e^3-*value*^{TM}: A Conceptual Value Modeling Approach for e-Business Development

Jaap Gordijn[13] **and Hans Akkermans**[12]

[1] Vrije Universiteit - Vuture.net - Centre for e-Business Research, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, {gordijn,hansa}@cs.vu.nl
[2] AKMC Knowledge Management, Klareweid 19, 1831 BV Koedijk, The Netherlands
[3] Cisco Systems - Internet Business Solutions Group, Hoogoorddreef 9, 1101 BA Amsterdam, The Netherlands

Abstract. e-Business models show why an e-business case might work from an economic perspective. They are often represented using informal textual representations. This inhibits a clear understanding of an e-business model by all stakeholders involved. We propose a more formal, conceptual approach for representing the business model, which we call a *value* model, to enhance the common understanding amongst stakeholders, but also to allow for an asssesment of an e-business model. We illustrate our modeling approach by a state of the art project in the free Internet arena.

1 Introduction

Successful e-business information systems often are characterized by *innovative* ways of doing business at their time of introduction. Such a way of doing business is called the *e-business model*, and because of its newness, an important design topic in e-business projects.

Currently the *e-business model* concept is overused and has many interpretations. In most cases, an e-business model is represented by a rough, textual, outline of the service to be delivered. This rather vague notion of a *business model* results in time-consuming- and mis-communications between stakeholders involved, not in the least between business oriented people, and IT people.

We propose to use a *conceptual modeling* approach with clearly defined modeling constructs, called e^3-*value* , to design and to reach common understanding about an e-business model. Our approach offers a number of interrelated core concepts, also called an ontology, which are be used to build a semi-formal conceptual e-business model. Our approach is unique because we focus on the concept of *economic value* as a central modeling construct. Consequently, we define an e-business model as a *conceptual, economic value* model, which shows objects of *value*, which are created, exchanged and consumed in a multi-actor network.

A conceptual value modeling approach has the following advantages. Firstly, it facilitates, by clearly articulating the value proposition, in reaching a better shared understanding and agreement between actors on a service to be offered, rather than using a ambiguous free text representation of the model. As we will show, we are capable to represent the heart-beat of a non-trivial e-business value model with just a few pictures, which can easily be communicated by stakeholders and have a clear meaning. Secondly, our technique allows for e-valuation of an e-business value model. E-valuation assesses whether the e-business model is profitable, or increases economic utility, for all stakeholders involved. The intention of e-valuation is not to give precise calculations about profit to expect, but more to build confidence in the commercial viability of the e-business model. To increase confidence, we exploit what-if scenarios which e-valuate an e-business model for expected changes (e.g. economic oriented) in the future. Such e-valutations, however, require a clearly articulated e-business model, rather than an ambigious business idea.

We illustrate the use of e^3-*value* with one of the e-business projects where we successfully applied our approach. The project at hand is about the provisioning of a value add news service. With respect to such a service, a regular newspaper called the *Amsterdam Times* wants to offer to all its subscribers on the regular newspaper the service to read articles online using the Internet. However, *Amsterdam Times* does not want to make any additinal costs for offering this online service. Therefore, we finance the execution of the business idea by telephone connection revenues, which are paid by the reader who has to set up a telephone connection for Internet connectivity.

In this article, we demonstrate how we used our e^3-*value* technique to explore this e-business value model, first by showing how we represent an e-business model, and by discussing what stakeholders can conclude from such a model, and second by illustrating e-valution as a confidence builder in the e-business idea at hand.

2 Representing an e-business value model

To represent an e-business value model, we use a lightweight ontology consisting of interrelated core concepts, and we utilize a well known lightweight scenario technique, called Use Case Maps [1]. A lightweight ontology contains a limited set of concepts and relations [7]. This 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 [6]) asks for a lightweight approach. For the same reasons, we use a light weight scenario technique as well. Below, we discuss the ontological concepts and the UCM scenario concepts briefly. More information can be found in [5], [1]. Throughout the text, we refer to an e-business value model example, presented in Fig. 1.

2.1 The e^3-*value* ontology

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. By doing *value activities* (see below) an actor makes profit or increases its utility. In a sound, viable, business value model *every* actor should be capable of making profit.

Example. Actors are represented as rectangles. For instance, *Last Mile* is a telecommunication company operating the local loop (the last mile from a telephone switch to an end-user).

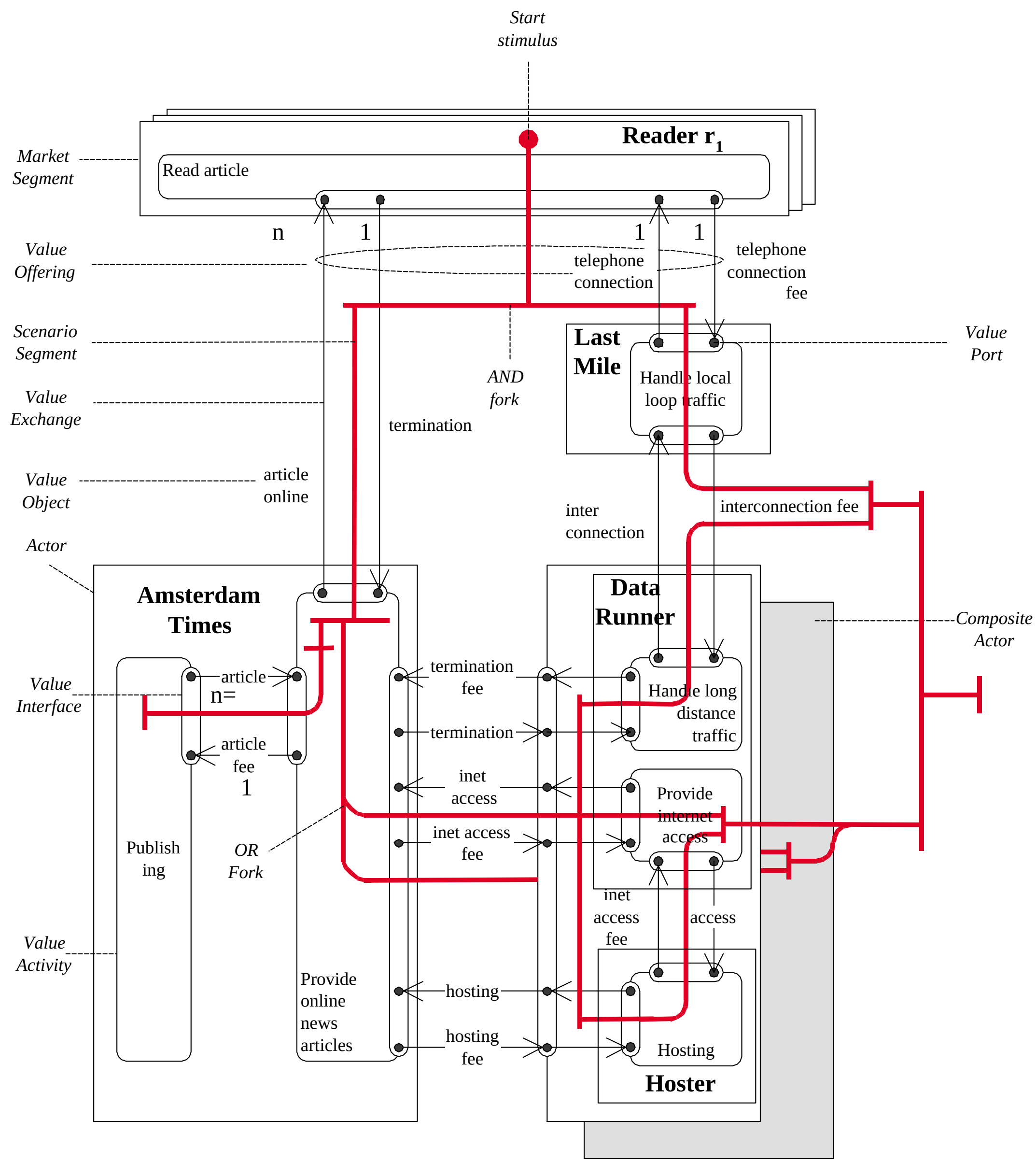

Fig. 1. An e-business value model for the newspaper case. The *reader* pays a local operator, *Last Mile*, a fee for a telephone connection. *Last Mile* uses a long distance carrier, *Data Runner* to set up the connection if the callee is located byond the geographical area *Last Mile* serves, and pays *Data Runner* for this (the interconnection fee). Because *Amsterdam Times* offers an article online service to its existing subscriber base, a large number of telephone calls are expected. Therefore, *Data Runner* is willing to pay *Amsterdam Times* a termination fee for the generated traffic. Furthermore, *Amsterdam Times* outsources the internet service provisioning (hosting and internet access) aspects of offering an article online and pays a fee for this. *Last Mile* and *Hoster* form a composite actor (a partnership) to offer hosting, access and call termination as one service to *Amsterdam Times*. There is a second composite actor (grey shaded) making the same offering also. *Amsterdam Times* can choose on a per scenario execution basis, which composite actor to use for service provisioning.

Value Object. Actors exchange value objects. A value object is a service, product or even a consumer experience. The important point here is that a value object is *of value* for one or more actors.

Example. Value objects are shows as text aside arrows (the value exchanges, see below). Most of the objects speak for themselves, but *Termination* as a value object needs further explanation. In the vocabulary of the telecommunication industry, telephone calls have to be terminated. This means, that if someone tries to set up a telephone connection, 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, telecommunication operators are willing to pay such an actor for that. *Amsterdam Times* causes these terminations by offering its existing, large subscriber base (on a regualar newspaper) an *article online* service, which requires a telephone and internet connection.

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 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'.

Example. Ports are shows as small black circles. For instance, *Amsterdam Times* has an outgoing port for *articles online*.

Value Interface. Actors have one or more value interfaces. A value interface groups individual value ports offering or requesting value objects. It shows the value object(s) an actor is willing to exchange *in return for* other value object(s). Such willingness is expressed by a decision function on the value interfaces, which shows on what conditions an actor wants to exchange a value object for another value object. The exchange of value objects 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 objects to be exchanged on its ports. It simply states which value object is available, in return for some another value object.

Example. A value interface is shown by a rounded box, connected to an actor. A *reader* has a value interface, 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.

Value Exchange. A value exchange is used to connect two value ports with each other. A value exchange represents one or more *potential* trades of value objects between value ports. As such, it is a prototype for actual trades between actors. The Enterprise Ontology [9] would call a value exchange a potential sale. It shows which actors are willing to exchange value objects with each other.

Example. A value exchange is shown by an arrow. Ports of *Amsterdam Times* and a *Reader* are connected by a value exchange, to exchange *articles online* and *terminations*.

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. If a value offering occurs, *all* value exchanges part of it should occur, or *none* at all. A value offering should obey the semantics of the connected value interfaces: that is values are exchanged via a value interface on *all* its ports, or *none* at all.

Example. The four value exchanges between *reader*, *Amsterdam Times*, and *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 [8], 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 share for value interfaces an equal decision function. Note that actors who are in a segment may also have in-similar value interfaces, because it is the actor-value interface combination, which build up a market segment.

Example. A market segment is shown by stacking actors graphically. 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 group of actors a composite actor.

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

Example. The topmost partnership consists of *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 1 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.

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 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 [3].

Example. Provide Internet access is an example of a profitable value activity, while *Read Article* is a utility increasing activity for the *reader*.

2.2 Use Case Maps

Scenario path. A scenario path consists of one or more segments, related by connection elements and start- and stop stimuli. 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.

Example. 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. Note that a scenario path must obey the semantics of the value interface: either objects are exchanged on *all* its ports or *none* at all.

Example. The AND-fork between the *reader*, *Last Mile*, and *Amsterdam Times* is important: it ensures, that if values are exchanged via the *reader*'s value interface, values must be exchanged via the value interfaces and of the *Amsterdam Times* and *Last Mile* respectively.

Responsibility point. 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 construct a profit/utility sheet (see Sec. 4). Such a 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 each actor.

3 Discussing an e-business value model

3.1 e-Business value model observations

While the previous section discussed what an e-business value model looks like in terms of elementary concepts and scenarios, some other characteristics of an e-business value model can be seen from the pictures we draw.

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 on a stimulus. Simply by following the scenario path, it can be seen which exchanges of values via value interfaces result in other value exchanges.

Example. In the e-business value model at hand the initial money flow is between the *reader*, and *Last Mile*. All other money flows are generated from the money earned by *Last Mile*.

Bundling of value ports- and objects. For several reasons, actors may decide to offer or request objects of value only *in combination*. For instance, objects are only of value for actor, if they are obtained in combination; or an actor may believe he earns more money if objects are sold in combination rather then separately. This is shown by grouping ports offering and requesting such objects into one value interface

Example. Article online, *telephone connection*, *termination*, and *telephone connection fee* are *bundled* into one value interface from a *reader*'s perspective. This models that a *reader* only values an *article online*, and a *telephone connection* in combination. Because we bundle these value objects, and because a *reader* needs to offer a compensation for these objects, *termination*, and *telephone connection fee* must also be part of the same value interface.

Customer ownership. In the situation that a customer (e.g. a *reader*) buys a product from only one seller regularly, such a seller start ti build up a relation with that customer, and owns the customer. Owning a customer is important, because it allows an actor for instance to build a profile of a customer, which can used to offer the customer new products or services in the future. Whether an actor *solely* ownes a customer can be seen by examining value offerings. If an offering is between two actors (a seller and a buyer), the seller 'owns' the buyer. However, if an offering contains more than one seller, customer owbership will be partitioned.

Example. 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, the reader now has to exchange values with *Amsterdam Times, and Last Mile*. The latter is even the party who 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 might be an unwanted situation from *Amsterdam Times*' point of view.

Power: Price setting. An important aspect of business power is the ability to determine the price for products or services to be delivered. By examining value interfaces and value exchanges, at least actors playing a role in pricing can be seen.

Example. The *reader* pays for the *article online* via a telephone connection fee to *Last Mile*. Unfortunately, no one, except *Last Mile* and perhaps a market regulation authority, can influence the pricing. Consequently, the success of the business value model depends largely on *Last Mile*. Also, the *Amsterdam Times* is unable to set a price itself for its *article online* offered to a *reader*.

Power: Selection. If a buyer is able to make a selection of a set of potential sellers, his selection power increases, due to competition. This can be seen if a buyer is connected to multiple sellers by multiple offerings.

Example. The *reader must* use *Last Mile* for local loop access. At the time the project was carried out, there was only one actor available controlling the local loop to subscribers. This can be concluded from figure 1, because only one actor for local loop access is drawn, and consequenlty only one offering between *Last Mile* and *reader* exists. Again, this makes the business value model very critical to the behavior of *Last Mile*.

Example. The business idea at hand has a special trick to enlarge the selection 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 deliver 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*.

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 [2]. This is graphically shown by placing cardinalities near value exchanges connected to ports.

Example. 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. 1 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.

4 E-valuating an e-business value model

We use a conceptual e-business value model, and scenario paths not only to create a common understanding of the e-business idea amongst stakeholders, but also to evaluate the economical feasibility of an e-business idea. We call this the *e-valuation* of the business idea, because it is based on an assessment of the value of objects by actors. Feasibility is in reach if *all* actors involved are able to make profit or increase their economic utility with an e-business idea. Again, our technique for determining feasibility is a light weight approach, and focusses on building *confidence* that an e-business idea is of interest for the actors involved, rather than offering a precise calculation of all profits. Our e-valuation approach consists of the following steps:

1. Creation of profit/utility sheets for all actors, and/or value activities;
2. Determination of a valuation scheme for each actor;
3. E-valuation of *what-if* scenarios.

Creation of a profit/utility sheet. Profit/utility sheets are created on an 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 space, we concentrate on the actor profit sheets only.

Table 1 shows a reduced profit/utility 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, we update the sheet with value objects flowing in- and out the actor. In Table 1, we have two composite actors. For brevity, we only show for composite actor 1 the profit sheets of its actors (*Data Runner* and *Hoster*).

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.

In some cases, it is not possible to remove all value objects, which are not money streams. This is especially the case for end-customers such as a *reader*. In such a situation, delivering or receiving such objects changes the economic utility for that actor. Due to lack of space, we do not discuss the valuation of such objects, but refer to [4], where we propose an approach for valuation of these objects. Therefore, we ommit the emphreader from the profit/utility sheets.

Determination of a valuation scheme. After reduction of the profti/utility sheet, the remaining value objects in the profit sheet have to be assigned a value by actors, expressed in monetary units (e.g. dollars). Table 1 shows such a valuation scheme, which we explain briefly:

Telephone connection fee. The telephone connection fee per scenario occurrence is based on a start tariff and a connection-time dependent tariff. To calculate the total monthly fees, the telephone connection fee is multiplied with the actual number of scenario occurrences.

Interconnection fee. The interconnection fee per scenario occurrence is based on a fraction (the interconnection factor) of the telephone connection fee, and on the physical distance *Data Runner* bridges.

Termination fee. The termination fee *Amsterdam Times* receives, 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. 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 keeps the same. This models a situation where *Data Runner* takes the risk of a decreasing interconnection factor solely.

Internet access fee for Amsterdam Times. 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 connection duration. *Data Runner* then allocates access ports based on this forecast, and can allocate remaining ports

Table 1. Profit/utility sheets and valuation schemes for the scenario *Read article online*.

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	$\textit{termination fees}_{composite1} = \textit{tel. connect. fee} * \textit{revenue sharing factor}_{composite1} * \textit{distance factor} * \textit{actual occ.} * p$	$\textit{inet access fees}_{composite1} = \textit{see Data Runner}$ $\textit{hosting fee}_{composite1} = \textit{see Hoster}$
Composite 1		
Data Runner	$\textit{interconnect. fees} = \textit{see Last Mile}$ $\textit{inet access fee}_{AmsterdamTimes} = \textit{inet. connect. tariff} * \textit{duration} * \textit{actual-forecast occ.} * p$ $\textit{inet access fee}_{Hoster} = \textit{occurences}_{forecast} * p \rightarrow \textit{needed bandwidth} \rightarrow \textit{fixed fee/month}$	$\textit{termination fees} = \textit{see Amsterdam Times}$
Hoster	$\textit{hosting fee} = \textit{concurrent occ.}_{forecast} * p \rightarrow \textit{concurrent pageviews} \rightarrow \textit{fixed fee/month}$	$\textit{inet access fee} = \textit{see Data Runner}$
Composite 2	...	...

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.
Internet access fee for Hoster. To make hosted internet applications available for the end-user, *Hoster* needs also internet access. The internet access fee to be paid by *Hoster* is entirely based on the forecasted number of scenario occurences. Based on this, we calculate a estimate of the required bandwidth, and calculate the price for this.
Internet hosting fee. 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. As an example, Table 2 shows a number of scenarios.

Table 2. Different valuation scenarios. The null-scenario uses the valuation parameters as we see them most likely for now. 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 scenario). 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

Null scenario. The *null* scenario is the situation at this moment. Observe that *all* actors make profit to cover additional costs.
Bad forecasting. What happens if the *Amsterdam Times* is not a good forecaster of scenario occurrences. It can be seen that *Amsterdam Times* will not make profit. For *Last Mile* and *Data Runner* there is still profit to cover costs. *Hoster* is insensitive to bad forecasts, because it is not dependent on the actual scenario occurrences.
Interconnection fee decreases. A decrease in the interconnection factor is reasonable to expect after a couple of month, because this factor is now high to stimulate competition between telecommunication operators. As soon as this competition works, the factor will decrease. *Amsterdam Times* does not feel such a decrease, but *Data Runner* will.

Revenue sharing fee decreases. Data Runner may decide to decrease its revenue sharing factor. As can be seen, this will harm *Amsterdam Times*.

In conclusion, by valueing the objects for each actor, and by making assumptions about the number of (forecasted) scenario accurrences, we can do a sensitivity analysis for the business idea hand. This sensitivity analysis is in most cases of more interest, than the valuation itself.

5 Conclusions

In this paper, we proposed a conceptual modeling approach for the development and representation of e-business models. The notion of *economic value*, and how objects are created, exchanged and consumed in a multi-actor network is a central theme in our ontology for e-business models.

We showed how a non-trivial business idea can be represented using our e^3-*value* technique. Besides the causality of revenue streams, our e^3-*value* models also show other aspects, such as bundling, customer ownership, power in pricing and actor selection, and partnership.

Using well-known scenario techniques, we constructed profit sheets, which in turn were used to get a first impression of the profitability of the business idea for each actor. These profit sheets should not be seen as absolute profit predictors, but more as calculation schemes with which a sensitivity analysis can be done. Using *what if* scenarios, which focus on expected changes in the future, even more confidence can be built.

For the project at hand, our approach was especially useful to articulate the business idea precisely, thereby creating a common understanding amongst stakeholders. The e-business model appeared to be too complicated, to communicate in natural language. Moreover, the valuation scheme and the what-if scenarios were of use during contract negotiations, especially to enhance transparency and to evaluate future developments and risks for all actors involved quickly.

Acknowledgement. This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project nr VWI.4949. Also, we thank *De PersCombinatie* for their approval to make this paper public.

References

1. R.J.A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.
2. Soon-Yong Choi, Dale O Stahl, and Andrew B. Whinston. *The economics of doing business in the electronic marketplace*. Macmillan Technical Publishing, Indianapolis, 1997.
3. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Business modelling is not process modelling. In Stephen W. Liddle and Heinrich C. Mayr, editors, *Conceptual Modeling for E-Business and the Web, Salt Lake City, Utah, 9-12 October*, volume LNCS 1921, pages 40–51. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.

4. 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.
5. 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.
6. Amir Hartman, John Sifonis, and John Kador. *Net Ready - Strategies for Success in the E-conomy*. McGraw-Hill, New York, 2000.
7. R. Jasper and M. Uschold. A framework for understanding and classifying ontology applications. In B. Gaines, R. Cremer, and M. Musen, editors, *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW'99 (16-21 October 1999, Banff, Alberta, Canada)*, volume I, pages 4–9–1 — 4–9–20, Calgary, 1999. University of Calgary, SRDG Publications.
8. P. Kotler. *Marketing management: analysis, planning, implementation and control*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
9. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The knowledge engineering review*, 13, 1998.