What's in an electronic business model?

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

[1] Vrije Universiteit Amsterdam
Vuture.net — Amsterdam Center of e-Business Research
De Boelelaan 1081a, NL-1081 LV Amsterdam, The Netherlands
{gordijn,HansAkkermans,hans}@cs.vu.nl

[2] Deloitte & Touche Bakkenist Management Consultants
Wisselwerking 46 NL-1112 XR Diemen, The Netherlands

[3] AKMC Knowledge Management BV
Klareweid 19, NL-1831 BV, Koedijk, The Netherlands

Submission 12th Int. Conf. on Knowledge Engineering and Knowledge Management
EKAW-2000, 2-6 October 2000, Juan-les-Pins, France

Abstract. An electronic business model is an important baseline for the development of e-commerce system applications. Essentially, it provides the design rationale for e-commerce systems from the business point of view. However, how an e-business model must be defined and specified is a largely open issue. Business decision makers tend to use the notion in a highly informal way, and usually there is a big gap between the business view and that of IT developers. Nevertheless, we show that conceptual modelling techniques from IT provide very useful tools for precisely pinning down what e-business models actually are, as well as for their structured specification. We therefore present a (lightweight) ontology of what should be in an e-business model. The key idea we propose and develop is that an e-business model ontology centers around the core concept of *value*, and expresses how value is created, interpreted and exchanged within a multi-party stakeholder network. Our e-business model ontology is part of a wider methodology for e-business modelling, called *e*^{3}*-value*^{TM}, that is currently under development. It is based on a variety of industrial applications we are involved in, and it is illustrated by discussing a free Internet access service as an example.

1 Introduction

The design of an electronic commerce application is in our view *not* primarily an IT-oriented activity. Rather, it consists of very different types of design problems [10]. The most important of these is the design of the *e-business model* which highlights the way of doing business. A business model should do so in a very precise way, because stakeholders such as chief executive officers, marketers, and business developers should agree on it, and because it is a crucial bottomline part of the requirements for an electronic commerce system. For example, how do we develop the IT infrastructure and application system for a free Internet service? This cannot be really done without knowing what the underlying business model for the service is in the first place.

Therefore, we propose an *ontology* [12, 3, 6] to define from a generic point of view what should be in an e-business model. The key idea we propose and develop in this paper is that an e-business model ontology centers around the core concept of *value*, and expresses how value is created, interpreted and exchanged within a multi-party stakeholder network of (extended) enterprises and customers. It is exactly this notion of value which is currently lacking in information modelling and analysis approaches, including various business-oriented ontologies that have been developed recently.

The present work is part of a broader methodology for e-business development, called e^3-*value*^{TM}, we are currently developing [10]. It reflects and structures the strategic business decisions that need to be made at the executive level on the e-business model and on business-IT alignment, before one can proceed to the technical design of an electronic commerce system. In Sec. 2, we discuss the need for an e-business model ontology. Sec. 3 describes our e-business model ontology, and we illustrate it by a case study. In Sec. 4 we discuss related work, and we briefly summarize the practical use of the ontology in consultancy and application projects.

2 The need for a business model ontology

Normally, the design of an electronic commerce system starts with the development of a business model. In most cases, such a business model is written down in natural language, perhaps with some informal sketches. The concepts and their interpretations used to describe a business model vary across different stakeholders, and this leads to important obstacles to achieve business-IT alignment in e-commerce applications. Given the enabling role of IT in electronic commerce, this alignment problem is no longer just an engineering issue: it has a strategic significance.

During the design of a business model, an ontology is therefore useful to *prescribe* which concepts and relations have to be present in a business model. An ontology should provide a reusable conceptualisation, in this case of the concept of *e-business model*, on which people can agree. By specializing and instantiating concepts and relations of the ontology for a particular case, the ontology can also be used to describe a particular business model in a precise and structured way. In the present context, we are mainly interested in ways to enhance communication between various stakeholders, that is, in shared meaning rather than automated reasoning. Thus, our current goal is to construct a so-called ‘lightweight’ ontology [17].

Furthermore, a business model ontology shows designers what kind of decisions should be taken during business model development. If stakeholders agree on a particular business model, a number of business decisions have been taken, so that the model serves as a precise set of business requirements for the electronic commerce information system. These requirements are useful for software architects who design the electronic commerce system from a technical point of view.

An ontology for e-business models must be capable of representing a range of business issues. These issues center, and this is our key proposal, around the generic concept of *value*, and how to create and exchange it in a network setting.

Informally, a business model highlights a network of actors and how they create or consume objects of value. These actors can be private persons, companies or enterprise alliances. Furthermore, a business model represents the services offered by and requested from actors. It should be capable to represent if an actor is willing to exchange an object of value (e.g., the right to listen to a music track) for another object of value (e.g., money). Also, a business model illustrates which actors can have economic transactions with other actors. A transaction is possible if actors offer each other objects of value in which both have a mutual interest. Finally, actors must perform activities to create value; for other actors or even for themselves. The assignment of activities to actors is an important element in e-business models for decision makers. The above business model concepts, which are more formally expressed in our *e*^{3}-*value*^{TM} ontology, originate from scientific studies from a variety of (non-IT) disciplines, in particular marketing [18], axiology [14], business administration [19, 20], and emerging e-commerce theory [4, 16, 22]. In the next section, we present a lightweight ontology that is capable of representing these business issues to various kinds of stakeholders.

3 An ontology for e-commerce business models

The *e*^{3}-*value*^{TM} ontology contains concepts, relations, and constraints, to describe actors, alliances between them, the exchange of objects of value, the value-adding activities, and the value interfaces between them. We identify three different views for describing business models for specific business cases. The *global actor* view shows which parties are involved in a business model and which objects of value they exchange. Its main purpose is to explain the overall business model to a wide range of stakeholders. The *detailed actor* view takes a further look at the decomposition aspects. It shows, for actors identified in the global actor view, alliances between parties, for instance virtual enterprises [5]. Finally, the *value activity* view shows the assignment of value-adding activities to actors. The ontology is illustrated by a small case study about a free Internet access service. In The Netherlands, a number of parties are offering such a service. Suppose one is asked to develop the business model of such a service (in actual fact, our example is taken from a real-life case). We show that our ontology can be used to answer such a 'fuzzy' question.

Global actor view. Figure 1 shows the global actor view of a business model for the free Internet access service. Its main purpose is to illustrate the overall business model to all stakeholders. The global actor view shows *actors* involved, such as *surfer* and *free Internet provider*, and the *exchange* of *value objects* between them. A value exchange has a direction, visualized by an arrow, indicating the direction the value object 'flows'. In this case, the *surfer* pays the *free Internet provider*. Value exchange links start and end at *value ports*. These ports are not visualized explicitly at this global level; they are the points connecting the value exchange with actors. Ports are grouped into a *value interface*, modelling the service an actor offers to its environment (also not drawn explicitly at this level). We note that this concept of ports and interfaces actually stems from ontologies relating to systems theory [3]. Some value exchanges relate to each other, for instance *payment* and *Internet access*. This is called an *offering*. In an offering both exchanges need to occur: there is no *Internet access* possible without *payment* and vice versa. Note that, apparently, the free Internet access service is actually not for free at all: a surfer has to pay for the telephone connection.

The business model in Figure 1 is a specialization and instantiation of concepts and relations in the *e*^{3}-*value*^{TM} ontology (Figure 2). They are discussed in more depth in this section, and so are the specialization and instantiation of concepts and relations in the ontology for the free Internet access business case. The explanation of our ontology is structured by presenting a description for each concept, properties of the concept,

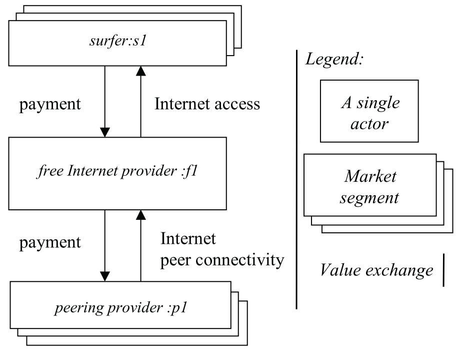

Fig. 1. Business model for the free Internet case: the global actor view.

relations with other concepts, constraints, and the visualization in a business model such as depicted in Figure 1. Each concept and relation is illustrated by a practical example.

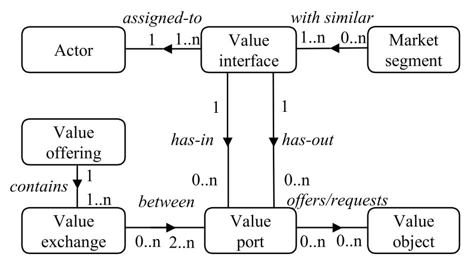

Fig. 2. Core concepts in the e^3-*value*^{TM} ontology of e-business models (global actor view).

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. Enterprises, strategic business units, and customers are examples of actors.

PROPERTIES. An actor may have a name, e.g. a company name.

EXAMPLE. We identify three specializations of the *actor* concept: (1) the *surfer* actor, (2) the *free Internet provider* actor, and (3) the *peering provider* actor (Figure 3). The *surfer* actor uses the *free Internet provider* to surf the Internet. The *free Internet provider* uses *peering providers* to deliver traffic at the Internet host the *surfer* selected. Peering is necessary to come at an interconnected network of Internet hosts. Instances of *surfer* (s1, s2,.., sn) , one instance of *free Internet provider* (f1) and a number of instances of *peering provider* (p1, p2,.., pn) are represented in Figure 1. Note that worldwide, a number of free Internet providers exist, but for this case, we are only interested in one.

Value Object. Actors exchange value objects. A value object is a service or product which is of value for the actors. Actors may value an object differently and subjectively, according to their own valuation preferences [14].

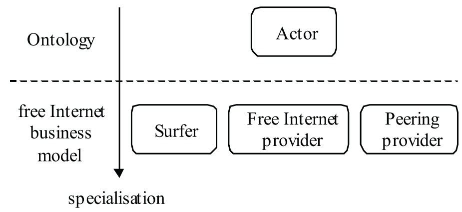

Fig. 3. Specialization of the actor concept.

PROPERTIES. A value object has one or more valuation properties. A valuation property has a name and a unit which indicates the measuring scale in which the valuation is expressed. In general, the quantification of value has to be done by means of a multi-dimensional utility function [2, 9].

EXAMPLE. *Internet access* is a specialization of *value object* and represents the service offered by the *free Internet provider* to *surfers*. *Internet access* is valued in terms of *connection time* which is expressed in seconds and the Committed Information Rate (*CIR*), measured in bits per second. Other value objects are *money* and *Internet peer connectivity* (Figure 4).

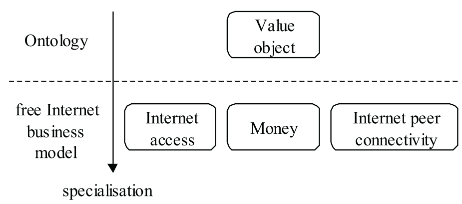

Fig. 4. Specialization of the value object concept.

Value Port. An actor uses a value port to provide or request value objects to or from its environment. Thus, a value port is used to interconnect actors so that they are able to exchange 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 model can be 'plugged in'. This is the value analogue of the separate external interfaces familiar from technical systems theory [3]. Take, for example, a bipolar in+out value multi-port, which is the most characteristic combination occurring in e-business models: an e-service port out and a money port in, or the other way around. Such a bipolar value port combination can be very well compared to an electrical wall outlet. As an external user, you don't want to be involved in what happens behind the wall outlet as long as it gives the right quality of service. The same approach holds for how external parties in an e-business model view the value ports of a service-offering actor: the ports only define how the external connections to other actors should be made.

RELATIONS. Value ports *offer* or *request* value objects. A value object can be requested or offered by multiple value ports.

EXAMPLE. Consider the *Internet access* port, as a specialization of the value port concept. The *offer or request* relation is specialised into a relation between the *Internet access* value port and the *Internet access* value object (Figure 5). The business model (Figure 1) shows two instances of the *Internet access* port. The *surfer* has an in-port and the *free Internet provider* has an out-port.

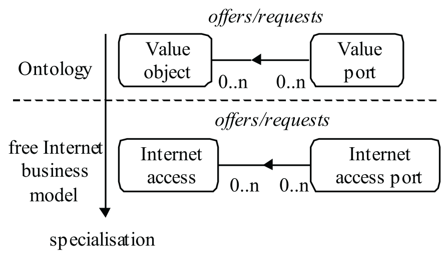

Fig. 5. Specialization of the value port concept and relations.

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 objects an actor is willing to exchange in return for other value objects via its ports.

PROPERTIES. A value interface has a *valuation function*. It expresses, given valuation properties of objects of all in-ports, the required valuation properties of objects on all out-ports, and vice versa. In other words, a valuation function shows if an actor is willing to exchange value objects in return for other value objects. The valuation of objects depends on a specific actor evaluating the valuation function [14]. The valuation function has a *direction* argument. If the direction is *in*, the valuation function returns the required valuation properties of the value objects on all in-ports. If the direction is *out*, the opposite happens.

RELATIONS. A value interface *is assigned to* one actor and *has* zero or multiple *in* value ports and *has* zero or multiple *out* value ports. A value interface has at least one value port. Multiple value interfaces can be assigned to an actor and a port belongs to exactly one value interface. If an actor has multiple value interfaces, s/he is offering different services to the environment.

CONSTRAINTS. The exchange of value objects is atomic at the level of the value interface. Either all exchanges occur as specified in the value interface or none at all. For instance, a *surfer* cannot obtain Internet access without paying. The value interface says nothing about the time ordering of objects to be exchanged on its ports. It simply states which value objects are available, in return for some other value objects.

EXAMPLE. The *surfer* has a specialized value interface called *S-Internet-access* which consists of a *payment* out-port and an *Internet access* in-port. It is important to recognize that the Internet access service is not free at all. The *surfer* has to pay for its telephone connection. The *free Internet provider* has a similar interface, with opposite port directions (Figure 6). Note that cardinality constraints for the *has-out* and *has-in* elations are specified more strictly for the specialization. For example, an *S-Internet-access* value interface consists of exactly one *payment* port and exactly one *Internet-access* port.

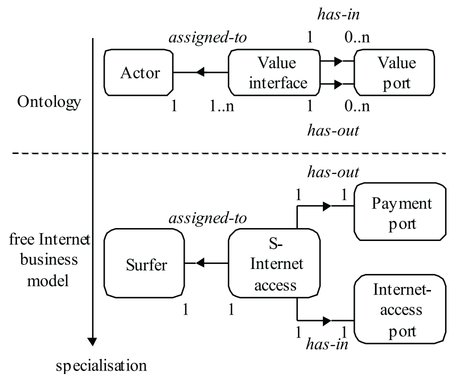

Fig. 6. Specialization of the value interface concept and relations.

As an example, for the *surfer* s1 a possible valuation function on its *S-Internet-access* value interface *vi-s1* is shown.

```
s1.vi-s1.valuation(dir:direction) {
    if dir == out {
        if Internet-access.Committed-Information-Rate < 3,000 {
            return:
            payment.amount <= (0.02/60) * Internet-access.duration;
        }
        else {
            return:
            payment.amount <= (0.04/60) * Internet-access.duration;
        }
    }
    else {
        if Internet-access.Committed-Information-Rate < 3,000 {
            return:
            Internet-access.Committed-Information-Rate < 3,000;
            Internet-access.duration <= payment.amount * (60/0.02);
        }
        else {
            return:
            Internet-access.Committed-Information-Rate >= 3,000;
            Internet-access.duration <= money.amount * (60/0.04);
        }
```

```
23     }
24 }
```

Lines 2-11 show a *surfer* willing to pay a certain amount of money for Internet access. This amount depends on the Committed Information Rate. If the CIR is less than 3,000 bps, s/he wants to pay a maximum of 2 dollar cents per minute; if the CIR is greater or equal than 3,000 bps s/he is willing to pay a maximum of 4 dollar cents per minute. Lines 13-23 show the CIR and minimum access duration a *surfer* requires for the money paid.

Value Exchange. A value exchange represents the trade of a value object between value ports. There are different kinds of value exchanges. First, seen from a port of an actor, value exchanges may occur to other ports of, possibly different, actors (Figure 7(a)). For instance, the port of actor A offering *music* can do so to ports of different actors B and C. This models the situation that multiple actors buy a track of music. Second, it is possible that a number ($>$ 2 ports) are involved in *one particular value exchange*. The following two situations may then occur. Figure 7(b) represents a *split* of the value object, in this case, an amount of money. Actor A pays an amount of money to actor B and C in one value exchange. The situation in Figure 7(c) models *duplication* of a value object. Duplication of a value object is only possible if the marginal costs to create a replica are zero. This may be the case for value objects such as music, video and information. Actor B and C both receive a duplicate of a music track of actor A in one value exchange.

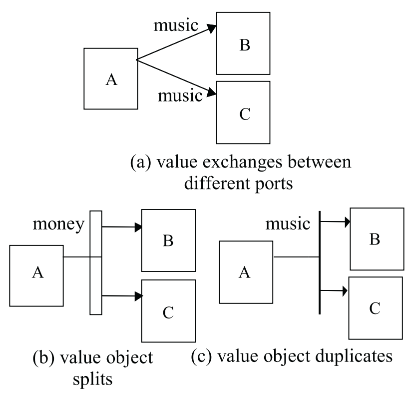

Fig. 7. Different types of value exchanges.

RELATIONS. The value ports involved in a value exchange are represented by the *between* relation. At least two value ports participate in a value exchange. A value port can be in multiple value exchanges.

CONSTRAINTS. A value exchange occurs between ports of opposite directions. A value object flows from an out-port into an in-port. Therefore, at least one in-port and one out-port should be present in a value exchange. Value ports can be seen as the end-points or terminals of a value exchange.

EXAMPLE. An *Internet access* exchange is a specialization of a *value exchange*. In an Internet access value exchange, exactly two value ports participate (Figure 8). Value exchanges occur between *surfers* and the *free Internet provider*.

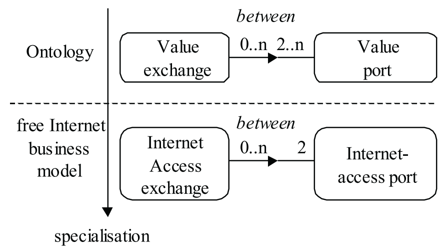

Fig. 8. Specialization of the value exchange concept and relation.

Value Offering. A value offering is an assembly of value exchanges. In an offering, value exchanges between multiple actors (≥ 2) can participate.

RELATION. A value offering *contains* a number of value exchanges. A value exchange participates in exactly one value offering.

CONSTRAINTS. All ports of an actor's value interface should be connected to other ports, but the ports of one actor's value interface may be connected with ports of different value interfaces. This is represented in Figure 9. If an offering occurs, exchanges of value objects on *all* its value exchanges occur.

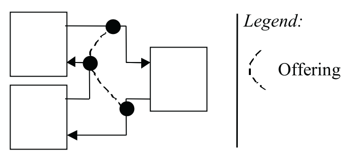

Fig. 9. Ports in an actor's interface connected to ports of two other actors.

EXAMPLE. A *free Internet access* offering contains exactly one *Internet-access exchange* and one *payment exchange* (Figurre 10). The two value exchanges between the *free Internet provider f1* and the *surfer s1* clearly are an offering (Figure 1). The same holds for the value exchanges between the *free Internet provider f1* and the *peering provider p1*.

Market segment. In the marketing literature [18], 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 a similar valuation function. Consequently, because valuation functions are bound to value interfaces, actors in a segment all have at least one similar value interface. Value exchanges and value offerings drawn to a segment are a shorthand notation for value exchanges and offerings between all actors of the segment, and other actors. Figure 11(a) shows an actor exchanging values with three other actors. Figure 11(b) shows the same but now with the three actors having a similar valuation function.

PROPERTIES. A market segment has a *count*, which indicates the number of actors in the segment. The count can be a number, unbounded, or unknown.

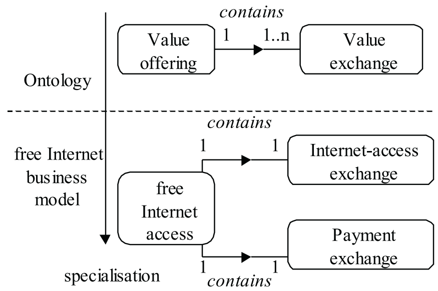

Fig. 10. Specialization of the value offering concept and relation.

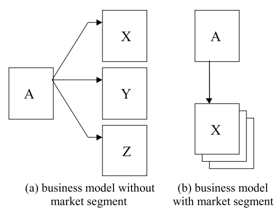

Fig. 11. A business model without and with market segment.

RELATIONS. Value interfaces of actors are *part of* zero or more market segments. A market segment contains one or more value interfaces.

CONSTRAINTS. Value interfaces of actors in a market segment should all have a similar valuation function (shown as a 'stack' of actors). Note that actors in a segment may also have in-similar value interfaces.

EXAMPLE. It is reasonable to expect that, with respect to the valuation function, a number of different *surfers* exist. Some surfers are willing to pay quite some money for high quality Internet access (heavy surfers) while others are only interested in sending low-bandwidth email and want to pay a small amount of money (light surfers). These can be grouped in a *heavy surfer* segment and a *light* segment (Figure 12).

Discussion. The global actor view shows the most important actors in a business model. Furthermore, it shows the objects of value exchanged between these actors, as well as offerings. The market segment notion is useful if offerings are of interest to a number of actors who share the same valuation function. The global actor view can easily be constructed in brainstorm sessions and workshops with all key actors. Also, this view can be used to present and explain the overall business model to stakeholders. For the free Internet access service, the global actor view illustrates that the free Internet access service is offered to surfers. However, the service is not for free at all, since the surfer has to transfer money for Internet access. This is due to costs for the telephone connection. Also, this view shows that, to offer an Internet access service, peering services have to be contracted with peering providers.

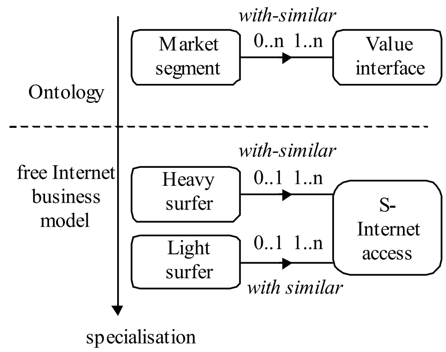

Fig. 12. Specialization of the market segment concept and relation.

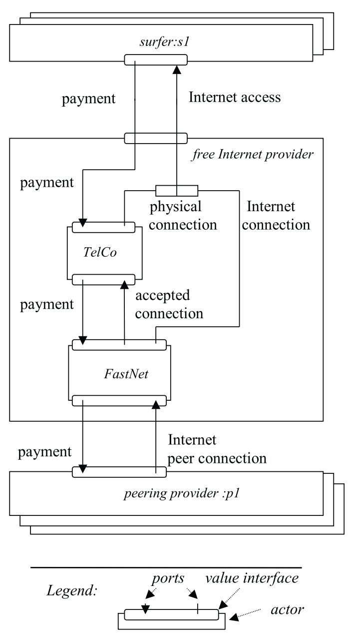

Fig. 13. Business model for the free Internet case: the detailed actor view.

The detailed actor view: decomposition aspects. The purpose of the *detailed actor view* (Figure 13) is to show alliances between actors. For reasons of space we only show and discuss the detailed actor view for the *free Internet provider*. A detailed actor view can be developed for the *peering provider* as well.

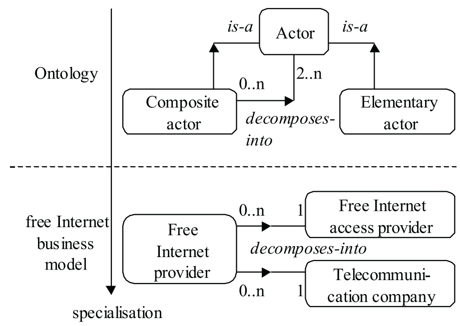

Fig. 14. Specialization and decomposition of the actor concept.

Composite actor and elementary 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 present themselves, as a single (virtual enterprise) actor 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. Actors can be composed of other composite actors and/or elementary actors. In the global actor view we do not state explicitly whether an actor is elementary or composite. In the detailed actor view, we refine actors of the global view into their constituents.

RELATIONS. A composite actor *is an* actor. An elementary actor *is an* actor. A composite actor *decomposes into* other actors. Actors may be part of zero or more composite actors.

EXAMPLE. *Telecommunication company* and *free Internet access provider* are specializations of the *elementary actor* concept. A telecommunication actor offers *physical connectivity* for data transport. A free Internet access provider offers *Internet access*. These actors jointly offer a free Internet service, resulting in a composite actor called *free Internet provider* (Figure 14). *TelCo* and *FastNet* (Figure 13) are instances of *Telecommunication company* and *free Internet access provider*, respectively.

Composite value object and elementary value object. *Composite* value objects can be decomposed into other value objects. A composite value object can be built from other value objects which may be provided by different actors. *Elementary* value objects cannot be decomposed any further. A value object can be in only one composite value object.

RELATIONS. An elementary value object and a composite value object *is a* value object. A composite value object *decomposes into* other value objects.

EXAMPLE. *Physical connection* and *Internet connection* are specializations of the *elementary value object*. These value objects can be composed to form *Internet Access* (Figure 15).

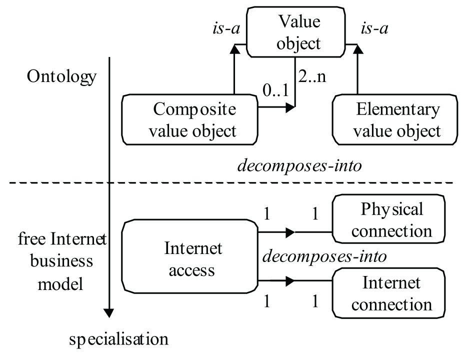

Fig. 15. Specialization and decomposition of the value object concept.

The exchange of value. A composite actor has a value interface to its environment. However, a value interface of a composite actor must be mapped onto one or more value interfaces of actors which are part of the composite. This mapping is represented by value exchanges and value offerings. To be able to present these mappings accurately, we use a rounded box to visualize a value interface of an actor and an arrow to presents a value port of the value interface. The direction of the arrow indicates whether a value object flows in or out the actor. In Figure 16, a composite actor *a1* is shown, consisting of actors *b1* and *b2*. The ports in the value interface of *a1* are connected using value exchanges with value ports of *b1* and *b2*. On port *p1*, a value object is offered to the environment of actor *a1*. This object is offered by port *p3 or* by port *p4*. Another object of value is requested in return on port *p2*. Internally this object is split in two objects, to port *p4*, and port *p6*.

RELATIONS. The value ports involved in a value exchange are represented in the *between* relation.

CONSTRAINTS. *All* connected ports in value exchange should have direction *in* or *out*.

EXAMPLE. The *free Internet provider* consists of two actors: *Telecommunication company TelCo* and *free Internet access provider FastNet*. These companies are jointly offering an Internet access service. The externally visible value port *Internet access* of the free Internet provider is mapped onto the *physical connection* port of *TelCo* and the *Internet connection* port of *FastNet*. The other externally visible port of the *free Internet provider* is the *payment*. This port is mapped onto the *payment* port of *TelCo* because *TelCo* receives payment of the *surfer*.

Discussion. The detailed actor view intends to represent actors jointly offering or requesting a product or service. For each actor in the global actor view, detailed actor views may be considered. Such a detailed view consists of actors sharing a particular value interface to their environment. Furthermore, the detailed actor view shows how the shared value interface is mapped onto value interfaces of the actors themselves. Therefore, in the detailed actor view we make the value interfaces and value ports explicit. Note that *FastNet* and *Telco* jointly only offer the Internet Access service. *FastNet* itself has a value interface for Internet peering; *TelCo* has nothing to do with this.

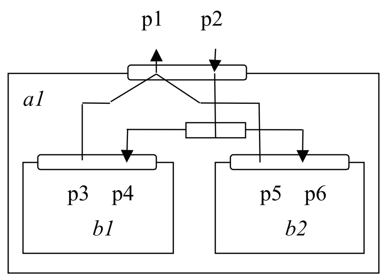

Fig. 16. Value exchanges between a composite actor and its composites.

The value activity view. We now discuss the *value activity* view. Its main purpose is to illustrate the assignment of value-adding activities to actors. Figure 17 shows this view for *TelCo* and *FastNet*. How value-adding activities are assigned to the various possible actors is a free variable that, as a result of the extended enterprise network setting, leads to many design options and choices in e-business models. Hence, this assignment is a key consideration in strategic e-business decision making.

Value Activity. A value activity is *performed by* an actor and produces objects of value *for* an actor. Both these actors can be different entities but they may also coincide. Consider an actor listening to music s/he bought in order to have a nice experience. In such a case, the actor performs a value activity (listening) and produces an object of value for him/herself (namely, a nice experience: note that what constitutes value may be rather abstract and interpretive). An important issue in e-commerce business model design is the *assignment* of value activities to actors. Therefore, we are interested in the collection of activities which can be assigned as a whole to actors. Such a collection we call a value activity. Therefore, the granularity of value activities should be such that they can be performed economically independent from other value activities [20], *and* they cannot be further decomposed into smaller economic activities that can be assigned to different actors (this gives a decomposition stop rule, which is by the way clearly different from business process or workflow decomposition). Value activities can be assigned to an elementary actor but also to a composite actor. In the latter case, the composite actor is not composed of other actors only (such as a virtual enterprise), but it can perform value activities by itself.

RELATIONS. A value activity *has* one or more value interfaces. A value interface belongs to exactly one value activity. A value activity is *assigned-to* precisely one actor. Multiple value activities can be assigned to an actor.

EXAMPLE. The *value activity* concept is specialized into the *call-delivering* and *Internet access* value activity. A *call-delivering* value activity has two value interfaces: (1) the *connection interface*, modelling a physical connection service which has to be paid for, and (2) an *acceptance interface* which models that a connection should be accepted by someone else, before one can speak of a connection (Figure 18). *FastNet*, which has been assigned the *Internet access value* activity, accepts physical connections for *TelCo*.

Discussion. The value activity view shows which value activities are assigned to specific actors, and how value interfaces of these activities map onto value interfaces of

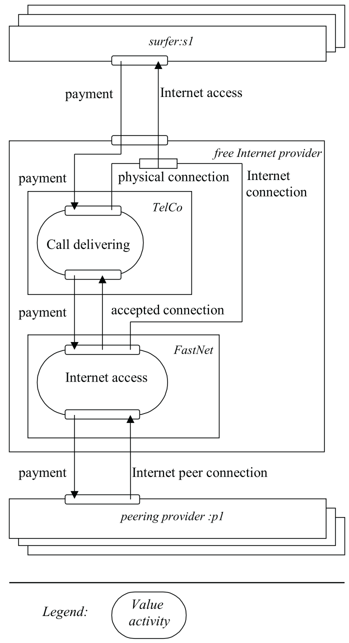

Fig. 17. Business model for the free Internet case: the value activity view.

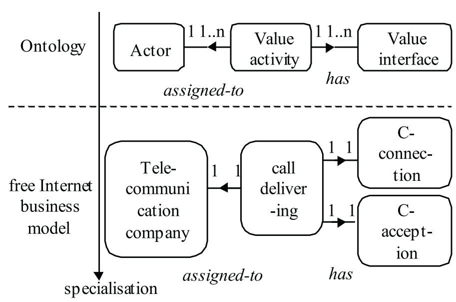

Fig. 18. Specialization of the value activity concept and relations.

actors. For the free Internet access service, the assignment of value activities is rather arbitrary. However, alternatives, not considered in this paper, are to assign the value activity *Internet access* to *TelCo* also, or to assign the value activities *Internet access* and *call delivering* to a telecommunication company only. Such alternative assignments would also lead to changes in the detailed actor view: they constitute different business models.

4 Discussion and conclusion

Related work. There are some related business-oriented ontologies, in particular the AIAI enterprise ontology [23] and the TOronto Virtual Enterprise Ontology (TOVE) [7]. The most important difference with our e^3-*value*^{TM} ontology is that we focus on the notion of value and the way objects of value are created, exchanged and consumed in a stakeholder network, while the enterprise ontology and TOVE concentrate on the enterprise itself, the latter resulting in a business process rather than external value perspective.

AIAI enterprise ontology. The enterprise ontology 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*^{TM} 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 *offering*, with the important difference that a sale is an actual agreement, and an offering only a potential one. An offering contains *value exchanges*. In the enterprise ontology, only two goods are exchanged in a sale. In contrast, in our ontology an offering 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 offerings. The case study in this paper illustrates the need for such a concept.

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

System-theoretic ontology. As pointed out earlier in this paper, the e-business ontology reuses several concepts from general and technical systems theory and associated ontologies [3]. In particular, the introduction of the concepts of ports and interfaces of a (network) system help to abstract away from the internal workings of an activity (or subsystem), and to independently specify the connection to the environment (external suubsystems). This is an important advance over what is typically done in business process and workflow modelling [8].

Use of the ontology in e-business development. In summary, this paper is premised on the observation that for the development of electronic commerce systems, e-business models must be specified precisely. Such a clear-cut specification is important for two reasons: (1) to reach agreement between stakeholders involved, and (2) to be able to serve as a specification for designers of the commerce system. The e^3-*value*^{TM} ontology discussed in this paper specifies which generic concepts have to be present in an e-business model. These concepts are based on the generic and reusable notion of *value*, and are capable of representing creation, exchange, valuation, and consumption of value objects in a network of actors.

Of course, for e-business development an ontology is only one of the necessary tools. It must be embedded in a wider process of e-business modelling and application development. The present paper is rather descriptive in nature, but the ontology has several more dynamic and practical uses in e-business development that are beyond the scope of this paper. In brief:

1. The e^3-*value*^{TM} ontology gives a baseline of shared concepts with which it is possible to construct e-business models. This baseline is much more rigorous, and therefore more amenable to IT systems follow-up, than value-oriented business approaches such as [19, 20]. It is also richer as it handles external value networks and not just value chains — an extension we believe to be essential for e-commerce.
2. It is our experience that e-business models (on the basis of this ontology, especially the global actor view) can be constructed during workshops or brainstorm sessions with stakeholders such as executive management. This is similar to experiences with management workshops in knowledge management, see e.g. [1, 21].
3. We have developed a set of steps, business rules and guidelines, and scenario techniques for practitioners (rooted in the ontology concepts) that structure, steer and simplify the *process* of designing and evaluating e-business models. More on this process is found in [10, 11].
4. Our ontology has been described in this paper in a semi-formal way. This is in line with its use as a lightweight ontology to enhance communication between different stakeholders [17]. However, tool development is ongoing, and a working Prolog implementation of the ontology has been constructed. There are thus no significant obstacles to formalize e^3-*value*^{TM} in terms of one of the formal language approaches to ontology [6, 15].
5. An important step not discussed in this paper is to extend the work to a quantitative formulation of the concept of value. This would enable to analyze and make choices between business models on quantitative grounds, by linking value analysis to methods and results from utility theory, decision theory and optimization. We are currently researching how to make the transition from qualitative and interpretive customer value notions [14] to quantitative utility analysis. For some application areas we have shown that this indeed can be done, see [9, 2] for applications to web selling of digital music content and to automatic cost-efficient home comfort management, respectively.
6. At the IT level, this then provides the basis for agent-based e-business system implementations and solutions. Corresponding, extensive and real-life applications where

economic agents make local decisions based on utility considerations, are described in some of our other work [24, 13, 25, 2].

Thus, an important virtue of the ontology approach is that it provides a foundation to express and discuss e-business models for specific business cases in a rigorous and structured fashion. In addition, this enhances business-IT alignment and smoothens the transition to e-commerce systems development.

Acknowledgment. This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project nr VWI.4949.

References

1. J.M. Akkermans, P.-H. Speel, and A. Ratcliffe. Problem, opportunity, and feasibility analysis for knowledge management: An industrial case study. 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 2–1–1 — 2–1–22, Calgary, 1999. University of Calgary, SRDG Publications.
2. E. Boertjes, J.M. Akkermans, R. Gustavsson, and R. Kamphuis. Agents achieving customer satisfaction — the COMFY comfort management system. In *Proceedings 5th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology PAAM-2000 (Manchester, UK, 10-12 April 2000)*, pages 75–94, Blackpool, UK, 2000. The Practical Application Company Ltd. ISBN 1-902426-07-X.
3. W.N. Borst, J.M. Akkermans, and J.L. Top. Engineering ontologies. *International Journal of Human-Computer Studies*, 46:365–406, 1997.
4. Soon-Yong Choi, Dale O Stahl, and Andrew B. Whinston. *The economics of doing business in the electronic marketplace*. Macmillan Technical Publishing, Indianapolis, 1997.
5. W.H. Davidow and M.S. Malone. *The virtual corporation - structuring and revitalizing the corporatio for the 21st century*. HarperCollings, New York, 1992.
6. D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, D, 2000. Series LNAI Vol., to appear.
7. M.S. Fox and M. Gruninger. Enterprise modelling. *AI Magazine*, pages 109–121, Fall 1998.
8. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Business modelling is not proces modelling. In *ECOMO-2000 Workshop on Conceptual Modelling Approaches for e-Business (Salt Lake City, Utah, 9-12 October 2000*, 2000. Submitted. Available from http://www.cs.vu.nl/ gordijn.
9. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Selling bits: A matter of creating consummer value. In *First International Conference on Electronic Commerce and Web Technologies ECWEB-2000 (Greenwich, UK, 4-6 September 2000*, 2000. Submitted. Available from http://www.cs.vu.nl/ gordijn.
10. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Value-based requirements creation for electronic commerce applications. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33)*, Los Alamitos, CA, January 4-7 2000. IEEE Computer Society. CD-ROM ISSN 0-7695-0493-0/00.

11. J. Gordijn, H. de Bruin, and J.M. Akkermans. Integral design of E-Commerce systems: Aligning the business with software architecture through scenarios. In H. de Bruin, editor, *Proceedings Conference on ICT Architecture in the BeNeLux*, Amsterdam, NL, 1999. Free University and Cap Gemini. Also available from http://www.cs.vu.nl/ gordijn.
12. T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal ontology in conceptual analysis and knowledge representation*, Amsterdam, 1994. Kluwer.
13. R. Gustavsson. Agents with power. *Communications of the ACM*, 42:41–47, March 1999.
14. Morris B. Holbrook. *Consumer value: a framework for analysis and research*. Routledge, New York, 1999.
15. I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, and R. Studer. The ontology-based web inference layer OIL. Technical Report OnToKnowledge EU-IST Project Draft Deliverable, University of Manchester, UK, March 2000. Available from http://www.ontoknowledge.com/oil.
16. J. Hagel III and A.G. Armstrong. *Net Gain - Expanding markets through virtual communities*. Harvard Business School Press, Boston, Massachusetts, 1997.
17. 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.
18. P. Kotler. *Marketing management: analysis, planning, implementation and control*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
19. R. Normann and R. Ramirez. *Designing interactive strategy - From value chain to value constellation*. John Wiley & Sons Inc., Chichester, 1994.
20. M.E. Porter and V.E. Millar. How information gives you competitive advantage. *Harvard Business Review*, pages 149–160, 1985.
21. A.Th. Schreiber, J.M. Akkermans, A.A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van Der Velde, and B.J Wielinga. *Knowledge Engineering And Management*. The MIT Press, Cambridge, MA, 2000. 455 + xiv pages, ISBN 0-262-19300-0.
22. Carl Shapiro and Hal R. Varian. *Information Rules*. Harvard Business School Press, Boston, Massachusetts, 1999.
23. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The knowledge engineering review*, 13, 1998.
24. F. Ygge and J.M. Akkermans. Decentralized markets versus central control — a comparative study,. *Journal of Artificial Intelligence Research*, 11:301–333, October 1999. Also available from http://www.jair.org.
25. F. Ygge, J.M. Akkermans, A. Andersson, M. Krejic, and E. Boertjes. The HomeBots system and field test — a multi-commodity market for predictive power load management. In *Proceedings 4th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology PAAM-99 (London, UK, 19-21 April 1999)*, pages 363–382, Blackpool, UK, 1999. The Practical Application Company Ltd. ISBN 1-902426-05-3. (Also available from http://www.enersearch.se/ygge).