

Jaap Gordijn · J.M. Akkermans

Value-based requirements engineering: exploring innovative e-commerce ideas

Published online: 15 May 2003
© Springer-Verlag London Limited 2003

Abstract Innovative e-commerce ideas are characterised by commercial products yet unknown to the market, enabled by information technology such as the Internet and technologies on top of it. *How* to develop such products is hardly known. We propose an interdisciplinary approach, *e³-value*, to explore an innovative e-commerce idea with the aim of understanding such an idea thoroughly and evaluating it for potential profitability. Our methodology exploits a requirements engineering way of working, but employs concepts and terminology from business science, marketing and axiology. It shows how to model business requirements and improve business–IT alignment, in sophisticated multi-actor value constellations that are common in electronic commerce. In addition to the *e³-value* approach methodology, we also present the action research-based development of our methodology, by using one of the longitudinal projects we carried out in the field of online news article provisioning.

Keywords Action research · Conceptual modelling · E-commerce · Economic value

1 Introduction

Over the past few years, many innovative e-commerce ideas have been considered. Such ideas are innovative, because they present new economic value propositions yet unknown to the market. A value proposition is

something offered by a party for consideration or acceptance by another party.

Recently, it became clear that many of these e-commerce ideas have not been successful [1], mainly caused by the lack of a sound and clear value proposition. A *sound* value proposition allows for each entity involved to make a profit or to increase its economic utility. *Clarity* is especially important for innovative products because customers hesitate to adopt new products if the added value is not obvious or the product is perceived to be too complex [2].

In contrast to the recent decline of many e-commerce initiatives, we believe that potential successful innovative e-commerce ideas still exist. Some industries are even *forced* to find new e-commerce-enabled value propositions. As an example, the music industry is facing a decrease in revenue because of the Internet-enabled piracy scene. A potential answer could be the use of the same Internet technology to offer consumers new products such as instant listening to a track of their choice, not only by ordering but also delivering the track via the Internet.

In the recent past, we carried out a number of innovative e-commerce projects. Such projects are, first, about ‘inventing’ and exploring an information technology (IT)-intensive product, which is of potential interest for customers. Once such a proposition is well understood, it should be put into practice, which comprises design and implementation of business processes and IT to deliver the proposition to customers.

While doing such projects, we encountered two problems ([3], chapter 2). A first problem is an *explosion* of the e-commerce ‘design space’. Many, mutual influencing design issues have to be decided on, ranging from strategic and marketing issues to technological issues. Whereas more traditional IT-intensive projects take place in a known business context, such a context is lacking for innovative e-commerce projects. Rather, an IT-enabled value proposition has to be invented that is commercially viable, which may significantly change the

J. Gordijn (✉) · J.M. Akkermans
Vuture.net, Centre for e-Business Research, Vrije Universiteit,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
E-mail: gordijn@cs.vu.nl
Tel.: +31-20-4447769
Fax: +31-20-4447769

J. Akkermans
AKMC Knowledge Management, Klareweid 19,
1831 BV Koedijk, The Netherlands

way a company does business. Such a proposition can be seen as an interdisciplinary design problem with interactions between business and technology issues, apart from intra-business and intra-technology trade-offs themselves.

Second, innovative e-commerce ideas tend to be formulated *vaguely* initially. Such an idea is a statement about an innovative value proposition utilising new technology, but it often lacks a precise description. As a result, many innovative e-commerce ideas are somewhat unfocused and inaccurate. This makes it difficult to put the idea into operation, and to develop supporting IT. Clearly, a gap exists with respect to the application of conventional requirements engineering methods, which suppose a sound and accurate understanding of a company's way of doing business.

Consequently, what is needed is a first exploration of an innovative e-commerce idea to find a direction in the numerous design options and to articulate the idea well. The result can be a starting point for a requirements engineering track to elicit, analyse and validate requirements for IT supporting and implementing the e-commerce idea.

This paper presents a methodology for exploring such an innovative e-commerce idea. Our e^3 -value approach 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 modelling* techniques, borrowed from the information systems community. This makes our research truly interdisciplinary. The motivation to use a semi-formal, conceptual approach for exploring an e-commerce idea is threefold. First, modelling such an idea explicitly contributes to a common understanding of the idea by all stakeholders involved, which is an often experienced need in e-commerce exploration tracks. Second, a semi-formal model of the e-commerce idea allows for a more rigorous assessment of potential business profitability of the idea. Third, a semi-formal model of the e-commerce idea provides the necessary bridge between a qualitatively expressed e-commerce idea and the stage where conventional requirements engineering methods can be applied in order to develop and roll out the needed supporting information systems.

This paper is structured as follows. Section 2 provides a rationale for a requirements engineering compatible approach to e-commerce idea exploration. One of the key points of our approach is that we model the value proposition. Constructs for doing so, as well as methodological issues, are discussed in Sects 3 and 4. The development of e^3 -value is a result of applying and improving the methodology in practice. Section 5 reports on such an e-commerce idea exploration project. We report on related work in Sect. 6. Section 7 summarises the key points of our approach and future research.

2 Value-based requirements engineering

2.1 Innovative and IT-intensive products

Increasingly, IT plays a dominant role in commercial products. Whereas traditionally IT supports business processes of enterprises and is seen as an *expense* only, IT-intensive commercial products are supposed to generate *revenues* for the offering enterprise.

Many of such products became possible only recently, due to the large-scale adoption of the Internet. For example, the digital product-sector is heavily affected by Internet technology adoption: music, movies and software can now be sold *and* delivered using Internet-based technology to a wide audience. Consequently, many of the Internet-enabled products are in an early adoption phase, in contrast to the adoption of the Internet itself, and are thereby innovative.

A central question is *how* to actually develop such innovative and IT-intensive commercial products. Because these products consist of computer programs and associated content (e.g. an online music shop consists of web-storefront and a database filled with music tracks), a first step could be an elicitation, analysis and validation of system requirements. It is our belief, however, that this is *not* a good starting point: to be able to discuss system requirements it is important *first* to thoroughly understand the product, embodied in software, from a *commercial* perspective. Traditional requirements engineering techniques are falling short here, because they lack the notion of economically valuable products. Also, pure business-oriented approaches are not suitable since they are not sufficiently precise to enable development of IT. Additionally, many business approaches are more usable for global visioning but not for implementing a specific business development track.

2.2 Pitfalls in developing e-commerce cases

To find a more suitable approach, we have identified a number of pitfalls in developing IT-intensive commercial products. These pitfalls are based on our extensive e-commerce consultancy experience in the field of the music industry, online news publishing, energy, banking, and insurance, and are discussed below.

2.2.1 *An innovative, sustainable, value proposition is hardly understood*

A difficulty in developing *innovative* products is that in advance the nature and consequences of the innovation can be at most vaguely articulated by stakeholders. Most e-commerce value propositions are 'invented' rather than elicited. This is caused by the novelty of *widely adopted* technology such as the Internet; hardly anyone really understands how this technological infrastructure

can be utilised in an economically sustainable way. This is clearly demonstrated by the industry itself: many e-commerce companies have gone bankrupt in the recent past [1], because they had no sustainable business idea. Consequently, many e-commerce projects should start with an exploration of a value proposition to increase confidence in its economic sustainability.

2.2.2 The value proposition is stated informally

E-commerce idea exploration tracks often yield IT-intensive value propositions which are stated informally, typically by using natural language only. Expressing value propositions this way has a number of drawbacks. First, there is the risk of a lack of common understanding of the new value proposition. This risk is high for e-commerce tracks, because such tracks involve a wide range of stakeholders. Not only do these stakeholders vary in focus (e.g. on the value proposition, supporting business process and needed IT), but often they also represent different enterprises. A second drawback of informally stated value propositions is that it is difficult to analyse and to evaluate such propositions, e.g. with respect to potential profitability for all parties involved. Finally, IT-intensive value propositions rely heavily on software. To create such software, it is commonly understood that engineers should not only know the software requirements themselves, but should also understand the business itself. An informal textual and often vague description of a value proposition leaves room for interpretation by engineers, which results in the undesirable situation that, in actual fact, software engineers take commercial business decisions themselves.

2.3 Highlights of our approach

Requirements engineering, and its extension that we call *value-based* requirements engineering, is an approach that can be of help in exploring an IT value proposition more thoroughly. Value-based requirements engineering stands for an approach that takes into account the *economic value* perspective when developing IT-intensive products through an iterative and co-operative process of analysing a business case, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained. This is an extended definition of the notion of *requirements engineering* itself; see [4], where it was defined as the process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.

Our approach, called *e³-value*, is intended for the very first phase of value-based requirements engineering. In this stage, an initial e-commerce idea should be better

understood and articulated. The aim is to find the right directions: one or more e-commerce ideas that seem to be attractive from the commercial perspective. Based on the experiences discussed previously, the *e³-value* value-based requirements engineering approach has the following characteristics.

2.3.1 A lightweight approach

E-commerce tracks are characterised by short development times. A typical timeframe is 3–6 months: from idea to a first implementation. Only a portion of this timeframe is available for exploration of e-commerce ideas. Moreover, in our experience in practice, exploration of such ideas is performed by a small number of persons (typically five to ten people). So, within a certain timeframe, only limited manpower is available. Consequently, the first phase of e-commerce requirements engineering should be a *lightweight* approach.

2.3.2 A multi-viewpoint approach

It is widely accepted that the exploration of requirements can be very complex, amongst others caused by a wide range of perspectives taken by various stakeholders. These perspectives are grounded in differences in skills, responsibilities, knowledge and expertise of stakeholders [5]. This holds even more for the development of innovative e-commerce information systems, where besides stakeholders with a technical or traditional business background, also value proposition oriented stakeholders like marketers and Chief x Officers (where x = Executive, Financial, Operational, Information, Technical) are involved. It is our experience that during innovative e-commerce projects CxO-like stakeholders even play a dominant role, because such projects create new revenue streams for an enterprise and consequently may become a significant factor in the overall profitability of a company.

The development of an IT-intensive value proposition requires the involvement of, amongst others, strategic decision makers, marketers, persons responsible for business process development, and information system architects, each with their own perspective. Requirement engineers deal with such a unfocused group stakeholders by developing multiple viewpoints. Viewpoints 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. This self-containment of viewpoints is also acknowledged in [5]. 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.

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 and organise stakeholder discussions, we use the various kinds of stakeholders as an important driver for viewpoint identification. We distinguish three stakeholder-type related viewpoints, which are grounded on viewpoint identification assumptions and criteria, which in turn are based on specific e-commerce consultancy experiences (see [3], chapter 2). In short, we assume that the specific types of viewpoints should be identified in advance rather than that identification would be part of the process itself, and we assume that we only develop a limited number of viewpoints. The latter is also stated by Sommerville and Sawyer [6], who suggest, especially in the early phases of requirements engineering, limiting the number of views to be developed. In the case of e-commerce idea exploration, this is really necessary given the short timeframes that are available for development. Additionally, (1) viewpoints should contribute from a content point of view to the assessment of economic feasibility of an e-commerce idea (to address a major concern, potential profitability), (2) viewpoints should be based on a similar focus of a group of stakeholders (to avoid time-consuming discussions between stakeholder groups with unrelated foci), and (3) a viewpoint's focus should have a minimum overlap with foci of other viewpoints (to allow decision making by stakeholder groups without consulting other stakeholder groups too much).

Table 1 presents a limited number of predefined viewpoints, and shows the name of each viewpoint, its focus, representation methods, the viewpoint holders, and the viewpoint engineers. A viewpoint *holder* is someone with a direct stake in the viewpoint, while a viewpoint *engineer* is someone facilitating the requirements engineering process [7].

2.3.2.1 The value viewpoint The top-level viewpoint of our electronic commerce framework concerns the value viewpoint. The value viewpoint focus is the (new) method of economic value creation, distribution and consumption. For viewpoint representation we employ e^3 -value models, which are explained in this paper. Viewpoint holders are CxOs 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 objects between actors.

2.3.2.2 The business process viewpoint The business process viewpoint, the middle level in Table 1, focuses on business processes, which are needed to put into practice a new value proposition, and focuses on ownership of these processes, to be able to contribute operational and capital expenses to the performing actor. To represent a business process view, a number of techniques are suitable, for instance the UML activity diagrams with swimlanes to represent actors [8, 9], interaction diagrams and sequence diagrams, high-level Petri nets [10], or role-based process-modelling techniques [11]. Also, business process (re)design approaches (see e.g. [12]) 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 processes themselves, so that stakeholders see that indeed processes can be developed which put into operation the requirements expressed on the value viewpoint.

2.3.2.3 The information system viewpoint The information system viewpoint, the bottom of Fig. 1, focuses on constituting components of an information system to be developed at a coarse granularity. Techniques are available to represent this viewpoint, such as the UML. Viewpoint holders are stakeholders responsible for development and exploitation of IT, 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.

2.3.3 A graphical, conceptual modelling approach

A *conceptual modelling* approach comprises the activity of formally defining aspects of the physical and social

Table 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
<i>Value</i> viewpoint	CxOs, marketers, consumer groups	Business developer	<i>Economic value</i> object creation, distribution and consumption	e^3 -value and UCM scenarios
<i>Process</i> viewpoint	Operational management	Business process (re)designer	<i>Process</i> ownership and flow, resources needed	UML activity, sequence, interaction diagrams, Petri nets
<i>Information system</i> viewpoint	IT department	System architect	<i>System component</i> ownership	Ownership diagrams

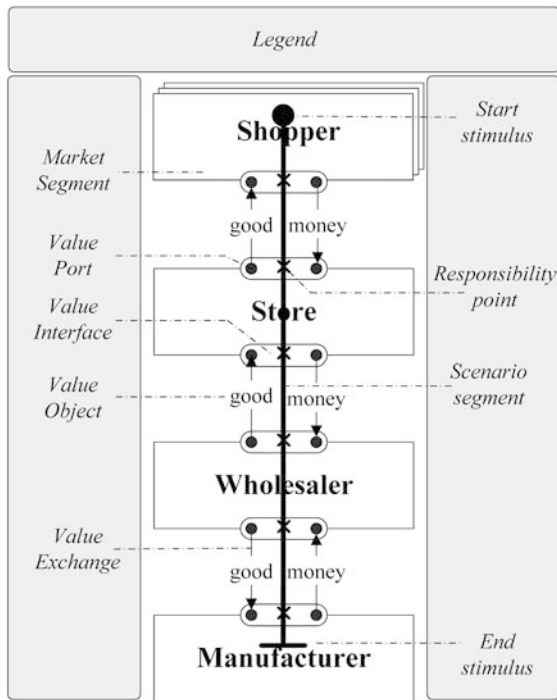


Fig. 1 An example value model, showing that a shopper receives a good, and pays money in return. The shop obtains goods from a store, who buys them from a wholesaler. The wholesaler obtains goods from a manufacturer

world around us for the purpose of understanding and communication [13]. *Formal* in this context means the abstraction, structure, and representation of knowledge in a way that makes it possible to reason about this knowledge [4]. The activity of modelling is well known and accepted in the requirements engineering 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), over-specification, contradictions, ambiguity, forward references, and wishful thinking [14].

Our experience is that a conceptual modelling 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 modelling 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 conceptual models, specifically for the value viewpoint. 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. In doing so, we use a semi-formal conceptual approach rather than a strictly logical approach because many stakeholders involved in this phase of idea exploration do not understand very formal models well (this is an understatement!).

To allow for *easy communication* with stakeholders, we opt for a lightweight approach, 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. [15]).

2.3.4 A scenario-based approach

Antón and Potts [16] distinguish (1) *operational* scenarios, and (2) *evolutionary* scenarios. By describing system behaviour, 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 in a broad sense; we see a network of actors exchanging things of value with each other as a system also. The e^3 -value methodology utilises both types of scenarios.

2.3.4.1 Operational scenarios A first use of operational scenarios is to explain and capture an e-commerce idea to create a common understanding of it. We use a graphical form of scenarios, called Use Case Maps (UCMs) [17]. UCMs have the notion of a path that shows how a particular scenario works out. A second motivation to use operational scenarios is to allow for evaluation of value propositions, as will be explained in Sect. 4.

2.3.4.2 Evolutionary scenarios The purpose of evolutionary scenarios is to do a sensitivity analysis of the potential profitability for all parties involved. These scenarios initially take an informal form, as they are expressed in natural language. Their content comprises possible, likely changes in the future, with respect to an e-commerce idea such as (dis)appearing actors, or a change in the way actors assign economic value to objects they receive or deliver. The initial scenarios are converted to changes in important parameters and variables in the value model so as to allow for quantitative forms of sensitivity analysis.

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 [18]. 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 to IT-intensive value propositions such that participating actors see how to make profit or obtain products which are of economic value for them by exploiting and using the system. This is our primary goal.

The remainder of this paper concentrates on the aforementioned value viewpoint, which takes a business economic perspective on innovative, IT-intensive value propositions. The next section introduces concepts needed to represent such a viewpoint, while Sect. 4 focuses on how to use these concepts in an e-commerce idea exploration track.

3 What is in a value model?

A value model shows actors who are exchanging things of economic value with each other. To express such a model, we have identified a number of generic concepts, relationships and rules, in short an ontology (see Fig. 2). This ontology is based on recent economics and business science literature on e-commerce [19, 20, 21], combined with formal ontology of systems theory [22]. Moreover, the ontology uses a conceptualisation [23] of use case maps.

We emphasise that a value model should not be confused with process or activity models. A value model

shows *what* is exchanged of economic value by whom, while a process model shows *how* this is operationally performed (for a more detailed discussion see [24]). Consequently, process modelling techniques like UML activity diagrams are not a good way to conceptualise a value model since the semantics of this technique focus on the flow of activities, while a value model presents what is offered to whom and what is requested for that in return in the economic sense.

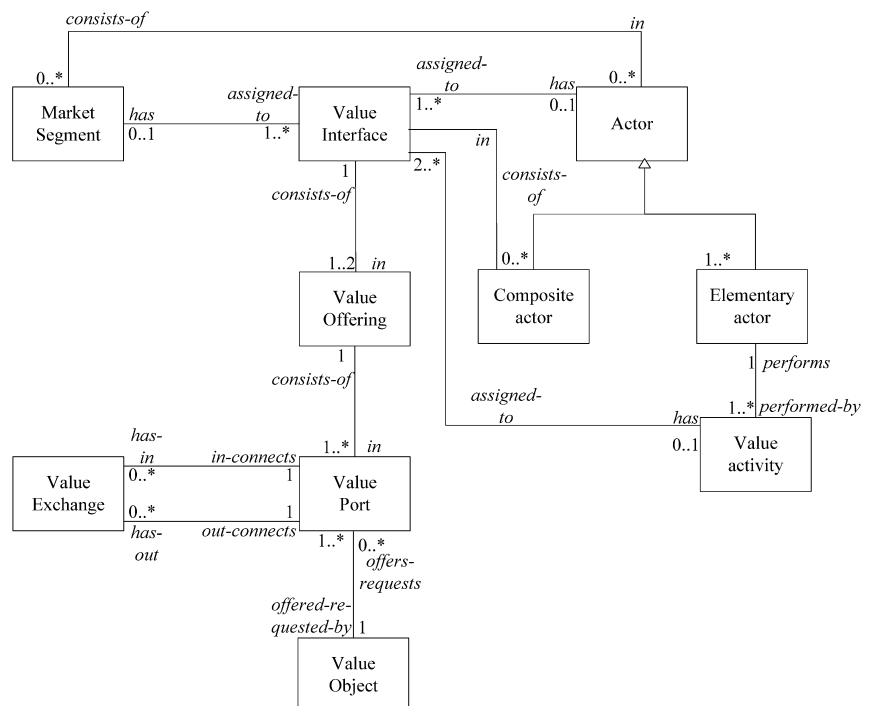
The conceptualisation of an e-commerce idea, which we call a value model, can be graphically represented (Fig. 1 shows a simple diagram; more realistic examples can be found in Sect. 5). For diagramming purposes, the reader can download a Visio tool stencil from our website at <http://www.cs.vu.nl/~gordijn/research.htm>. From the same website, an elementary Prolog implementation is available for value model representation and reasoning. Currently, we are developing advanced tool support in the EC-IST project Obelix (see <http://obelix.e3value.com>). The *e³-value* ontology has been extensively discussed elsewhere ([25, 3] chapter 3), so what follows is a summary of the most important concepts.

3.1 The e³-value ontology

3.1.1 Actor

An actor is perceived by its environment as an independent economic (and often also legal) entity. *Economically independent* refers to the ability of an actor to be profitable after a reasonable period of time (in the case of an enterprise), or to increase economic utility for

Fig. 2 Concepts and relations of the e³-value ontology for value models in e-commerce. The notation is based on UML class diagrams [9]. 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



him/herself (in the case of an end-consumer). In a sound, viable, value model *each* actor should be capable of making a profit or to do utility increase.

3.1.2 Value object

Actors exchange value objects, which are services, goods, money, or even consumer experiences. The important point here is that a value object is *of value* for one or more actors. Actors may value an object differently and subjectively, according to their own valuation preferences [19]. We deal with valuation in more detail in Sect. 4.

3.1.3 Value port

An actor uses a value port to show to its environment that it wants to provide or request value objects. The concept of port enables us to abstract away from the internal business processes, and to focus only on how external actors and other components of the value model can be ‘plugged in’.

3.1.4 Value offering

A value offering models what an actor offers to (an outgoing offering) or requests from (an ingoing offering) its environment, and closely relates to the *value interface concept* (see below). An offering is a set of equally directed value ports. The exchange of value objects via ports in an offering is atomic; all ports exchange an object or none at all.

3.1.5 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 clusters one ingoing and one outgoing value offering. It shows the mechanism of economic reciprocity. *Economic reciprocity* refers to rational 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 ingoing offering) in return. So, with the value interface, we can model that an actor is willing to offer something of value to its environment but requests something in return, whereas a value offering models that objects can only be requested or delivered in combination.

The exchange of value objects 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, or none at all. This ensures that if an actor offers something of value to someone else, it always gets in return what it wants. How this is ensured is a matter of a robust business process design, legal agreements, or sometimes use of technology, but this is not of interest for the value model.

3.1.6 Value exchange

A value exchange is used to connect two value ports with each other. It represents that two actors owning the connected ports are willing to exchange value objects with each other. As such, it corresponds to a *potential sale* in the AIAI Enterprise Ontology [26].

3.1.7 Market segment

A market segment is a concept that breaks a market (consisting of actors) into segments that share common properties [27]. Accordingly, our concept *market segment* shows a set of actors that for one or more of their value interfaces value objects equally from an economic perspective. In most cases, the individual actors of a market segment are left implicit. This is also the modelling purpose of the market segment construct: to have a shorthand for a large number of actors. However, it sometimes occurs that actors, being part of a market segment, exchange more value objects than only those mentioned by the market segment. Such actors should be modelled explicitly. The *consists-of/in* relationship between an actor and a market segment is then used to represent that an actor inherits the value interfaces from the market segment it is part of, in addition to the value interfaces such an actor already has.

3.1.8 Composite actor

A *composite actor* clusters value interfaces of other actors. Also, a composite actor has its own value interfaces to its environment. The purpose of a *composite actor* is twofold. First, it can be used to reduce complexity of a value model. We then group a number of actors into a *value constellation* [28]. Such a constellation is 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 second reason to introduce a composite actor 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. [29]). These actors then decide on one common value interface to their environment.

3.1.9 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 must yield a profit or should increase economic value for the performing actor. Consequently, we only distinguish a value activity if at least one actor, but hopefully more, believes that it can execute the activity profitably. 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.

3.2 Use case maps

The concepts above allow us to model who wants to do business with whom, but cannot represent *all* value exchanges needed to satisfy a particular end-consumer need. It often occurs that, to satisfy an end-consumer need, numerous other actors have to exchange objects of value with each other. As an example imagine a shop that exchanges economic value with an end-consumer: as a result, the shop must also exchange value with a wholesaler (see Fig. 1). It is our experience that showing all such value exchanges to satisfy an end-consumer need contributes largely to a common understanding of an e-commerce idea. To that purpose we use an existing scenario technique called Use Case Maps (UCMs) [17]. UCMs show which value exchanges should occur as a result of a consumer need (which we call a start stimulus), or as a result of other value exchanges. Below, the main UCM modelling constructs are briefly discussed.

3.2.1 Scenario path and segment

A scenario path consists of one or more *segments*, related by connection elements, start and stop stimuli, and responsibility points. A path indicates via *which* value interfaces objects of value must be exchanged, as a result of a start stimulus, *or* as result of exchanges via *other* value interfaces.

3.2.2 Stimulus

A scenario path starts with a **start stimulus**, which represents a consumer need. The last segment(s) of a scenario path is connected to a **stop stimulus**. A stop stimulus indicates that the scenario path ends.

3.2.3 Connection element

Connections are used to relate individual segments. An **AND fork** splits a scenario path into two or more sub-paths, while the **AND join** collapses sub-paths into a single path. An **OR fork** models a continuation of the scenario path into one direction that is to be chosen from a number of alternatives. The **OR join** merges two or more paths into one path.

3.2.4 Responsibility element

Another way to connect path segments is to use a responsibility element. A responsibility point shows that a scenario path 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 Sect. 4).

4 How to construct a value model

Whereas the previous section outlined the concepts present in a value model, this section provides a prototypical approach for constructing value models (see Fig. 3). An extensive discussion of this process can be found in [3] (chapter 5).

4.1 Construction of a baseline value model

Our approach assumes that stakeholders have already an e-commerce idea in mind. The goal of e^3 -value is to clarify and evaluate such an idea more thoroughly, not to find the ideas themselves. A first step is to construct a baseline value model for such an idea and comprises the following tasks.

4.1.1 Operational scenario identification

Value model construction starts with identification of operational scenarios. These are initially only short sentences, denoting the product, service, or experience

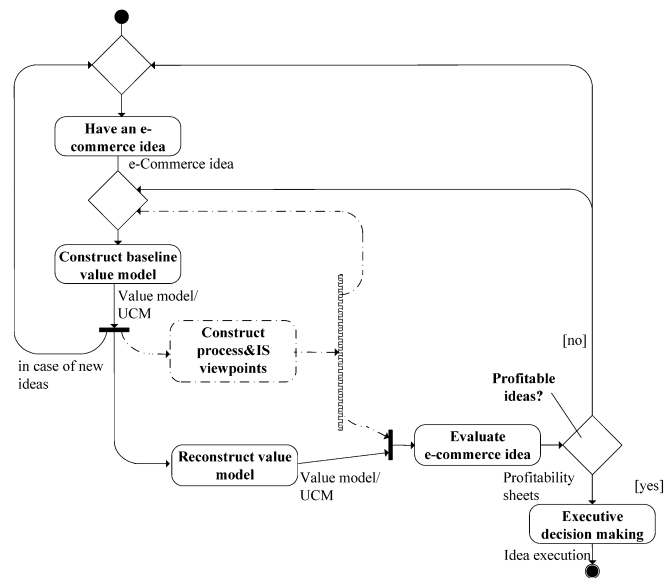


Fig. 3 The e^3 -value methodology supposes an innovative e-commerce idea for which a baseline value model will be constructed. Often, construction of such a model yields new ideas, because stakeholders are forced to think about their business in a predefined framework. Once a baseline value model exists, variations can be found on this model by reconstructing it in a structural way. Finally we evaluate e-commerce ideas for potential profitability. This diagram uses the UML activity diagram notation, which is not part of e^3 -value

desired by a *customer*. It is our experience that it is hard to find these scenarios and to articulate them well in a first step. Consequently, as can be seen from Fig. 3, construction of a value model is a cyclic process. It is our experience that after a number of cycles stakeholders can define scenarios more accurately.

In practice we have experienced that an idea is often described by outlining the business processes supporting an e-commerce idea, rather than by outlining the aforementioned scenarios. If we focus on processes, we implicitly suppose some valuable objects wanted by a consumer. This inhibits us from explicitly discussing the objects of value and consequently we take the risk to overlook promising propositions, so taking a process perspective is not a good starting point.

4.1.2 Actor identification

After scenario identification, 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, it happens that some actors have been removed or added to this list, caused by a better understanding of the needed kind of actors for an e-commerce idea.

Actors are mentioned by stating their company name (named actors), or in the case of end-consumers by the role they play (non-named actors). The distinction between named and non-named actors has also been made by Ould [11], who distinguishes an *actor* (e.g. George Bush) from a *role instance* (e.g. the president of the United States).

Additionally, we distinguish *environmental* actors. These are actors we are not interested in from a profitability perspective (we assume that they are profitable beforehand), 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 its value objects from someone.

4.1.3 An actor versus market approach

If a tentative list of actors is known, we explore *what* these actors are offering to each other. We have found two approaches to do so: (1) an actor-driven track, and (2) a 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 its environment, and related concepts such as value interfaces, value ports and objects. Hereafter, value exchanges with other actors are identified. We use this track if the e-commerce is initiated by only one actor. 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. The market-driven track is of use if a consortium of

actors initiates the idea, which happens more and more in practice.

4.1.4 Value object identification

Both the actor track and the market track suppose identification of value objects specifying *what* is offered or requested by an actor (actor track), or *what* is exchanged between actors (market track). The criterion used for distinguishing value objects is that a value object must be of economic value for *at least* one actor. Thus, a value object does not need 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 [19], and consequently not both actors have to assign economic value to an object.

We use three guidelines to find value objects: (1) analysis of the e-commerce idea and scenarios, (2) use the notion of economic reciprocity, and (3) use causally related value objects. First, the e-commerce idea and scenarios should trigger identification of value objects. If at least one value object is found, stakeholders can be asked for *reciprocal* value objects. A reciprocal value object is something of value that should be offered in return for obtaining another value object, and refers to the notion of rational economic behaviour. It is our experience that for nearly each found value object at least one reciprocal value object can be elicited. Finally, we search for *causally related* value objects. To be able to offer a value object to its environment, it is likely that an actor must obtain at least one other object, which we call a causally related value object. This is, for instance, the case for a trading company. Objects that are sold must also be bought.

4.1.5 Grouping value ports into value offerings and interfaces

If an actor-oriented approach is followed, value ports should be specified which offer or request the found value objects. Moreover, the ports should have a direction, indicating whether objects are offered or requested. In case of a market-oriented track, value exchanges are elicited, which exchange the found value objects. Value ports are then the end-points of these exchanges.

Value ports are grouped into value offerings. The grouping of value ports into a value offering denotes the decision that the exchange of objects via these ports can only be done in combination.

A value offering is of use for representing a number of situations. First, some objects may only of value for an actor if they are obtained in combination. In-ports exchanging such objects then form an ingoing offering. Second, actors may decide to offer objects only in combination to their environment. Ports offering such objects then form an outgoing offering. An example of an outgoing offering is the case of *mixed bundling*. This 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 if they were sold separately [30].

Then, found value offerings of an actor are grouped into a value interface. This is used to model economic reciprocity. Consequently, the reciprocity heuristic we used previously to identify value objects can also be used to group value offerings into a value interface. In contrast, causal value ports and offerings are not grouped into a value interface. 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 is only willing to exchange an object, o_{out} , if it obtains another object o_{in} in return. Moreover, it must assign to object o_{in} a higher economic value than to object o_{out} .

4.1.6 Scenario path identification

Scenario paths show which value objects need to be exchanged as a result of a customer's need. This need is shown as a *start* stimulus. To satisfy this need, an actor must exchange objects of value via a value interface, which we show by connecting the start stimulus with a *segment* to a *responsibility element* touching the value interface. In case the actor can choose from more than one value interface for need satisfaction, the start stimulus is connected via an *OR-fork* connection element and multiple segments to responsibility points touching these value interfaces. The exchange of value objects via an actor's value interface always implies exchanges via a value interface of another actor. This results in a continuation of the scenario path by using a scenario segment and a responsibility element again. If no exchanges are needed anymore, the scenario path stops with an *end-stimulus*.

4.2 Construction of other viewpoints

After the construction of a baseline value model, we explore requirements from a business process viewpoint and an information systems viewpoint. The main purpose for doing so is to reveal substantial operational and capital expenses. Also, exploration of such viewpoints provides a first glance on the technical feasibility of the e-commerce idea. This paper focuses on the exploration of the value viewpoint only. More information on exploration of other viewpoints in conjunction with the value viewpoint can be found in [3] (chapter 9).

4.3 Value model deconstruction and reconstruction

In practice, many variations on a baseline model can be thought of. Value model deconstruction and reconstruction is intended to find such design variations in a

structural way and is inspired by Business Science literature (see [31, 21, 32]).

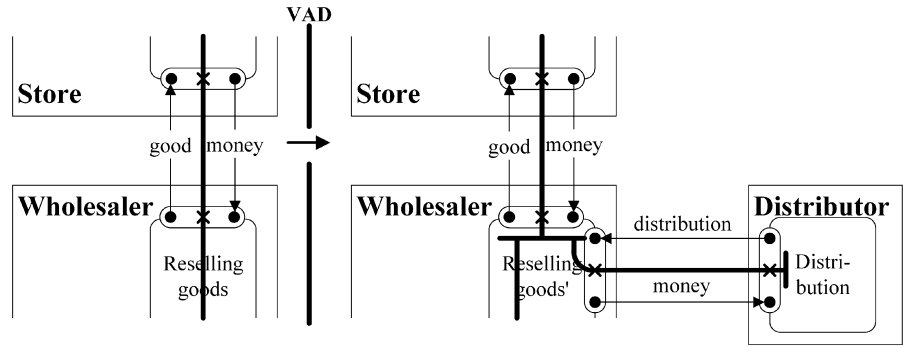
To deconstruct a value model, *e³-value* defines value model *deconstruction* operators. These are part of a value model deconstruction and reconstruction process, during which we *de-assign* activities from their performing actors, try to find alternatives and/or more activities by deconstructing existing 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?

For a baseline value model, we initially assume one value activity per actor. Such an activity should name the profitable or utility-increasing activity performed by that actor. We then use deconstruction operators, to break down the constructs in value model into smaller pieces.

One of these operators is the value activity deconstruction (*VAD*) operator (other deconstruction operators are discussed in [3], chapter 6). This operator is used to 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. 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 example to allow specialised actors to perform one of these value activities. Other operators include the value interface deconstructor *VID*, partitioning ports of a value interface into smaller value interfaces while keeping in each partition economic reciprocal ports, and the value port deconstructor *VPD*, deconstructing a port into a number of ports each offering or requesting a value object.

Figure 4 exemplifies how to use this operator in practice. For the actors in Fig. 1, we assume for each actor one value activity; for the wholesaler this activity is *reselling goods*. By using value activity deconstruction, we split up this activity into two smaller activities: *reselling goods'* and *distribution of goods*. Apparently, we implicitly assumed that the original value activity *reselling goods* includes the commercial activity of selling (*reselling goods'*) and includes the value activity *distribution*. In addition to splitting up a value activity into smaller parts, extra value exchanges, interfaces and ports may be required to ensure that still the same value objects are offered and requested as was the case before deconstruction. In the example, this results in value exchanges between the activities *reselling goods'* and *distribution*. Note that in this example deconstruction represents *outsourcing* of distribution of goods. After deconstruction, a new value model has to be reconstructed by assigning newly found activities to potential new actors, in this case a distributor.

Fig. 4 The activity *Reselling goods* is deconstructed in two other value activities



4.4 Idea evaluation

If an e-commerce idea has been articulated well by developing one or more value models, the idea should be evaluated for feasibility. In this paper, we focus on evaluation of value models only; note that for a more comprehensive evaluation of feasibility other viewpoint types are required. 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.

4.4.1 Profitability sheet creation

Profitability sheets are constructed for each actor involved, and present revenues and expenses associated with the execution of the e-commerce idea under consideration. It contains for each actor value objects flowing into and out of the actor as a result of scenario path execution. 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*.

Table 2 shows a profitability sheet for the actor *wholesaler*, based on the value model presented in Fig. 1

4.4.2 Value assignment

After a profitability sheet for each actor has been constructed, actors are asked to assign economic value to objects flowing into or out of 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 into account operational expenses as a result of executing business processes and exploiting an information system. Also, investments

Table 2 The structure of the profitability sheet, which shows the ingoing and outgoing of value objects for a specific actor and scenario path execution

Actor	wholesaler	
Scenario	<i>need a good</i>	
Occurrences/timeframe	...	
Scenario path	1	
	Value Object In payment good	Value Object Out good payment

needed are not part of this profitability number. However, if for one of the actors profitability is less than or equal to zero, the e-commerce idea is not likely to be profitable for such an actor, given the identified model and estimates on scenario occurrences 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; and (2) end-consumers: these are actors who do not resell value objects, but use obtained objects to *create value* for themselves.

Enterprises want to maximise their profit: in short, revenues minus expenses. As such, we only take into account value objects representing money flows to calculate an enterprise's profitability sheet. This is also suggested by investment theory (see e.g. [33]), taking into consideration cash-in and -out flows only. We assume that all other objects (not representing money) flow into an enterprise, and after some time flow out of the same enterprise, and are not of relevance to determining profitability. Consequently, enterprise actors are asked to determine a *valuation* function, which returns the amount of money to be paid for products exchanged via the same value interface. Determination of such a function can be done by one actor, or can be the outcome of a negotiation process between actors exchanging objects of value.

In contrast, 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 to an unequal extent. Some objects will fulfil end-consumer's needs nearly completely, while others do so only in a very limited way. 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. [27]). Second, to obtain an object, an end-consumer must give another object in return. In most cases this is a fee in some amount of money (say, euros). According to [27], the end-consumer then will choose the object that delivers the most utility per euro, if he/she is a rational acting person. This is in axiology literature also known as *consumer value* maximisation [19]. As a consequence, to assess to what extent an end-consumer maximises 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 a monetary unit. By doing so, we make non-monetary objects comparable with monetary objects seen from a utility perspective. In [34], we discuss in more detail this utility-based end-consumer valuation, to reason about scenarios to sell music legally on the Internet versus illegal downloading.

4.4.3 Assessment of evolutionary scenarios

If we know the number of scenario path executions per timeframe, we can calculate the expected profitability for each actor. Note that in this paper we only consider the value viewpoint, so the profitability per actor should be seen as an amount of money that should be sufficient to cover operational and capital expenses, plus a profit margin.

It is our experience that *numbers* on profitability *themselves* are not very useful for stakeholders involved, because it is not possible to predict profitability numbers for innovative e-commerce ideas accurately. Results of exploiting 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 important for stakeholders, however, 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 reason about profitability, we employ *evolutionary* scenarios. In contrast to *operational* scenarios, which describe behavioural 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 analysed, as well as effects of wrong estimations.

Two extreme positions on finding scenarios exist [35]. At one extreme, scenarios can be collected empirically.

This is often done by interviewing stakeholders, or having workshops on scenario identification. At 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 organise scenarios but also to generate scenarios. We employ a middle-out approach. Scenarios are elicited by interviewing stakeholders and by doing executive workshops, with types of evolutionary scenarios in mind. We distinguish the following scenario types: (1) scenarios that capture a change in valuation functions, (2) scenarios that represent a change in the expected number of scenario path occurrences, and (3) scenarios that suppose a change in the structure of the value model itself, such as actors entering or leaving the value model.

5 An explorative e-commerce project

We have developed the *e³-value* methodology by following an action research approach (see [36, 37] and [38] for a tutorial). 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 organisations. As innovative e-commerce idea exploration is an (inter-) organisational 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.

During the course of our 4-year research period, we have been working for major firms in the realm of strategic e-commerce consultancy, doing innovative e-commerce idea exploration tracks. Additionally, we have used an academic context to reflect on the *e³-value* methodology after using it in these real-life project settings. As a result of experiences in carrying out such projects, the *e³-value* methodology has matured. In this section we report on one such project, which is about the exploration of an online news service. We have also done real-life projects on free Internet access provisioning [3] (chapter 3), a contacts ads service [39], and the possibility of selling music [34].

5.1 E-commerce idea

A newspaper publisher wants to offer an archive of online newspaper articles for free. Only a subscription on the paper-based version of the newspaper is required and a telephone connection for data transport (based on the TCP/IP protocol) between the online article archive and the reader of articles.

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 dialling 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, it is capable of generating a large number of terminations for an *article online* service.

5.2 Value model

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 a call termination or call origination model.

5.2.1 Terminating value model

A *terminating* value model is shown in Fig. 5. 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.

5.2.1.1 Readers A start stimulus is caused by a reader if he/she wants to read an online article. Readers are subscribers on a newspaper, the *Amsterdam Times*, and

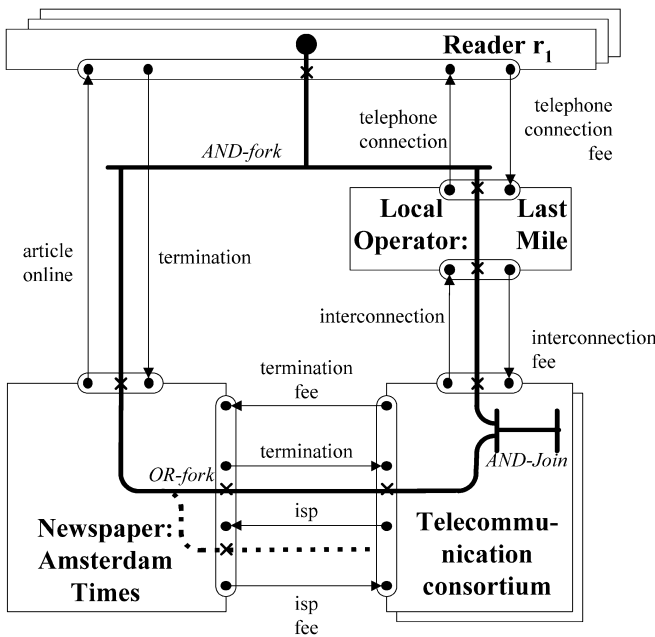


Fig. 5 A value model based on call termination

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*.

5.2.1.2 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 its large subscriber base, the *Amsterdam Times* has the potential to generate a large number of terminations.

5.2.1.3 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 fibre 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 TCP/IP protocol. At the time this exploration track was carried out, only one local operator existed in the Netherlands, so only one such actor has been modelled.

5.2.1.4 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 suggests, 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 access server, but in reality only operates the last mile of 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 as 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 utilising the infrastructure of the telecommunication consortium.

Multiple telecommunication consortia Finally, note that Fig. 5 shows two telecommunication consortia, rather than one, to enlarge the power of the *Amsterdam Times* with respect to selection of telecommunication parties. 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. 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 (dashed line) in the scenario path, which models the supplier selection by the *Amsterdam Times*.

5.2.2 Originating value model

Figure 6 presents an *originating* value model. In contrast to the terminating model, the originating model assumes that the *Amsterdam Times* offers its online article service *directly* without any intermediate partners visible to its 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.

5.2.2.1 *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.

5.2.2.2 *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 no longer needs to obtain a telephone connection (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.

5.2.2.3 *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.

5.2.2.4 *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.

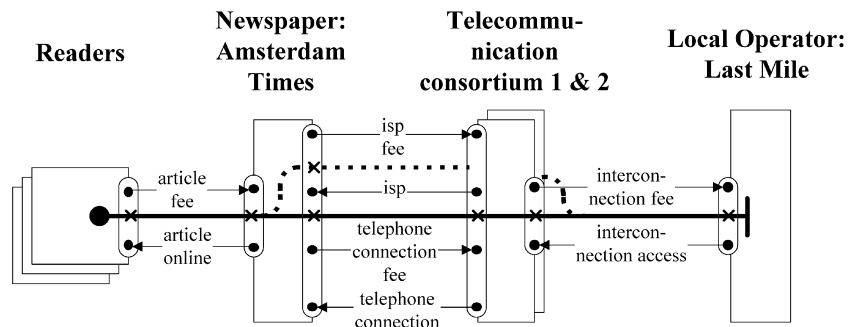
5.3 Evaluation

5.3.1 Profitability sheet and valuation

Table 3 shows a profitability sheet based on call termination. Calculation of fees is done as follows.

5.3.1.1 *Telephone connection fee* The telephone connection fee per scenario occurrence is based on a start

Fig. 6 A value model based on call origination



tariff and a connection time-dependent tariff. To calculate the total monthly fees, the telephone connection fee is multiplied by the realised number of scenario occurrences.

5.3.1.2 Interconnection fee The interconnection fee per scenario occurrence (here only shown for *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 interconnection bridges.

5.3.1.3 Termination fee The termination fee the *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 analysing the effects of a decreasing interconnection factor (e.g. influenced by a market regulator), while the revenue-sharing factor remains the same. This models a situation where the telecommunication consortium takes the risk of a decreasing interconnection factor.

5.3.1.4 IP access fee The telecommunication consortium charges the *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; the consortium wants to have the opportunity to assign unused IP access ports to others. Therefore, the *Amsterdam Times* is asked to forecast the number of scenario occurrences on a monthly basis, including the average duration. The telecommunication consortium then allocates access ports on this forecast, and can allocate remaining ports to others. To motivate the *Amsterdam Times* to do good forecasting, the following valuation is used: if the number of realised scenario occurrences drops below an inaccuracy factor (e.g. 75%)

```

1 forecast-IP(realized-occurrences,
              forecast-occurrences, inaccuracy-factor)
2 {
3   if realized-occurrences < forecast-occurrences×
   inaccuracy-factor then
4     return (forecast-occurrences×
   inaccuracy-factor)/realized-occurrences;
5   else
6     return 1;
7   endif
8}

```

Scheme 1 Forecast formula for the use of IP access by the *Amsterdam Times*

of the forecast occurrences, we use 75% of the forecast occurrences for the calculation of the monthly IP access fee. Otherwise, we use the realised number of scenario occurrences (see Scheme 1).

5.3.1.5 Hosting fee The hosting fee is calculated in a similar way as the IP access fee. The telecommunication consortium uses a fee per page view times the average number of page views times a factor, which motivates the *Amsterdam Times* to be a good forecaster.

5.3.2 Assessment of evolutionary scenarios

To assess sensitivity of the profits for actors due to expected future events or wrong estimates, we employ evolutionary scenarios. As an example, Table 4 shows some of such scenarios. The estimates for the value objects are based on observable properties for parts of the value functions (e.g. telecommunication tariffs), and estimates on properties which are negotiable such as interconnection and termination fees. Moreover, the number of scenario occurrences is based on data on the online behaviour of subscribers (such as minutes online/period and number of Internet enabled PCs/subscriber).

The *null* scenario refers to a best estimate at the time the exploration was carried out. Observe that *all* actors make a profit. What happens if the *Amsterdam Times* is not a good forecaster of scenario occurrences? It can be seen that the *Amsterdam Times* will not make a profit. For *Last Mile* and the telecommunication consortium

Table 3 Profit sheet for the scenario *Read article online*, for the terminating value model

Scenario	Read article online	
Actor	Value Object In	Value Object Out
<i>Last Mile</i>	tel. connect. fee = tel. start tariff + (tel. connect. tariff × duration)	interconnect. fee _{composite 1} = tel. connect. fee × distance factor _{composite 1} × interconn. factor _{composite 1}
<i>Amsterdam Times</i>	termination fee _{composite 1} = tel. connect. fee × rev. sharing factor _{composite 1}	inet access fees _{composite 1} = see composite 1
Composite 1	interconnect. fee = see <i>Last Mile</i> inet access fee = inet. connect. tariff × duration × forecast-IP(...) hosting fee = page view tariff × average page views × forecast-PV(...)	termination fees = see <i>Amsterdam Times</i>
Composite 2

Table 4 Different valuation scenarios. The null-scenario is a best estimate. A second scenario assumes that the *Amsterdam Times* forecasts inaccurately. A decrease in the interconnection is expected to occur, especially due to competition between telecommunication

Scenarios	Profit		
	<i>Amsterdam Times</i>	<i>Last Mile</i>	<i>Telecomm.</i>
Null-scenario	164,400	102,000	121,800
Forecast(1,500,000) >> Actual (150,000)	-28,560	10,200	34,680
Decrease in interconn. factor (1.0 to 0.4)	164,400	346,800	-600
Decrease in revenue sharing factor (0.5 to 0.1)	-19,200	102,000	213,600

there is still a profit to cover the expenses. It is reasonable to expect a decrease in the interconnection factor after some months, because presently this factor is high to stimulate competition between telecommunications operators. As soon as this competition works, this factor will decrease. The *Amsterdam Times* does not feel such a decrease, but the telecommunication consortium will. As a result, the consortium may decide to decrease its revenue-sharing factor. As can be seen, this will harm the *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 at hand. This sensitivity analysis is in many cases of more business interest than the numbers of the valuation itself.

5.4 Lessons learned

E-commerce idea exploration for the case at hand, as well as its implementation, took place during December 1999 to February 2000. The project was carried out for a publisher of daily newspapers in the Netherlands. In September 2001, the publisher publicly announced a stop to most of its Internet-related activities, of which the service outlined in this paper is part. It is likely that the online article service explored in this paper will be phased out in the coming years. Because of this, we revisited the publisher in November 2001. The goal of this visit was first to understand the publisher'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.

5.4.1 Profit and loss responsible business units of an enterprise should be visible in a value model

The publisher has a number of newspapers called titles, which serve a specific market segment. These titles were not explicitly modelled in our value models (see Figs 5 and 6). We have only identified an actor called the *Amsterdam Times*, denoting the publisher and all its titles. In addition, the publisher has a more technical, facilitating, department, focusing on printing regular

actors increases (see the third case). The fourth scenario supposes a drop in the revenue sharing factor between *Data Runner* and the *Amsterdam Times*

newspapers, and managing online services from a technical perspective. We have not modelled this facilitator also.

Not distinguishing the publisher'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 the publisher.
- *Interests of the publisher's business units such as titles and facilitator are unclear*: the model does not show how business units as independent profit centres earn money with the online article service.

To address the mentioned drawbacks, consider Fig. 7, which illustrates a possible value model for the publisher including its business units. Titles now are responsible for offering online articles to their subscribers. To stimulate selling, titles receive a modest fee (a fraction of the termination fee), which directly relates to the use of the online article service.

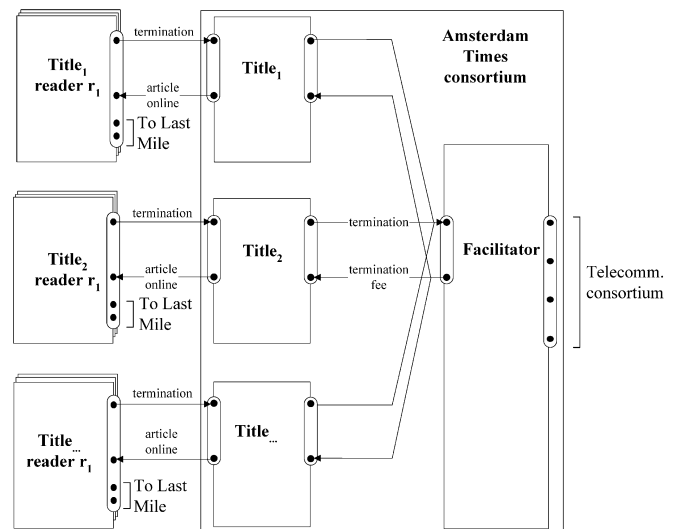


Fig. 7 A fragment of a value model emphasising that (1) individual titles have the end-consumer relations and are responsible for 'selling' the online articles, and (2) titles receive fees based on the amount of traffic they generate with their content

5.4.2 Value models change over time

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 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.

5.4.2.1 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 the publisher and the telecommunication consortia continued. They felt that the designed value model was too complex, and so they decided to choose a model represented by Fig. 8. The difference from the original model (see Fig. 5) is that the publisher pays a very modest fee to the telecommunication consortium for hosting and access. So, in the new model the publisher is not paid, but rather must pay a modest fee itself.

Such a new value model only works if there are revenues for the publisher 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 such a solution, as can be seen from Fig. 8: fees are only paid by the publisher and not received. It is also not clear how the business units themselves are funded for this service. This is one of the main reasons why the online article service cannot survive.

5.4.2.2 Unbundling access and online articles The original value model (see Fig. 5) assumes that the only way to access an online article is to set up a telephone connec-

tion with a selected telecommunication actor. With such a telecommunication actor, the publisher 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 are paid to the telecommunication consortium and the publisher.

Some titles have chosen not to bundle access and the online article (see Fig. 9). Readers of a specific title can choose an Internet Service Provider (ISP) themselves to access the online articles. To do so, the online article archive 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 the publisher does not receive a termination fee. This disrupts the designed value model presented in Fig. 5, but also shakes up the implemented value model in Fig. 8. In the latter case, the telecommunication consortium no longer receives fees to finance its service offered. As a result, this actor may charge an additional fee, e.g. to the title responsible for unbundling. It is questionable (denoted by the question mark in Fig. 9), 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.

From both examples, we conclude that an e-commerce idea continuously evolves significantly over its lifetime. A value model should therefore be maintained and evaluated for each major change. For the specific case at hand, the consequences of removing the termination fee value exchange between the telecommunication consortium and the *Amsterdam Times*, as well as unbundling the online article and access, can be shown using our e^3 -value modelling technique. Also, Fig. 8 illustrates that the *Amsterdam Times* has no income after

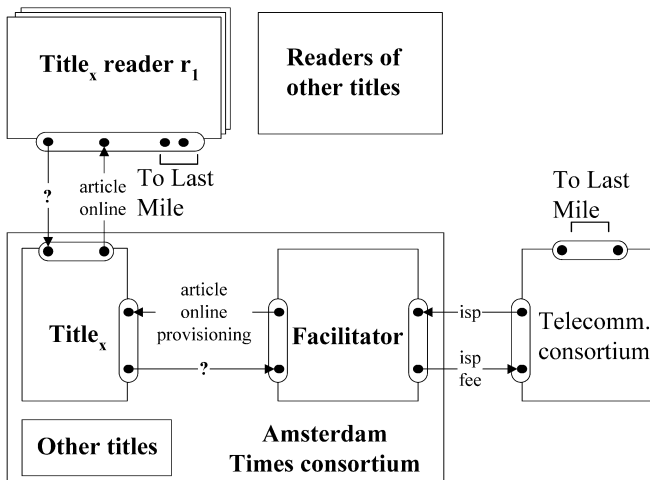


Fig. 8 A fragment of the value model implemented, which differs from the proposed model (see Fig. 5): *Amsterdam Times* has no additional funding

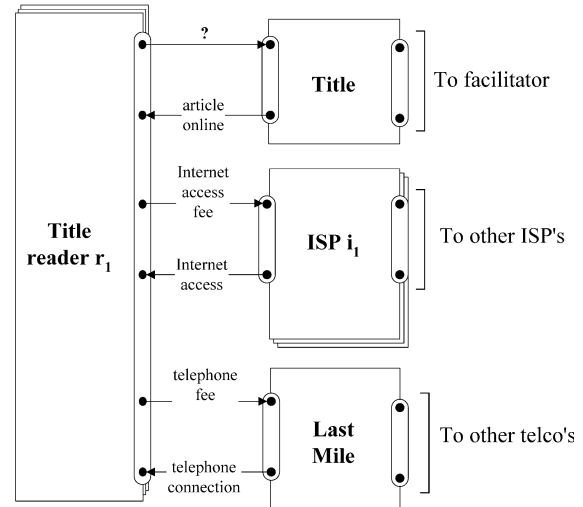


Fig. 9 A fragment of a value model implemented by one of the titles: unbundling of access and online articles

changing the value model, and by constructing new scenario paths and profitability sheets, it can be seen from Fig. 9 that the telecommunication actor misses revenues as a result of unbundling.

6 Related work

There have been a few other ontological approaches on business modelling like the AIAI Enterprise Ontology (AIAI EO), the Toronto Virtual Enterprise Ontology (TOVE), and the Resource Event Agent (REA) ontology, which are briefly discussed below. The drawback of these ontologies is that they are not lightweight, but in contrast consist of a large number of concepts and relationships. Additionally, these ontologies do not come with suitable graphical description techniques, and also they have no guidelines for use during idea exploration tracks.

6.1 AIAI enterprise ontology

The AIAI enterprise ontology [26] 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 section illustrates the need for such a concept.

6.2 Toronto Virtual Enterprise ontology

The TOVE ontology [40] identifies concepts for the design of an agile enterprise. An agile company integrates his/her structure, behaviour and information. The

TOVE ontology currently spans knowledge of activity, time and causality, resources, cost, quality, organisation 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.

6.3 Resource Event Agent ontology

The Resource Event Agent (REA) ontology [41, 42] shows from an ontological perspective many similarities to the e^3 -value ontology. REA calls *actors agents*. Agents are offering or requesting *resources* (in e^3 -value called *value objects*) by economic *events*. The latter can be compared to *value ports* in e^3 -value. REA relates economic events of different actors by *exchanges* which correspond to e^3 -value *value exchanges*. Finally, economic events of an agent are related by a *duality* relation. This models economic reciprocity which is handled by e^3 -value by the notion of *value interface*.

From an ontological perspective, e^3 -value and REA differ with respect to the notion of *value activity*. This concept is lacking in REA, but is important for e-commerce idea exploration. A value activity is a potential profitable activity for one or more actors. Because e-commerce development tracks are characterised by shifts in actors performing these activities, it is important to model value activities explicitly.

From a methodological point of view, REA is not an approach for business development, whereas e^3 -value provides a methodology for doing so, e.g. by value model construction and reconstruction, and by profitability-based sensitivity analysis.

7 Key points and future research

7.1 Key points

The e^3 -value methodology is about the economic value-aware exploration of innovative e-commerce ideas, which utilises principles from both requirements engineering and conceptual modelling, and focuses on the exploration of an IT-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*. 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.

Although the development of an e-commerce model focuses on business requirements in general and potential profitability of the idea in particular, the model can also be used as a starting point for a more software-oriented requirements engineering process. A value model then contributes to a better understanding of the e-commerce by system architects and software developers and thereby frames the software requirements engineering process. Based on experiences in exploring e-commerce ideas, we propose a methodology which: (1) is lightweight, (2) is a graphical, conceptual modelling approach, (3) is based on multiple viewpoints, (4) exploits scenarios, both operational and evolutionary, and most importantly (5) recognises the importance of economic value creation and distribution, which is key to innovative e-commerce initiatives.

To represent and analyse the economic value perspective in a model-based way, we have developed an ontology which can be used to represent a multi-actor network exchanging objects of value. Operational scenarios are used to analyse the model for profitability in conjunction with evolutionary scenarios to do a sensitivity analysis on expected profits. Thereby, we recognise that for innovative e-commerce ideas it is nearly impossible to predict profitability; rather we aim at the more modest goal to reason about factors influencing this profitability. Finally, to find variations on a baseline value model, we use an approach called value model deconstruction and reconstruction to create in a structured way new value models.

7.2 Future research

The e^3 -value methodology is only the beginning of value-based requirements engineering. Much work is yet to be done to understand IT-intensive new business development.

For instance, how can we relate the three identified viewpoints, 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. In recent work [43], we propose the use of a feature-solution graph [44] to do so. Viewpoints are split up into features and solutions, which are connected by different types of relations. Some features, for example, can have multiple solutions, or can be positively influenced by 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 results in requirements on the information system viewpoint. By modelling these relations explicitly, we can reason about choices for a particular feature and solution on each viewpoint.

Another topic for future research is the use of *value patterns*. In the realm of information technology, analysis [45] and design patterns [46] 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.

One issue, proposed by our industry contacts, is to come to more reliable profitability predictions. Currently, the profitability sheets are estimates to do a sensitivity analysis and should not be seen as predictions for profitability and consumer value. 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 cannot 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 easily possible yet to predict whether or how quickly 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, for serving only two articles online per minute a lightweight web-server may be sufficient, while for thousands of articles per minute a heavyweight solution such as a load-balancing farm of web-servers is needed.

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 [47]. 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).

Finally, tool support is needed for drawing and checking models (e.g. for compliance with the *e³-value* ontology), as well as to evaluate value models. At the time of writing, no integrated tool support is available. We will develop a value-modelling case tool in the EC-IST project OBELIX [48].

Acknowledgements This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project VW1.4949, EU-IST project IST-2001-33144 Obelix and EU-EESD project NNE5-2001-00256 BusMod.

References

1. Shama A (2001) Dot-coms' coma. *J Syst Softw* 56(1):101–104
2. Rogers EM (1999) Diffusion of innovations. Free Press, New York
3. Gordijn J (2002) Value-based requirements engineering: exploring innovative e-commerce ideas. PhD thesis, Vrije Universiteit, Amsterdam. Also available from <http://www.cs.vu.nl/~gordijn/>
4. Loucopoulos P, Karakostas V (1995) System requirements engineering. McGraw-Hill, Maidenhead, UK
5. Finkelstein A, Kramer J, Nuseibeh B, Finkelstein L, Goedicke M (1992) Viewpoints: a framework for integrating multiple perspectives in system development. *Int J Softw Eng Knowl Eng* 2(1):31–58
6. Sommerville I, Sawyer P (1997) Viewpoints: principles, problems and a practical approach to requirements engineering. *Ann Softw Eng* 3:101–130
7. Motschnig-Pitrig R, Nissen HW, 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), June 1997. Also CREWS Report 97-11
8. Fowler M, Scott K (1995) UML distilled: applying the Standard Object Modelling Language. Addison-Wesley Longmann, Reading, MA
9. Rumbaugh J, Jacobson I, Booch G (1999) The Unified Modelling Language reference manual. Addison-Wesley Longmann, Reading, MA
10. van Hee KM (1994) Information systems engineering: a formal approach. Cambridge University Press, Cambridge
11. Ould MA (1995) Business processes: modelling and analysis for re-engineering and improvement. Wiley, Chichester
12. Davenport TH (1993) Process innovation : reengineering work through information technology. Harvard Business School Press, Boston, MA
13. Mylopoulos J (1992) Conceptual modeling and Telos. In: Conceptual modelling, databases and CASE: an integrated view of information systems development. Wiley, New York, pp 49–68
14. Meyer B (1985) On formalism in specifications. *IEEE Softw* 2(1):6–26
15. Wieggers KE (1999) Software requirements. Microsoft Press, Redmond, WA
16. Antón AI, Potts C (1998) A representational framework for scenarios of system use. *Requirements Eng* 3(3/4):219–241
17. Buhr RJA (1998) Use case maps as architectural entities for complex systems. *IEEE Trans Softw Eng* 24(12):1131–1155
18. Yu ESK, Mylopoulos J (1998) Why goal-oriented requirements engineering. In: Dubois E, Opdahl AL, Pohl K (eds) Proceedings of the 4th international workshop on requirements engineering: foundation for software quality (RESFQ 1998). Presses Universitaires de Namur, Namur
19. Holbrook MB (1999) Consumer value: a framework for analysis and research. Routledge, New York
20. Porter ME (2001) Strategy and the Internet. *Harvard Business Rev March*:63–78
21. Tapscott D, Ticoll D, Lowy A (2000) Digital capital—harnessing the power of business webs. Nicholas Brealy, London
22. Borst WN, Akkermans JM, Top JL (1997) Engineering ontologies. *Int J Hum-Comput Stud* 46:365–406
23. Amyot D, Mussbacher G (2000) On the extension of UML with use case maps concepts. In: Evans A, Kent S, Selic B (eds) UML 2000: The Unified Modeling Language: advancing the standard. Lecture notes in computer science, vol 1939. Springer, Berlin Heidelberg New York pp 16–31
24. Gordijn J, Akkermans JM, van Vliet JC (2000) Business modelling is not process modelling. In: Liddle SW, Mayr HC (eds) Conceptual modeling for e-business and the web. Lecture notes in computer science, vol 1921. Springer, Berlin Heidelberg New York, pp 40–51. Also available from <http://www.cs.vu.nl/~gordijn/>
25. Gordijn J, Akkermans JM (2001) Designing and evaluating e-business models. *IEEE Intell Syst – Intelligent e-Business* 16(4):11–17
26. Uschold M, King M, Moralee S, Zorgios Y (1998) The enterprise ontology. *Knowl Eng Rev* 13(1):31–89
27. Kotler P (1988) Marketing management: analysis, planning, implementation and control. Prentice-Hall, Englewood Cliffs, NJ
28. Normann R, Ramirez R (1994) Designing interactive strategy: from value chain to value constellation. Wiley, Chichester
29. Davidow WH, Malone MS (1992) The virtual corporation: structuring and revitalizing the corporation for the 21st century. HarperCollins, New York
30. Choi S-Y, Stahl DO, Whinston AB (1997) The economics of doing business in the electronic marketplace. Macmillan Technical, Indianapolis, IN
31. Evans P, Wurster TS (2000) Blown to bits: how the new economics of information transforms strategy. Harvard Business School Press, Boston, MA
32. Timmers P (1999) Electronic commerce: strategies and models for business-to-business trading. Wiley, Chichester
33. Horngren CT, Foster G (1987) Cost accounting: a managerial emphasis, 6th edn. Prentice-Hall, Englewood Cliffs, NJ
34. Gordijn J, Akkermans JM, van Vliet JC, Paalvast ERMR (2000) Selling bits: a matter of creating consumer value. In: Bauknecht K, Madria SK, Pernul G (eds) First international conference on electronic commerce and web technologies (EC-Web 2000). Lecture notes in computer science, vol 1875. Springer, Berlin Heidelberg New York, pp 48–62. Also available from <http://www.cs.vu.nl/~gordijn/>
35. Carroll JM, Rosson MB (1992) Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Trans Inform Syst* 10(2):181–212
36. Avison D, Lau F, Myers M, Nielsen PA (1999) Action research. *Commun ACM* 42(1):94–97
37. Checkland P, Holwell S (1995) Business processes: modelling and analysis for re-engineering and improvement. Wiley, Chichester
38. Baskerville RL (1999) Investigating information systems with action research. *Commun AIS* 2(3):4
39. Gordijn J, de Bruin H, Akkermans JM (2001) Scenario methods for viewpoint integration in e-business requirements engineering. In: Sprague RH Jr (ed) Proceedings of the 34th Hawaii international conference on system sciences (HICSS-34). IEEE CS Press, Los Alamitos, CA. Also available from <http://www.cs.vu.nl/~gordijn/>
40. Fox MS, Gruninger M (1998) Enterprise modelling. *AI Mag* 19(3):109–121
41. Geerts G, McCarthy WE (1999) An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intell Syst App July/August*:89–94

42. McCarthy WE (1982) The REA accounting model: a generalized framework for accounting systems in a shared data environment. *Account Rev* 554–578
43. Baida Z, de Bruin H, Gordijn J (2003) Business cases assessment: from business value to system feasibility. *Int J Web Eng Technol* (in press)
44. de Bruin H, van Vliet JC (2001) Scenario based generation and evaluation of software architectures. In: Bosch J (ed) *Proceedings of the third international conference on generative and component-based software engineering (GCSE 2001)*. Lecture notes in computer science, vol 2186. Springer, Berlin Heidelberg New York, pp 128–139
45. Fowler M (1997) *Analysis patterns*. Addison-Wesley Longman, Reading, MA
46. Gamma E, Helm R, Johnson R, Vlissides J (1997) *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman, Reading, MA
47. BusMod consortium (2001) BusMod Project NNE5-2001-00256: business models in a world characterised by distributed generation. Annex I: Description of work. See also <http://bus-mod.e3value.com>
48. Obelix consortium (2001) Obelix Project IST-2001-33144: Ontology-Based ELectionic Integration of CompleX Products and Value Chains: Annex I: Description of work. See also <http://obelix.e3value.com>