e^3-*value* : Design and Evaluation of e-Business Models

Jaap Gordijn[1,2] **and Hans Akkermans**[1,3]

[1] Vrije Universiteit Amsterdam, De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands (email: gordijn@cs.vu.nl)
[2] Cisco Systems, International Internet Business Solutions Group
[3] AKMC Knowledge Management BV, The Netherlands

Submission IEEE Intelligent Systems, special issue on intelligent e-business, April 2001

The purpose of e-business models is to show why an e-business case works from an economic perspective. e-Business models are also important to frame the business requirements that must be satisfied by IT systems and networks. They are usually represented by a mixture of very informal textual, verbal and ad-hoc graphical representations. This limits a clear understanding of the e-business issues that confront the various stakeholders, and maintains the gap between business executives, the main e-business model stakeholders, and the IT developers who deliver the information systems that operationalize the e-business model.

We discuss a more rigorous, conceptual modelling approach called e^3-*value*^{TM} for representing e-business models. A key aspect is that it is *value-based*, since an e-business model defines how economic value is created and exchanged within a network of actors. Our methodology is based on a generic value-oriented ontology specifying what's in an e-business model. It furthermore offers a lightweight graphical approach that guides the definition, derivation, and analysis of the multi-enterprise relationships, e-business scenarios and operations requirements, in both qualitative and quantitative ways. Illustrated by experiences from e-business definition and roll-out projects, we show that our e^3-*value* approach offers important benefits: (1) better communication of and decision making about the essentials of an e-business model by the stakeholders; (2) an upfront, sharper and more complete, understanding of e-business operations and requirements through scenario analysis and quantification.

1 Discovering and Clarifying e-Business Requirements

Successful e-business information introduce *innovative* ways of doing business at their time of introduction. Such a novel way of doing business is called the *e-business model*, and because of its innovativeness it confronts stakeholders with new and unsolved issues. Consequently, the e-business model represents an important design problem in e-business projects, and serves as a first step in requirements analysis.

Currently, however, the *e-business model* concept is overused and has many interpretations. In most cases, an e-business model is represented by a very rough, textual, outline of the service to be delivered. This vague notion of an e-business model results in time-consuming (and often mis-) communications between the stakeholders involved, not in the least between business executives and IT function representatives. Its lack of precision also leaves many business operations and IT issues undiscovered and undiscussed, increasing the risk of failure of new e-business models.

Conceptual modelling and analysis techniques are helpful to improve this situation. In order to design and to achieve a common understanding of an e-business model, the e^3-*value* approach contains a limited set of interrelated core concepts (an ontology) representing the essential components of any e-business model, together with a number of graphical modelling constructs for

Requirement viewpoint	Stakeholders involved	e-Requirement viewpoint *focuss*	e-Requirement viewpoint *representation*
Business *value* viewpoint	C*O's Marketeers Customers	*Values*, actors, exchanges	**???**
Business *process* viewpoint	Tactical Marketeer, Operational Management	*Processes*, workers, information, good, and control flows	UML - Activity diagrams - Sequence diagrams - Interaction diagrams High Level Petri Nets
System architecture viewpoint	IT-department	*Hard/software* components, data and control flows, code organization, ...	UML - Class diagrams - State transition diagrams - Sequence diagrams - Interaction diagrams - Deployment diagrams Architecture Description Languages

Fig. 1. For the development of e-business information systems three distinct viewpoints are important: (1) the **value** viewpoint, representing the way economic value is created, exchanged and consumed in a multi-actor network, (2) the **process** viewpoint, representing the operationalization of the value viewpoint in terms of business processes, and (3) the **system architecture** viewpoint, representing the information systems that enable and support e-business processes. For the process- and system architecture viewpoints, useable representation techniques are available, but for the business value point such techniques are lacking. An e-business model provides a starting point to frame and provide inputs for process- and system architecture requirements.

the definition and analysis of e-business scenarios. A unique feature of our approach is a focus on the concept of economic value as a central modelling construct. We generally define an e-business model as a conceptual model that shows how objects of *value* are created, exchanged, and consumed within a multi-actor network.

In our opinion, it is important to distinguish between different viewpoints relevant to e-business design: (i) the articulation of the economic value proposition, i.e. the e-business model proper; (ii) the lay-out of business processes that operationalize the e-business model; (iii) the IT systems architecture enabling and supporting the e-business processes (see Fig. 1). These viewpoints provide a separation of concerns, and thus help to manage the complexity, regarding different requirements and design aspects of e-business. There are several good ways to represent business process and architectural models, but corresponding techniques to express and analyze the value viewpoint are lacking. This is what we focus on in our current work.

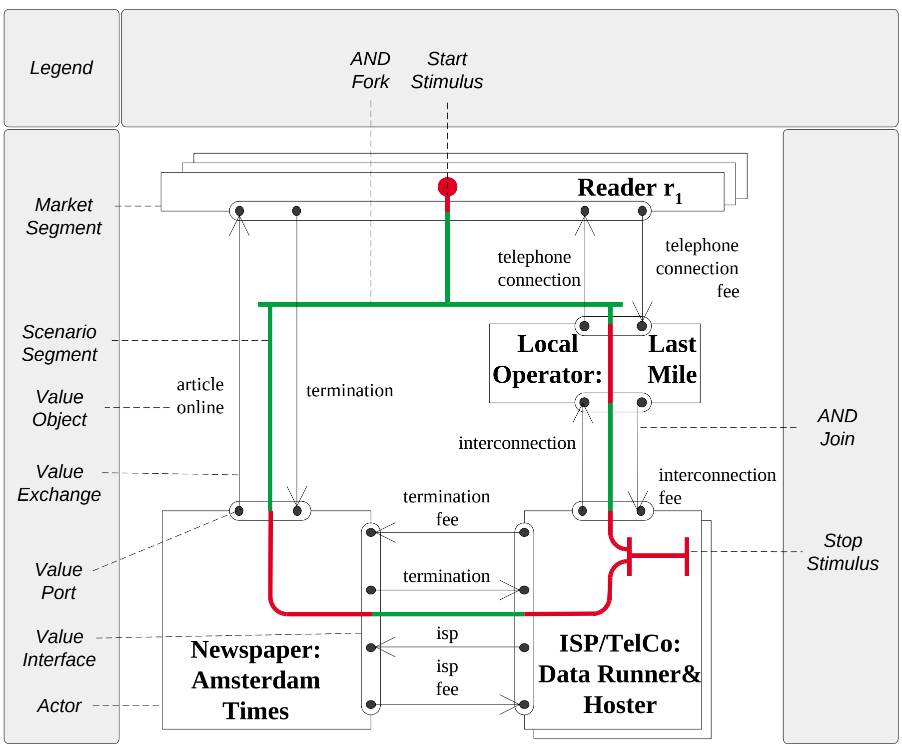

Fig. 2. The *terminating* e-business value model. The reader pays the local operator a fee for a telephone connection. This fee is used to pay an Internet Service Provider and another TelCo, a long distance carrier, and finally the newspaper. The local operator sets the price. The legenda of the graphical constructs is shown in separate boxes; these constructs are graphical representations of concepts from our e^3-*value* ontology for e-business models.

We illustrate the use of the e^3-*value* approach with one of the e-business projects where we successfully applied our approach. The e-business application is about the provisioning of a value-added news service. A regular newspaper called (say) the *Amsterdam Times* wants to offer to all its subscribers the service to read articles online using the Internet, but with making hardly any costs. Therefore, the business idea is to finance this by the telephone connection revenues, which are paid by the reader who has to set up a telephone connection for Internet connectivity. We note however that this can be achieved by two very different e-business models: the *'terminating'* model and the *'originating'* model. These models are represented in Figs. 2 and 3, following our graphical modelling constructs (see Legenda) which are directly linked to the e^3-*value* ontology discussed below.

Experience has shown that many features and implications of these e-business models were not easy to discover during the project without the help of our model representations. In the terminating model (Fig. 2), a telecommunication company sets the tariff for the service delivered, and money flows from the end customer to the Telco(s) and finally to the newspaper, whereas the originating model (see Fig. 3) assumes that the newspaper can set the prices, and money flows from the end customer to the newspaper, and from there to the Telco(s).

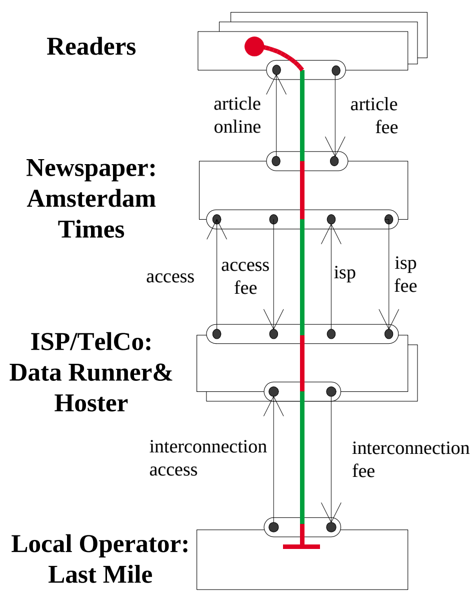

Fig. 3. The *originating* e-business value model. The reader pays the newspaper directly, who pays the ISP and long distance carrier, who in turn pays the local loop provider. The newspaper sets the price.

As we will discuss in more detail, we are capable to represent the heart-beat of an e-business value model with just a few pictures, which can easily be communicated to stakeholders and have a clear meaning. Moreover, our techniques allow for e-valuation of e-business models, by simple spreadsheet-like profit and utility calculations for *all* actors involved [7].

2 Value as the Basis for e-Business Modelling

To represent an e-business model, we have developed a lightweight ontology (for more information see [6]), and on top of that we utilize a well-known lightweight scenario technique, called Use Case Maps [2]. A lightweight approach that can also be graphically expressed is important here, as it allows us to communicate our ontology easily to intended users such as business consultants and CxO's. Moreover, the agility of e-business projects (the need to define, explore, and execute a business idea fast [8]) asks for a lightweight approach. We now discuss the ontological concepts (outlined in Fig. 4) and the UCM scenario concepts in brief. An important difference between the e^3-*value* ontology and other business ontologies such as the Toronto TOVE [4] and Edinburgh Enterprise ontologies [10], is that the latter focus on the business process rather than the value exchange aspects (distinguished as separate viewpoints in Fig. 1).

2.1 The e^3-*value* ontology for e-business models

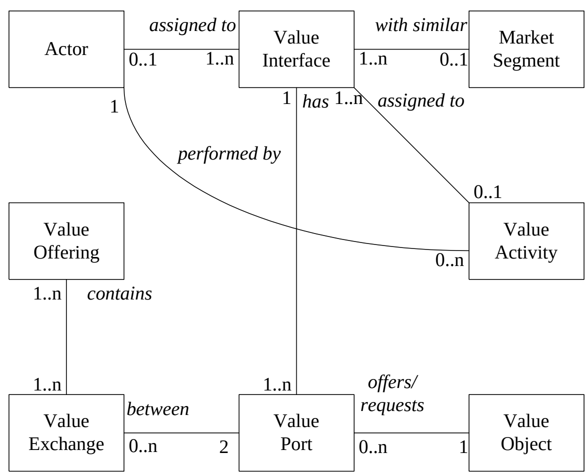

Fig. 4. Generic concepts and relationships underlying an e-business model. Actors produce, distribute or consume objects of *value* by performing value activities. The associated input and output objects are exchanged via value interfaces of actors or activities, denoting that if a value object is offered to the environment, some other value object is requested in return. Value interfaces have value ports offering or requesting objects of value. An analogy may be seen with electrical wall outlets that define the nature of the interface to the external environment, while at the same time hiding the internal way of working. A potential trade of value objects is represented by value exchanges, which interconnect value ports of actors or value interfaces.

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. By carrying out *value activities* an actor makes a profit or increases its utility. In a sound, viable, e-business model *each* actor should be capable of making a profit. *Amsterdam Times* is an example of an actor (see Fig. 2).

Value Object. Actors exchange value objects. A value object is a service, product, money, or even a consumer experience. The important point here is that a value object is *of value* for one or more actors. In most cases, we are interested in the *kind of* value objects actors exchange, and not so much in the actual instances themselves (except for profit and utility increase analysis). An *article online* is an example of a value object, because it is supposed to be of value for the *reader*.
Value Port. An actor uses a value port to show to its environment that it wants to provide or request value object instances. The concept of port is important, because it enables to abstract away from the internal business processes, and to focus only on how external actors and other components of the e-business value model can be 'plugged in'. This is the value analog of the separate external interfaces familiar from technical systems theory [1]. Value ports are represented as small black circles.
Value Interface. Actors have one or more value interfaces. A value interface groups individual value ports. (One can see this as a direct analogon to how a wall outlet is an assembly of plug-in ports in a technical system). It shows the value object an actor is willing to exchange *in return for* another value object via its ports. Such willingness is expressed by a decision function on the value interfaces, which expresses under what conditions an actor wants to exchange a value object for another value object. The exchange of value object instances is atomic at the level of the value interface. Either *all* exchanges occur as specified by the value interface or *none* at all. The value interface says nothing about the time ordering of object instances to be exchanged on its ports. It simply states which value object is available, in return for some other value object.

As an example, the *reader* has a value interface with ports, which says that s/he wants to give its environment a fee for a telephone connection and a *termination*, but wants an article online and a telephone connection in return for that (see Fig. 2). Most of these ports exchange understandable value objects, but *termination* is rather tricky. In the vocabulary of the telecommunications industry, telephone calls have to be terminated. This means, that if someone tries to set up a telephone connection by dialing a telephone number, another actor must pick up the phone, that is, *terminate* the connection as it is called by telephone operators. Two actors are needed for a telephone 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, because telephone connections generate revenues for operators by utilizing their infrastructure. Consequently, the *reader* offers something of value to the *Amsterdam Times*: the ability to cause a termination of a connection. *Amsterdam Times* resells this termination opportunity to a telecommunications operator. This way, the *Amsterdam Times* utilizes its existing, large subscriber database on the normal newspaper, to create a large volume of terminations of telephone calls on the network of a telecommunication operator, by offering its subscribers access to an online article database.
Value Exchange. A value exchange is used to connect two value ports with each other. Depending on the context of the ports that a value exchange connects (whether the ports connect an actor or a value activity), slightly different interpretations and constraints apply. Here, we only address the *actor-actor* connection.

An actor-actor value exchange represents one or more *potential* trades of value object instances between value ports. As such, it is a prototype for actual trades between actors. The Enterprise Ontology [10] would call a value exchange a potential sale. It shows which actors are willing to exchange value object instances with each other. So, it does not model *actual* exchanges of value object instances, which we call *value exchange instances*.
Value Offering. A value offering is a set of value exchanges. It shows which value objects are exchanged via value exchanges *in return for* other value objects, and it is the prototype for *value offering instances*. If a value offering instance occurs, *all* value exchange instances should occur, or *none* at all. A value offering should obey the semantics of the connected value interfaces: values are exchanged via a value interface on *all* its ports, or *none* at all. Consequently, the four value exchanges between *reader*, *Amsterdam Times*, and *The Last Mile* are all part of *one* offering,

because the value interface of the *reader* prescribes that either *all* these exchanges should occur, or *none* at all.

Market segment. In the marketing literature [9], a market segment is defined as a concept that breaks a market (consisting of actors) into segments that share common properties. Accordingly, our concept *market segment* shows a set of actors that for value interfaces share the same decision function. We realize that in practice all actors behave different and consequently cannot have equal decision functions. However, to be able to design understandable e-business value models, we *assume* that some groups of actors have decision functions which are, from a modeling viewpoint, the same. Note that actors who are in a segment may also have differring value interfaces, because it is the actor-value interface combinations which build up a market segment. A *reader* is an example of a market segment.

Composite actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. However, for providing a particular service, a number of actors may decide to work together, and to offer objects of value jointly to their environment. Such actors decide on one or more common value interfaces to their environment. We call such a *virtual enterprise* group of actors a composite actor.

Note that both composite actors and market segments internally consist of multiple actors. A composite actor shows a *common/shared* value interface of its internal actors, while in contrast a market segment is seen by its environment as a set of fully *independent* actors, for which we assume the same decision function for a particular set of value interfaces.

As an example, *Data Runner* and *Hoster* are together a composite actor (see Fig. 5), because they have one value interface (see Fig. 5) to their environment, and offer on that interface a complete solution for internet access and hosting, an *ISP out of the box*.

Value Activity. A value activity is *performed by* an actor and increases profit or utility *for* such an actor. The value activity is included in the ontology to discuss and design the *assignment* of value activities to actors. As such, we are interested in the collection of activities which can be assigned as a whole to actors, and as a consequence, such an activity should be profitable or increase utility to be interesting to perform. Value activities can be decomposed into smaller value activities, but these still should be profitable or increase utility for the performing actor. This gives a decomposition stop rule, which is by the way clearly different from business process or work flow decomposition. *Provide Internet access* is an example of a profitable value activity, while *Read Article* is a utility increasing activity for the *reader* (see Fig. 5).

Note that the value exchange construct is also used to connect value ports of value activities (a value activity-value activity value exchange), and to connect value ports of value activities with actors (an actor-value activity value exchange).

2.2 Constructing e-business scenarios: use case maps

Scenario path. A scenario path consists of one or more segments, related by connection elements and start- and stop stimuli. From a mathematical point of view, it is an a-cyclic directed graph. It represents 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. Thus a scenario path shows causal relations between value interfaces.

Stimulus. A scenario path starts with a **start stimulus**. A start stimulus represents an event, possibly caused by an actor. If an actor causes an event, the start stimulus is drawn within the box representing the actor. The last segment(s) of a scenario path is connected to a **stop stimulus**. A stop stimulus indicates that the scenario path ends. An example of a start stimulus is the desire to read an article online by a *reader*.

Segment. A scenario path has one or more segments. Segments are used to relate value interfaces with each other, possibly via connection elements, to show that an exchange on one value interface causes an exchange on another value interface. Using connection elements, sophisticated causal relations can be represented.

Connection. Connections are used to relate individual segments. An **AND fork** splits a scenario path into two or more sub path, while the **AND join** collapses sub path into one path. An **OR fork** models a continuation of the scenario path into one direction, to be chosen from a number of alternatives. The **OR join** merges two or path into on path. Finally, the **direct** connection interconnects two individual segments.

Connection elements can be bounded to value activities, and thereby to actors. This models the assignment of a decision task (splitting, synchronizing, choosing) to an actor.

A scenario path must obey the semantics of the value interface connected by value exchanges: either objects are exchanged on *all* its ports or *none* at all. To illustrate this, these parts of the scenario path are green colored, while other scenario segments are red colored. The AND-fork (Fig. 5) between the *reader*, *Last Mile*, and *Amsterdam Times* is thus important: it ensures, that if values are exchanged via the *readers*' value interface, values must be exchanged via the value interfaces and of the *Amsterdam Times* and the *Last Mile* respectively. The *Last Mile* is a telecommunication operator who handles the local loop: the last mile of communication between a telephone switch and the home of a subscriber.

Responsibility points and profit sheets. Scenario segments may hit value interfaces. Such a hit is called a responsibility point. By following a scenario path and by finding the responsibility points along the path, we can construct what we call a 'profit sheet'. The sheet shows which objects of value are entering and leaving the actor and/or value activity (depending on the level of interest), as a result of executing the scenario. By valuing the objects in this sheet, and by making assumptions on the number of executions per time frame of the scenario path, we obtain a basic idea about the profitability or utility increase for a specific actor.

3 Analyzing the Implications of Alternative e-Business Models

This concludes our discussion of what an e-business model looks like in terms of generic value-based concepts and scenarios. These are instantiated for specific e-business cases and then serve as the basis to analyze the characteristics and implications of alternative e-business models. For the e-business news service application we discuss this for more detailed versions of (1) the terminating model (see Fig. 5), and (2) the originating model (see Fig. 6).

3.1 The 'terminating' e-business model

Causality of revenue streams. The most obvious observation, which can made by looking at a conceptual e-business value model, is the causality of revenue streams, in reaction to a stimulus event. Simply by following the scenario path, it can be seen which exchanges of values via value interfaces result in other value exchanges. In the terminating e-business value model, it is important to recognize that initially, the money flow is between the *reader*, and *Last Mile*, who is only responsible for the local loop data traffic. All other money flows are generated from the money earned by *Last Mile*.
Bundling. An article online is not of value for the reader without a telephone connection, and thus, *article online*, *telephone connection*, *termination*, and *telephone connection fee* are bundled

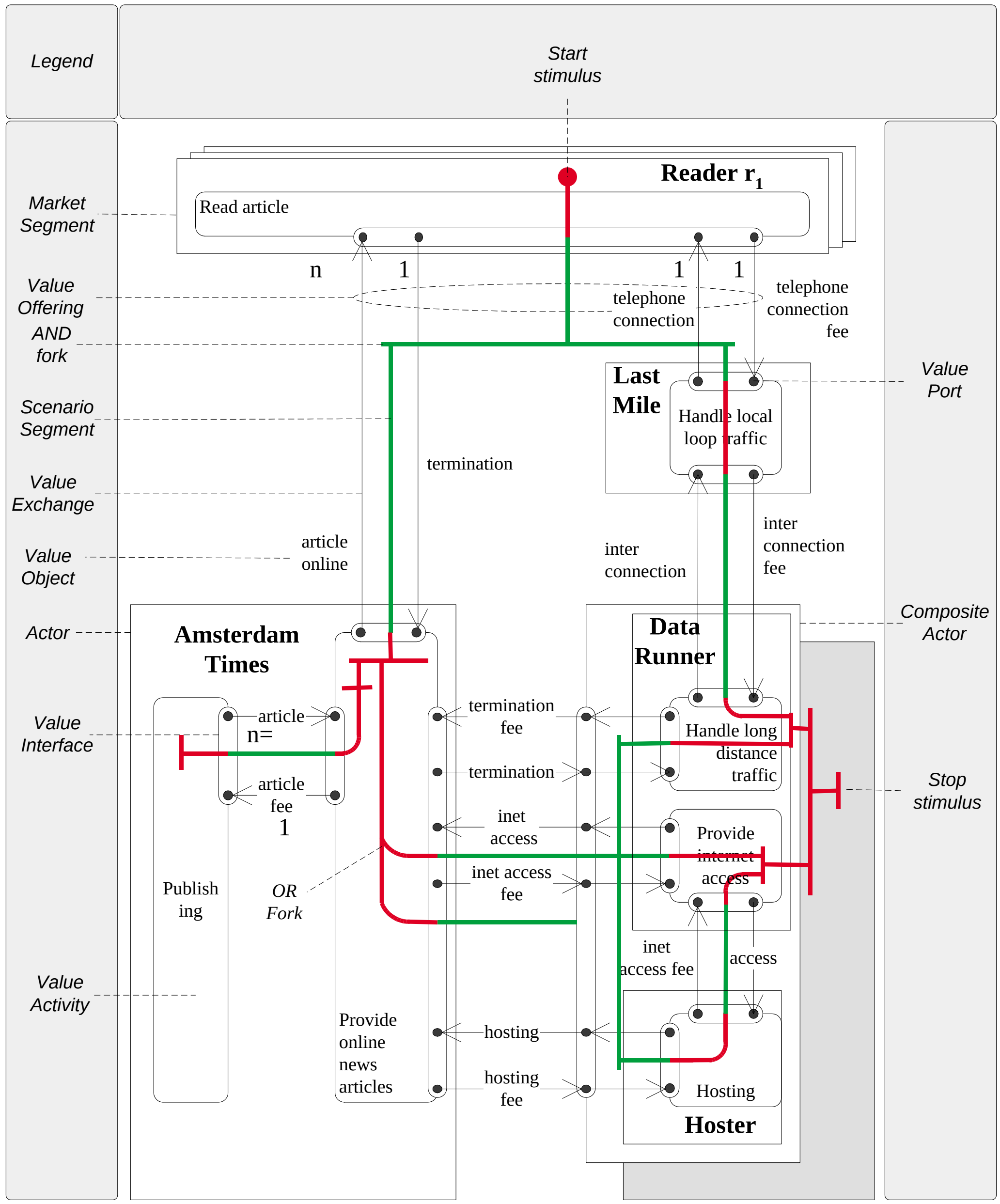

Fig. 5. The *terminating* e-business model in a more detailed version: *Data Runner* and *Hoster* form a consortium to offer long distance traffic and internet service provisioning to *Amsterdam Times*. There is another consortium doing the same. *Amsterdam Times* selects, on a per scenario occurrence basis, which consortium handles the data traffic.

from a reader perspective. This is represented in Fig. 5 by grouping the value ports associated with the value objects into one value interface. Bundling/unbundling in general is a key e-business mechanism especially for digital products and for services, that is clearly represented by our methodology.

Customer ownership shift to *Last Mile*. Originally, the reader was a full customer of *Amsterdam Times*, because the reader is part of *Amsterdam Times* regular subscriber database. However, for the online service, as can be seen from Fig. 5, the reader now has to exchange values with *Amsterdam Times, and Last Mile*. The latter is the party that receives the only payment for the delivered service. This can be seen as a shift in customer ownership from *Amsterdam Times* to *Last Mile*, which is an undesirable situation from *Amsterdam Times*' point of view.

Price setting: *Last Mile* is in control. As can be seen, the *reader* pays for the *entire* telephone connection to the *Last Mile*. Unfortunately, no one, except *Last Mile* and perhaps a market regulation authority, can influence the pricing. Consequently, the success of the e-business model depends largely on *Last Mile*.

Power: No choice for local loop. The *reader must* use the *Last Mile* for local loop access. At the time the project was carried out, there was only one actor available controlling the local loop to subscribers. This can be concluded from Fig. 5, because only one actor for local loop access is drawn. Again, this makes the e-business model very critical to the behavior of *Last Mile*.

Duplication of assets against nearly zero marginal cost. An important property of digital assets, such as news articles, is the ability to reproduce the asset against nearly zero marginal cost [3]. The *Amsterdam Times*, and especially the value activity *Provide Online News Articles*, buys an article from another value activity *Publishing* (the activity necessary to produce articles for a regular newspaper anyway) if it is requested by the *reader*. However, it only buys this article once, so if multiple readers ask for the same article, the *Provide online news articles* only pays once the *Publishing* activity. Fig. 5 shows this by the cardinalities on the value exchanges near the value interface of value activity *Publishing*: for *n* equal articles, only *one* fee has to be paid.

Partnership: composition of actors. Fig. 5 shows two partnerships (or composite actors): actors who decide to have a common value interface to their environment. These two partnerships are equivalent in a way that they offer comparable objects of value to their environment. Each partnership consists of a number of other actors. For the topmost partnership, these are *Data Runner*, a telecommunication company and *Hoster*, an internet service provider. Both these companies decide to offer hosting and internet access jointly as a bundle, under certain special conditions. A special condition can be the price, which might be cheaper for *Amsterdam Times* than an alternative, such as obtaining the objects of value from other actors separately. In this specific case, *Data Runner* and *Hoster* can offer services jointly cheaper, because they co-locate technical equipment such as a telephone switch, internet access servers, and web servers at one physical site, thus saving costly wide area connections to interconnect all these components.

Figure 5 shows two of such partnerships who offer comparable services. Note that these two partnerships are not a market segment, because their decision function may differ. For instance, the first partnership may offer the same services (access, hosting) for lower prices than the second partnership. The second partnership is shown as a gray box, to prevent unnecessary cluttering of the diagram.

Power: service selection. This business idea has a special 'trick' to enlarge the power of *Amsterdam Times* with respect to the telecommunication and internet service providers. The *Amsterdam Times* can choose from two different composite actors to actually offer the article online (from an access and hosting perspective), and this selection can be done on a per scenario execution base. The reason for this is that the *Amsterdam Times* does not want to be dependent on one provider for access and hosting. By distributing the amount of traffic over these two (composite) providers, the *Amsterdam Times* controls the distribution of revenues for the two composite actors, and motivates both partnerships to deliver a high level quality of service. This is graphically shown using an OR-fork in the scenario path, which models the service selection by *Amsterdam Times*.

3.2 The 'originating' e-business model

Causality of revenue streams. In the originating e-business model (see Fig. 6), the causality of revenues has been reversed, compared to the terminating e-business model. A start stimulus by the *reader* now causes value exchanges at the interface of the *Amsterdam Times*, causing the need to buy-in internet access and hosting, and in turn local loop access. This can easily be seen from figure 6, simply by following the scenario path.
Customer ownership. Figure 6 shows an e-business model founded on other grounds than the terminating e-business model. The *reader* as the end customer only 'sees' the *Amsterdam Times*, and not the *Last Mile* anymore for reading an article online. Also, the *reader* pays the *Amsterdam Times* directly for everything needed to read an article online. Because *reader* pays to the same party that delivers the service, there is no shift of customer ownership.
Price setting. In this model, the *Amsterdam Times* controls the price of the *article online* service itself by being able to set the telephone costs for the *reader*. It can even decide for the, unlikely, case to *pay* the *reader* for reading articles online, a situation which is impossible for the terminating e-business model.
Assignment of activities to actors. In the originating business model there is a change in who is doing what: the *Data Runner* actor now takes care of physical and internet access provisioning, and of hosting. We can easily shift these activities amongst actors, assuming that these activities are profitable for the performing actor.

In sum, many important e-business features and implications can be directly derived from semi-formal graphical representations of e-business models as we propose.

4 E-valuating e-Business Models

The next step is to e-valuate the economic feasibility of an e-business idea in quantitative terms, based on an assessment of the value of objects for all actors involved. Feasibility of an e-business model means that *all* actors involved are able to make a profit or to increase their utility with an e-business idea. Again, our technique for determining feasibility is a lightweight approach, and focuses on building *confidence* that an e-business idea is of real interest for all actors involved, rather than offering a precise calculation of all profits (to expect the latter is simply unrealistic). Our approach is to take into account the net in- and out-flows of value objects, because the net value of these flows should be sufficient to cover all other expenses. An additional confidence-building step is to analyze *what-if* scenarios: they help to understand the sensitivity of e-business models with respect to financial parameters, future trends, and other parameters such as customer behaviors, and find the weak and strong points of e-business models.

Our e-valuation approach consists of the following steps:

1. Creation of profit sheets for all actors, or value activities, in the e-business model.
2. Valuation of the value object in the profits sheet by the actors.
3. E-valuation of *what-if* scenarios.

Creation of a profit sheet. Profit sheets are created on the actor or value activity level. The actor level is of interest to create confidence in an e-business idea, while profits sheets on the value activity level are useful to evaluate whether activities are really profitable. Due to lack of

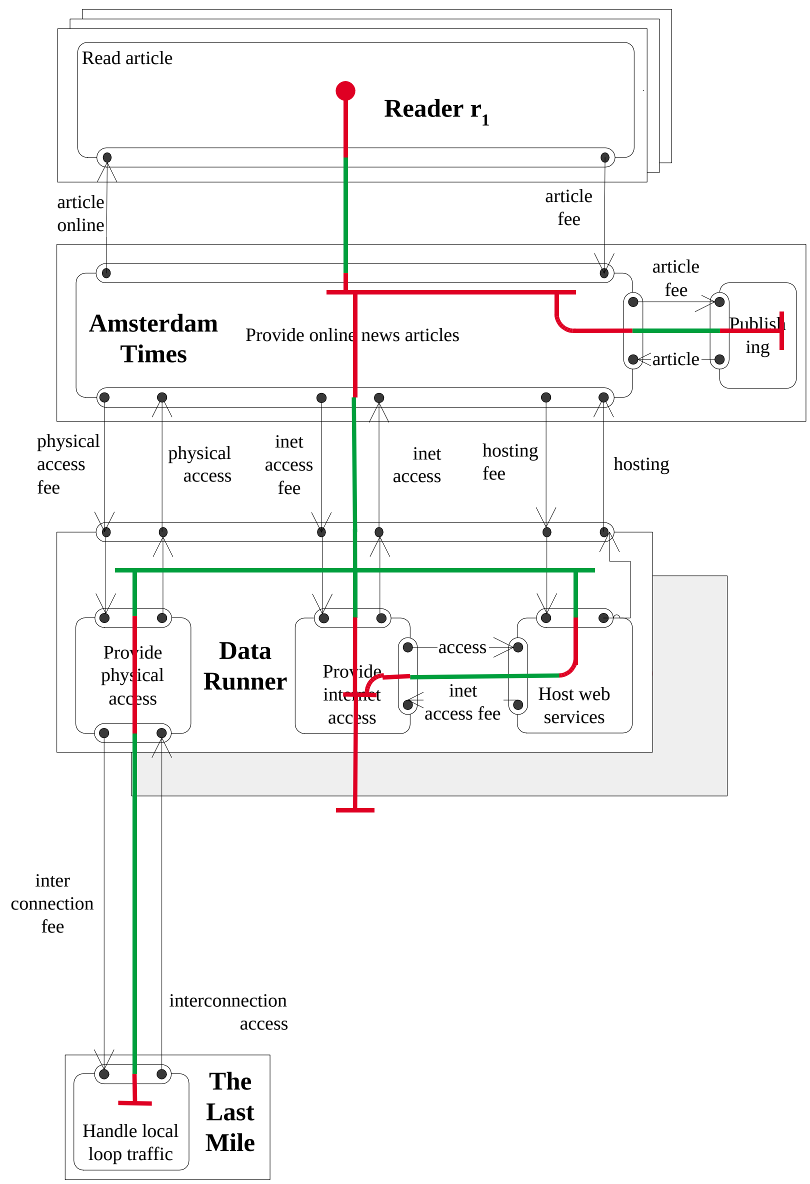

Fig. 6. The *originating* e-business model in an enriched version: the newspaper sets the price of telephone ticks and gets paid accordingly. Also, the *Data Runner* now offers both the handling of long distance traffic and internet service provisioning.

space, we concentrate on the actor profit sheets, and only for the terminating business idea, but the same approach can be used to develop profit sheets for value activities and other business ideas. Also, we leave out the utility-increase analysis for the *reader* actor (see for a discussion on end-customer utility analysis using our approach [5]).

Table 1 shows a reduced profit sheet for the scenario *read article online*. We create this sheet by following the scenario path, starting at the start stimulus, and each time the path crosses a value interface of an actor, the sheet is updated with value objects flowing in and out of the actor. In sheet 1, we have two composite actors. We show for composite actor 1 the profit sheets of its actors (*Data Runner* and *Hoster*). For brevity, table 1 does not account for *composite 2*; we deal with *composite 2* equally as we do for *composite 1*.

For comprehensibility, we *reduce* the profit sheet by removing for each actor all value objects, which are (1) not money streams, *and* (2) which are entering the actor and leaving the actor (possibly in an enriched form, as the result of performing an activity) on the same scenario path. For example, we remove *telephone connection* and *interconnection* from the actor *Last Mile*, because the *telephone connection* is an enriched *interconnection*. *Last Mile* enriches the *interconnection* by exploiting a district telephone switch and a list mile of copper or fibre optics.

Table 1. Profit sheet for the scenario *Read article online*, for the terminating e-business value model.

Scenario	Read article online	
Actor	*Value Object In*	*Value Object Out*
Last Mile	$\textit{tel. connect. fees} = (\textit{tel. start tariff} + (\textit{tel. connect. tariff} * \textit{duration})) * \textit{actual occ.}$	$\textit{interconnect. fees}_{composite1} = \textit{tel. connect. fee} * \textit{distance factor}_{composite1} * \textit{interconn. factor} * \textit{actual occ.} * p$
Amsterdam Times	$\textit{termination fees}_{composite1} = \textit{tel. connect. fee} * \textit{revenue sharing factor}_{composite1} * \textit{distance factor} * \textit{actual occ.} * p$	$\textit{inet access fees}_{composite1} =$ *see Data Runner* $\textit{hosting fee}_{composite1} =$ *see Hoster*
Composite 1		
Data Runner	$\textit{interconnect. fees} =$ *see Last Mile* $\textit{inet access fee}_{AmsterdamTimes} = \textit{inet. connect. tariff} * \textit{duration} * \textit{actual-forecast occ.} * p$ $\textit{inet access fee}_{Hoster} = \textit{occurences}_{forecast} * p \rightarrow$ *needed bandwidth* $\rightarrow$ *fixed fee/month*	$\textit{termination fees} =$ *see Amsterdam Times*
Hoster	$\textit{hosting fee} = \textit{concurrent occ.}_{forecast} * p \rightarrow$ *concurrent pageviews* $\rightarrow$ *fixed fee/month*	$\textit{inet access fee} =$ *see Data Runner*
Composite 2	...	...

Valuation of objects. Value objects in the profit sheet have to be assigned a value, expressed in monetary units (e.g. Euros or dollars). We explain this valuation briefly:

The telephone connection fee per scenario occurrence is based on a start tariff and a connection-time dependent tariff. To calculate the total monthly fees, the telephone connection fee is multiplied with the actual (realized) number of scenario occurrences.
The interconnection fee per scenario occurrence (here only shown for actors in *composite 1*) is based on a fraction (the interconnection factor) of the telephone connection fee, and on the physical distance *Data Runner* bridges.
The termination fee *Amsterdam Times* receives, in this case from *composite 1*, is calculated analogously to the termination 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. 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 *Data Runner* takes the risk of a decreasing interconnection factor.
Data Runner charges *Amsterdam Times* an internet access fee in return for giving *readers* access. This fee is based on an access tariff per second. We want to account for the situation that internet access equipment is a very scarce resource; *Data Runner* wants to have the opportunity to assign unused access ports to others. Therefore, *Amsterdam Times* is asked to forecast the number of scenario occurrences on a monthly basis, including the average duration. *Data Runner* then allocates access ports on this forecast, and can allocate remaining ports to others. To motivate *Amsterdam Times* for good forecasting, the following valuation is used: If the actual scenario occurrences drop below 75 % of the forecasted occurrences, we use 75 % of the forecasted occurrences for the calculation of the monthly internet access fee. Otherwise, we use the actual, realized number of scenario occurrences.
The internet access fee to be paid by *Hoster* is entering based on the forecasted number of scenario occurences. Based on this, we calculate an estimate of the required bandwidth, and the price for this.
Hoster uses a forecast of *Amsterdam Times* of the number of concurrent page views, which in turn is based on an average number of page views per forecasted scenario occurrence. This results in a fixed fee per month for hosting.

E-valuation of *what-if* scenarios. Using the valuation in Table 1, we e-valuate several scenarios, which model expected changes in the future regarding valuation. Important assumptions are shown in Table 2. As an example, Table 3 shows a number of scenarios.

The *null* scenario refers to the present situation. 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 *Amsterdam Times* will not make a profit. For *Last Mile* and *Data Runner* there is still a profit to cover the costs. *Hoster* is insensitive to bad forecasts, because it does not depend on the number of actual scenario occurrences.
It is reasonable to expect a decrease in the interconnection factor is reasonable 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. *Amsterdam Times* does not feel such a decrease, but *Data Runner* will.
Data Runner may decide to decrease its revenue sharing factor. As can be seen, this will harm *Amsterdam Times*.

In conclusion, by valuing the objects for each actor, and by making reasonable assumptions about the number of (forecasted) scenario occurrences, we can perform a sensitivity analysis for

Table 2. Basic assumptions for the null-scenario, used to e-valuate the terminating e-business model.

scenario occurences/month = 1,500,000	service selection ratio = 0.5	internet access fee = 0.003 /minute
concurrent scen. occ. = 10,000	interconnection factor = 1	valuation composite 1 = valuation composite 2
concurrent page views = 10,000	revenue sharing factor = 0.55	prices for bandwidth/hosting using a ladder
average scenario duration = 480 s	distance factor = 0.8	bandwidth/user = 1024 bps
		forecase = actual

Table 3. Different valuation scenarios. The null-scenario uses the valuation in table 2. A second scenario assumes that *Amsterdam Times* forecasts inaccurately. A decrease in the interconnection is expected to occur, especially of competition between telecommunication actors increases (see the third case). The fourth scenario supposes a drop in the revenue sharing factor between *Data Runner* and *Amsterdam Times*.

	Profit			
Scenarios	*Amsterdam Times*	*Last Mile*	*Data Runner*	*Hoster*
Null-scenario	164,400	102,000	113,800	8,000
$Forecast(1,500,000) >> Actual(150,000)$	-28,560	10,200	26,680	8,000
Decrease in interconn. factor (1.0 to 0.4)	164,400	346,800	-8,600	8,000
Decrease in revenue sharing factor (0.5 to 0.1)	-19,200	102,000	205,600	8,000

the business idea hand. This sensitivity analysis is in many cases of more business interest than the numbers of the valuation itself.

5 Business Process and System Requirements

Thus, a conceptual e-business model increases understanding between stakeholders and shows the heartbeat of an e-business idea. Moreover, if we assume valuations by actors, we can construct what-if scenarios, and e-valuate whether an e-business model is profitable for all actors involved.

In addition, an e-business model also highlights important requirements for its supporting business processes and information systems. Below, we discuss a few examples of these:

For the case at hand, the tariff for hosting is based on the assumption that equipment such as servers can be hosted at the same physical location as the access points (the telephone switches). Costly data-connections between access points and servers are avoided in this way. Thus, the physical location of equipment is an important requirement, which is typically expressed by the system architecture viewpoint.

Furthermore, the e-business model stresses that *Amsterdam Times* only coordinates the facility for reading articles online, but completely outsources the entire operation from a technical perspective. Consequently, business processes implementing the *Provide online news articles*, should focus on

1. managing these outsourced processes, based on agreed service levels, and
2. ensuring that the subscriber database of *Amsterdam Times* on the regular newspaper is in sync with authentication databases of the actors who provision the actual online service.

Therefore, from a systems architecture point of view, a system should be in place that keeps the subscription database of *Amsterdam Times* and the providers in sync.

6 Conclusion

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

Non-trivial e-business ideas can be clearly represented using our e^3-*value* methodology. It has the capabilities to express and analyze many different general mechanisms that are important in e-business, including the causality of revenue streams, (un)bundling of value objects, customer ownership, the ability to set prices, the ability to choose alternative actors to deliver objects of value, and partnership issues.

Using well-known scenario techniques, we construct profit sheets, which in turn are used to get a clear view on the profitability of the business idea for each actor. These profit sheets should not be seen as absolute profit predictors, but rather as calculation schemes that enable a rational justification of, stakeholder confidence building in, and strength-weakness sensitivity analysis of e-business models.

In the application project setting, our approach turned out to be especially useful to articulate the e-business idea precisely, creating a common understanding amongst stakeholders. The business ideas, the terminating and originating models, appeared to be too complicated to communicate them in natural language.

Finally, the e-business model embodies decisions that impact other requirements, especially business process and information system requirements. Consequently, e-business modelling yields a framework for such requirements, and can be seen as a first step in requirements engineering for e-business information systems. The concept of value provides the crucial vantage point to understand innovative e-business models and their operationalization.

Acknowledgements. This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project nr VWI.4949. Also, we thank *De PersCombinatie* for their permission to publicize some e-business project experiences and results discussed in this paper.

References

1. W.N. Borst, J.M. Akkermans, and J.L. Top. Engineering ontologies. *International Journal of Human-Computer Studies*, 46:365–406, 1997.
2. R.J.A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.
3. Soon-Yong Choi, Dale O Stahl, and Andrew B. Whinston. *The economics of doing business in the electronic marketplace*. Macmillan Technical Publishing, Indianapolis, 1997.
4. M.S. Fox and M. Gruninger. Enterprise modelling. *AI Magazine*, pages 109–121, 1998.
5. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Selling bits: A matter of creating consumer value. In Kurt Bauknecht, Sanjay Kumar Madria, and Gunther Pernul, editors, *First International Conference on Electronic Commerce and Web Technologies EC-Web 2000, London, UK, 4-6 September*, volume LNCS 1875, pages 48–62, Berlin, 2000. Springer Verlag. Also available from http://www.cs.vu.nl/˜gordijn.
6. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. What's in an electronic business model. In R. Dieng and O. Corb, editors, *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference, EKAW 2000, Juan-les-Prins, France, October 2000*, volume LNAI 1937, pages 257–273. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.
7. J. Gordijn, H. de Bruin, and J.M. Akkermans. Scenario methods for viewpoint integration in e-business requirements engineering. In *Proceedings of the 34rd Hawaii International Conference On System Sciences (HICSS-34)*, pages CD–ROM. IEEE, January 3-6 2001. Also available from http://www.cs.vu.nl/˜gordijn.
8. Amir Hartman, John Sifonis, and John Kador. *Net Ready - Strategies for Success in the E-conomy*. McGraw-Hill, New York, 2000.
9. P. Kotler. *Marketing management: analysis, planning, implementation and control*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
10. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The knowledge engineering review*, 13, 1998.