engineering (GCSE 2001)', Vol. 2186 of *LNCS*, Springer Verlag, Berlin, D, pp. 128–139.

Buhr, R. J. A. (1998), 'Use case maps as architectural entities for complex systems', *IEEE Transactions on Software Engineering* **24**(12), 1131–1155.

Buhr, R. J. A. & Casselman, R. S. (1999), *Use Case Maps for Object-Oriented Systems*, Prentice Hall, Englewood Cliffs, NJ.

BusMod consortium (2001), 'BusMod project NNE5-2001-00256: Business models in a world characterised by distributed generation: Annex I - description of work'.

Carroll, J. M. & Rosson, M. B. (1992), 'Getting around the task-artifact cycle: How to make claims and design by scenario', *ACM Transactions on Information Systems* **10**(2), 181–212.

Chan, H., Lee, R., Dillon, T. & Chang, E. (2001), *E-Commerce: Fundamentals and Principles*, John Wiley & Sons Inc., Chichester, UK.

Chandrasekaran, B. (1990), 'Design problem solving: A task analysis', *AI Magazine* **11**(4), 59–71.

Checkland, P. (1991), From framework through experience to learning: The essential nature of action research, *in* H.-E. Nissen, H. K. Klein & R. Hirschheim, eds, 'Information Systems Research: Contemporary Approaches & Emergent Traditions', North-Holland, Amsterdam, NL, pp. 397–403.

Checkland, P. & Holwell, S. (1995), *Business Processes - Modelling and Analysis for Re-engineering and Improvement*, John Wiley & Sons, Chichester, UK.

Choi, S.-Y., Stahl, D. O. & Whinston, A. B. (1997), *The Economics of Doing Business in the Electronic Marketplace*, MACMillan Technical Publishing, Indianapolis, IN.

Christensen, E., Curbera, F., Meredith, G. & Weerawarana, S. (2001), *Web Services Description Language WSDL 1.1)*, Microsoft and IBM Research.
URL: *http://www.w3.org/TR/wsdl/*

Clarke, I., Sandberg, O., Wiley, B. & Hong, T. W. (2000), Freenet: A distributed anonymous information storage and retrieval system, *in* 'Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability', Vol. 2009 of *LNCS*, Springer Verlag, Berlin, D, pp. 46–66.

Craver, S., Yeo, B.-L. & Yeung, M. (1998), 'Technical trials and legal tribulations', *Communications of the ACM* **41**(7), 45–54.

Davenport, T. H. (1993), *Process Innovation : Reengineering Work Through Information Technology*, Harvard Business School Press, Boston, MA.

Davidow, W. H. & Malone, M. S. (1992), *The Virtual Corporation - Structuring and Revitalizing the Corporation for the 21st Century*, HarperCollings, New York, NY.

Drury, C. (1998), *Management and Cost Accounting, third edition*, Chapman and Hall, London, UK.

e-Music (2001).
URL: *http://www.emusic.com/*

Evans, P. & Wurster, T. S. (2000), *Blown to Bits - How the New Economics of Information Transforms Strategy*, Harvard Business School Press, Boston, MA.

Fensel, D. & Musen, M., eds (2001), *IEEE Intelligent Systems - Cooking up the Semantic Web*, Vol. 16, IEEE Computer Society Publications, Los Alamitos, CA.

Finkelstein, A., Gabbay, D., Hunter, A. & Nuseibeh, B. (1994), ‘Inconsistency handling in multi-perspective specifications’, *IEEE Transactions on Software Engineering* **20**(8), 569–578.

Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L. & Goedicke, M. (1992), ‘Viewpoints: A framework for integrating multiple perspectives in system development’, *International Journal of Software Engineering and Knowledge Engineering* **2**(1), 31–58.

Fowler, M. & Scott, K. (1995), *Design patterns: Elements of Reusable Object-oriented Software*, Addison Wesley Longmann, Inc., Reading, MA.

Fowler, M. (1997), *Analysis Patterns*, Addison Wesley Longmann, Inc., Reading, MA.

Fox, M. S. & Gruninger, M. (1998), ‘Enterprise modelling’, *AI Magazine* **19**(3), 109–121.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1997), *UML Distilled - Applying the Standard Object Modelling Language*, Addison Wesley Longmann, Inc., Reading, MA.

Gervasi, V. & Nuseibeh, B. (2000), Lightweight validation of natural language requirements, *in* ‘Proceedings of 4th IEEE International Conference on Requirements Engineering (ICRE’2000)’, IEEE CS Press, Los Alamitos, CA, pp. 140–147.

Glaser, B. G. & Strauss, A. L. (1967), *Discovery of Grounded Theory : Strategies for Qualitative Research*, Aldine Publishing Co, Chicago, IL.

Gnutella (2001).
URL: *http://www.gnutella.co.uk/*

Gordijn, J., Akkermans, J. M. & van Vliet, J. C. (2000*a*), Value based requirements creation for electronic commerce applications, *in* R. H. Sprague Jr., ed., ‘Proceedings of the 33rd Hawaii International Conference On System Sciences (HICSS-33)’, IEEE CS Press, Los Alamitos, CA. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J., Akkermans, J. M. & van Vliet, J. C. (2000*b*), What's in an electronic business model, *in* R. Dieng & O. Corb, eds, 'Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference (EKAW 2000)', Vol. 1937 of *LNAI*, Springer Verlag, Berlin, D, pp. 257–273. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J., Akkermans, J. M. & van Vliet, J. C. (2000*c*), Business modelling is not process modelling, *in* S. W. Liddle & H. C. Mayr, eds, 'Conceptual Modeling for E-Business and the Web', Vol. 1921 of *LNCS*, Springer Verlag, Berlin, D, pp. 40–51. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J., Akkermans, J. M., van Vliet, J. C. & Paalvast, E. R. M. R. (2000*d*), Selling bits: A matter of creating consumer value, *in* K. Bauknecht, S. K. Madria & G. Pernul, eds, 'First International Conference on Electronic Commerce and Web Technologies (EC-Web 2000)', Vol. 1875 of *LNCS*, Springer Verlag, Berlin, D, pp. 48–62. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J., de Bruin, H. & Akkermans, J. M. (1999), Integral design of E-Commerce systems: Aligning the business with software architecture through scenarios, *in* H. de Bruin, ed., 'ICT-Architecture in the BeNeLux', Vrije Universiteit, Amsterdam, NL. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J., de Bruin, H. & Akkermans, J. M. (2001), Scenario methods for viewpoint integration in e-business requirements engineering, *in* R. H. Sprague Jr., ed., 'Proceedings of the 34rd Hawaii International Conference On System Sciences (HICSS-34)', IEEE CS Press, Los Alamitos, CA. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J. & van Vliet, J. C. (1999), On the interaction between business models and software architecture in electronic commerce, *in* M. Lemoine, ed., 'Addendum to the proceedings of the 7th European Software Engineering Conference/Foundations of Software Engineering, Toulouse'. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J. & Akkermans, J. M. (2001*a*), 'Designing and evaluating e-Business models', *IEEE Intelligent Systems - Intelligent e-Business* **16**(4), 11–17.

Gordijn, J. & Akkermans, J. M. (2001*b*), e^3*-value* : A conceptual value modeling approach for e-business development, *in* J. Gordijn & J. M. Akkermans, eds, 'Proceedings of K-CAP 2001, First International Conference on Knowledge Capture, WS 2, Workshop Knowledge in e-Business', pp. 29–36. Also available from http://www.cs.vu.nl/˜gordijn/.

Gordijn, J. & Akkermans, J. M. (2001*c*), Ontology-based operators for e-Business model de- and reconstruction, *in* Y. Gil, M. Musen & J. Shavlik, eds, 'Proceedings of the First International Conference on Knowledge Capture', ACM-Press, New York, NY, pp. 60–67. Also available from http://www.cs.vu.nl/˜gordijn/.

Gruber, T. R. (1994), Towards principles for the design of ontologies used for knowledge sharing, *in* N. Guarino & R. Poli, eds, 'Formal Ontology in Conceptual Analysis and Knowledge Representation', Amsterdam, NL.

Hartman, A., Sifonis, J. & Kador, J. (2000), *Net Ready - Strategies for Success in the E-conomy*, McGraw-Hill, New York, NY.

van Hee, K. M. (1994), *Informations Systems Engineering - A formal approach*, Cambridge University Press, Cambridge, UK.

van der Heijden, K. (1996), *Scenarios: The Arts of Strategic Conversation*, John Wiley & Sons Inc., New York, NY.

Heskett, J. L., Sasser, W. E. & Schlesinger Jr., L. A. (1997), *The Service Profit Chain*, The Free Press, New York, NY.

Holbrook, M. B. (1999), *Consumer Value: A Framework for Analysis and Research*, Routledge, New York, NY.

Horngren, C. T. & Foster, G. (1987), *Cost Accounting: A Managerial Emphasis, sixth edition*, Prentice-Hall, Englewood Cliffs, NJ.

$IDEF_0$ *Method Report* (1981).
URL: *http://www.idef.com/Complete_Reports/idef0/*

Intertrust (2001).
URL: *http://www.intertrust.com/*

Jackson, D. & Wing, J. (1996), 'Formal methods light: Lightweight formal methods', *IEEE Computer* **29**(4), 21–22.

Jasper, R. & Uschold, M. (1999), A framework for understanding and classifying ontology applications, *in* B. Gaines, R. Cremer & M. Musen, eds, 'Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management (KAW'99)', Vol. I, University of Calgary, SRDG Publications, Calgary, CA, pp. 4–9–1–4–9–20.

Kotler, P. (1988), *Marketing Management: Analysis, Planning, Implementation and Control*, Prentice Hall, Englewood Cliffs, NJ.

Kruchten, P. B. (1995), 'The 4+1 view model of architecture', *IEEE software* **12**(6), 42–50.

Kuhn, T. S. (1970), *The Structure of Scientific Revolutions*, The University of Chicago Press, Chicago, IL.

Lassing, N. (2002), Architecture-level Modifiability Analysis, PhD thesis, Vrije Universiteit, Amsterdam, NL.

Leveson, N. G. (1996), 'Software safety, why, what, and how', *Computing Surveys* **18**(2), 125–163.

Leymann, F. (2001), *Web Services Flow Language (WSFL 1.0)*.
URL: *http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf*

Liquid Audio (2001).
URL: *http://www.liquidaudio.com/*

Loucopoulos, P. & Karakostas, V. (1995), *System Requirements Engineering*, McGraw-Hill, Berkshire, UK.

Memon, N. & Wong, P. W. (1998), 'Protecting digital media content', *Communications of the ACM* **41**(7), 35–43.

Menezes, A. J., van Oorschot, P. C. & Vanstone, S. A. (1996), *Handbook of Applied Cryptography*, CRC Press Inc, Boca Raton, FL.

Meyer, B. (1985), 'On formalism in specifications', *IEEE Software* **2**(1), 6–26.

Microsoft Windows Digital Rights Management (DRM) (2001).
URL: *http://www.microsoft.com/windows/windowsmedia/drm.asp*

Minas, M. & Hoffmann, B. (2001), 'Specifying and implementing visual process modeling languages with DiaGen', *Electronic Notes in Theoretical Computer Science* **44**(4).

Morpheus (2001).
URL: *http://musiccity.streamcastnetworks.com/*

Motschnig-Pitrig, R., Nissen, H. W. & Jarke, M. (1997), View-directed requirements engineering: A framework and metamodel, *in* 'Proceedings of the 9th International Conference on Software Engineering and Knowledge Engineering (SEKE'97)'. Also CREWS Report 97-11.

Motta, E., Stutt, A., Zadrahal, Z., O'Hara, K. & Shadbolt, N. (1996), 'Solving VT in VITAL:A study in model construction and reuse', *Journal of Human-Computer Studies* **44**(3/4), 333–372.

Mylopoulos, J. (1992), Conceptual modeling and telos, *in* 'Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development', Wiley, New York, NY, pp. 49–68.

Normann, R. & Ramírez, R. (1993), 'From value chain to value constellation: Designing interactive strategy', *Harvard Business Review* (july-august), 65–77.

Normann, R. & Ramírez, R. (1994), *Designing Interactive Strategy - From Value Chain to Value Constellation*, John Wiley & Sons Inc., Chichester, UK.

Nunes, N. J. & Cunha, J. F. (2000), 'Wisdom: A software engineering method for small software development companies', *IEEE Software* **17**(5), 113–119.

Obelix consortium (2001), 'Obelix project IST-2001-33144: Ontology-based ELectronic integration of compleX products and value chains: Annex I - description of work'.

OMG Unified Modeling Language Specification, Version 1.3 (1999).
URL: *http://www.rational.com/media/uml/post.pdf*

Oram, A. (2001), *Peer-to-Peer Harnessing the Power of Disruptive Technologies*, O'Reilly, Sebastopol, CA.

Ould, M. A. (1995), *Business Processes - Modelling and Analysis for Re-engineering and Improvement*, John Wiley & Sons, Chichester, UK.

PCM Bezuinigt op Internet (2001).
URL: *http://www.pcm.nl/persbericht1.phtml?id=999867741*

Porter, M. E. (1985), *Competitive Advantage - Creating and Sustaining Superior Performance*, Free Press, New York, NY.

Porter, M. E. (2001), 'Strategy and the Internet', *Harvard Business Review* (march), 63–78.

Porter, M. E. & Millar, V. E. (1985), 'How information gives you competitive advantage', *Harvard Business Review* (july-august), 149–160.

Quine, W. V. O. (1961), *From a Logical Point of view, Nine Logico-philosophical Essays*, Harvard University Press, Cambridge, MA.

Rappa, M. (2000), *Managing the Digital Enterprise*.
URL: *http://ecommerce.ncsu.edu/business_models.html*

Ringland, G. (1998), *Scenario Planning: Managing for the Future*, John Wiley & Sons Inc., New York, NY.

Rogers, E. M. (1995), *Diffusion of Innovations*, Free Press, New York, NY.

Rolland, C., Achour, C. B., Cauvet, C., Ralyte, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E. & Heymans, P. (1998), 'A proposal for a scenario classification framework', *Requirements Engineering* **3**(1), 23–47.

Rumbaugh, J., Jacobson, I. & Booch, G. (1999), *The Unified Modelling Language Reference Manual*, Addison Wesley Longmann, Inc., Reading, MA.

Saloner, G. & Spence, A. M. (2002), *Creating and Capturing Value: Perspectives and Cases on Electronic Commerce*, John Wiley & Sons Ltd., Chichester, UK.

Schreiber, A. T., Akkermans, J. M., Anjewierden, A. A., de Hoog, R., Shadbolt, N., van der Velde, W. & Wielinga, B. J. (2000), *Knowledge Engineering and Management*, The MIT Press, Cambridge, MA.

Schuitemaker, K. (2000), EVORT: A tool for representing and validating models based on the e^3-*value* ontology, Master's thesis, Vrije Universiteit, Amsterdam, NL.

Secure Digital Music Initiative (2001).
URL: *http://www.sdmi.org/*

Shama, A. (2001), 'Dot-coms' coma', *The Journal of Systems and Software* **56**(1), 101–104.

Shapiro, C. & Varian, H. R. (1999), *Information Rules*, Harvard Business School Press, Boston, MA.

SICStus Prolog version 3.8 manual (1999).

Slywotzky, A. J. (1996), *Value Migration - How to Think Several Moves Ahead of the Competition.*, Harvard Business School Press, Boston, MA.

Sommerville, I. & Sawyer, P. (1997), 'Viewpoints: Principles, problems and a practical approach to requirements engineering', *Annals of Software Engineering* **3**, 101–130.

Strassmann, P. (1997), *The Squandered Computer: Evaluating the Business Alignment of Information Technologies*, The Information Economics Press, New Canaan, CT.

Tan, Y.-H. (2002), ‘Formal aspects of a generic model of trust for electronic commerce’, *Journal of Decision Support Systems and Electronic Commerce* (to appear).

Tapscott, D., Ticoll, D. & Lowy, A. (2000), *Digital Capital - Harnessing the Power of Business Webs*, Nicholas Brealy Publishing, London, UK.

Timmers, P. (1999), *Electronic Commerce: Strategies and Models for Business-to-Business Trading*, John Wiley & Sons Ltd., Chichester, UK.

Turban, E., Lee, J., King, D. & Chung, H. M. (2002), *Electronic Commerce - A Managerial Perspective*, Prentice Hall, Englewood Cliffs, NJ.

Uschold, M., King, M., Moralee, S. & Zorgios, Y. (1998), ‘The enterprise ontology’, *The Knowledge Engineering Review* **13**(1), 31–89.

Warmer, J. & Kleppe, A. (1999), *The Object Constraint Language*, Addison-Wesley Longman, Inc., Reading, MA.

Wiegers, K. E. (1999), *Software Requirements*, Microsoft Press, Redmond, WA.

Wurman, P. R., ed. (2001), *IEEE Internet Computing - Virtual Marketplaces*, IEEE Computer Society Publications, Los Alamitos, CA.

Yu, E. S. K. & Mylopoulos, J. (1998), Why goal-oriented requirements engineering, *in* E. Dubois, A. L. Opdahl & K. Pohl, eds, ‘Proceedings of the 4th International Workshop on Requirements Engineering: Foundation for Software Quality (RESFQ 1998)’, Presses Universitaires de Namur, Namur, B.

Zeithaml, V. A. (1988), ‘Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence’, *Journal of Marketing* (52), 2–22.

Subject index

Author index

SIKS Disseration Series

1998

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W. Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification Using Decision Trees and Neural Nets

1999-3 Don Beal (Queen Mary and Westfield College)
The Nature of Minimax Search

1999-4 Jacques Penders (KPN Research)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek Wijngaards (VU)
Re-design of Compositional Systems

1999-7 David Spelt (UT)
Verification Support for Object Database Design

1999-8 Jacques Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation

2000

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien Metselaar (UvA)
Sociaal-Organisatorische Gevolgen van Kennistechnologie; een Procesbenadering en Actorperspectief

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-Theoretic Planning of Clinical Patient Management

2000-8 Veerle Coupé (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

2001

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as Problem Solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-Based User Interface Design

2001-7 Bastiaan Schönhage (VU)
DIVA: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics

2001-9 Pieter Jan ’t Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a Multi-Agent Modeling and Simulation Language for Work Practice Analysis and Design

2001-11 Tom van Engers (VU)
Knowledge Management: The Role of Mental Models in Business Systems Design

2002

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace: Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

Curriculum Vitae

Jaap Gordijn was born on January 29th, 1965 in Soest, The Netherlands. He studied Technical Computer Science (Bachelor) and Business Information Science (Master). Thereafter, he started to work in the field of Applied Information Technology research at the largest Dutch applied research organization. Five years later, his professional activities shifted towards management consultancy at Bakkenist Management Consultants, with a focus on Internet commerce and strategic ICT issues. At the same time, he was responsible for some Business Information Science courses at the Vrije Universiteit as well as research in the field of e-commerce. Again four years later, he continued his consultancy work, but now at Cisco's Internet Business Solution Group, one of the leading companies in utilizing the Internet commercially. Recently, he joined the Vrije Universiteit fulltime as an assistant professor on e-business.