Value Based Requirements Creation for Electronic Commerce Applications

Jaap Gordijn
Vrije Universiteit
Bakkenist Management Consultants
e-mail: gordijn@cs.vu.nl

Hans Akkermans
Vrije Universiteit
AKMC Knowledge Management
e-mail: hansakkermans@cs.vu.nl

Hans van Vliet
Vrije Universiteit
e-mail: hans@cs.vu.nl

Abstract

Electronic commerce applications have special features compared to conventional information systems. First, because electronic commerce usually involves yet non-existing business activities, requirements for e-commerce applications have to be created from scratch rather than elicited. Second, design decisions about e-business models and the associated information systems architecture cannot be sequentially made in a decoupled way, because business and technology considerations are strongly linked. On these counts, current methodology for requirements engineering is inadequate for electronic commerce applications. We outline a structured approach to e-commerce requirements creation. This e^3-VALUE approach enables one to clarify business model decisions to be made by management, by modelling the end-to-end value activities and exchanges in the e-commerce stakeholder network. In addition, this value network model enables system developers to derive high-level requirements concerning the software architecture.

1. Introduction

Requirements elicitation is the commonly used term for the very first stage of a systems development project [12]. It suggests that requirements are already somewhere present in the minds of prospective system users, and thus only have to be discovered by the system developers. This may be so for many conventional information systems, but it is certainly misleading in e-commerce applications. Here, system requirements cannot be discovered because usually the underlying business activities themselves are also new and in a state of being designed. Thus, system requirements have to be created rather than discovered, in parallel with e-business model design and decision making. Currently, requirements methodology for such situations is inadequate. In this paper, we sketch a structured approach, called e^3-*VALUE*, that gives a better grip on requirements elicitation in open-ended business situations as commonly encountered in innovative areas as electronic commerce.

First, we propose a semi-formal way to represent an e-business model, showing the essentials of doing business between different actors. Our e-business model centres around the key concept of *value*, and how value is created and exchanged within the stakeholder network. This value-configuration model can be constructed with the help of conceptual modelling techniques adapted from the information systems analysis field. Such a semi-formal value-configuration model supports e-business design and decision-making in a much more structured fashion than allowed by current methods from the business and management literature. Second, our value-configuration model is employed to generate high-level requirements on e-commerce application systems. We discuss how value-based business considerations do have an identifiable impact upon architecture design of e-commerce applications. Both these features of our approach, value modelling for e-business design and subsequent identification of associated requirements for the envisaged software architecture, advance requirements methodology for creative business areas such as electronic commerce.

This paper is structured as follows. Sec. 2 sketches our overall e^3-*VALUE* framework for electronic commerce applications. It distinguishes three levels at which different conceptual design decisions are to be made: e-business model development, e-business process design, and software architecture requirements. This provides a partial but useful separation of e-business design concerns. We suggest that the right entry point to e-commerce application development is value analysis for the various actors involved. An important contribution of our paper is its proposal of a limited baseline set of generic concepts to be used in value modelling, and of a corresponding six-step approach guiding the requirements creation process. This e^3-*VALUE* approach is then illustrated by two industrial case studies, one concerning international web-based advertising (Sec. 3), and another one on electronic added-value services in a deregulated energy business (Sec. 4). Both case studies are based upon our consultancy experiences in the e-business field. Sec. 5 summarises the general key points of our approach to electronic commerce applications.

2. The *e³-VALUE* framework for e-commerce applications

The development of an electronic commerce business application is not, as many see it, a requirement elicitation process [12], [17], which presupposes that stakeholders have tacit requirements that "just need to be extracted" by system developers. Instead, it is much better seen as a requirements *creation* process. This is caused by the novel nature of electronic commerce applications, the innovative character of e-business models, and the rapid developments in internet and web technology. Moreover, the design of an electronic commerce system is not in the first place an IT-oriented activity. Rather, it consists of very different types of design problems which have to be tackled simultaneously. We distinguish the following design problems: (1) the business model design, (2) the business process design, and (3) the software architecture design (Figure 1). One reason to distinguish these design processes is the separation of concerns for different stakeholders. For example, general managers will want to take decisions about business models, but usually do not wish to be strongly involved in decisions regarding the software architecture of a system. Thus, requirements creation in electronic commerce applications must be grounded in different stakeholder *views* upon the systems to be built.

The view-based approach upon requirements engineering is by now well established, see e.g., [12]. An important open issue, however, in electronic commerce is what views are relevant (our answer being given in Figure 1), and how these views are to be developed.

The top-level view of our electronic commerce framework concerns the *electronic-commerce business model*. The business model describes the way of doing business between actors, and so sparks off requirements at a business level. Stakeholders are general managers of companies participating in the execution of the business model, marketers and customers. Business developers are the primary designers of the model. An important reason to consider business models in some detail is that they are useful to analyze and solve tradeoffs between varying, and potentially conflicting, stakeholder interests. Tradeoffs appear in the separate design processes and should be addressed during these processes. However, tradeoffs may also exist between the business model, business process and software architecture.

For instance, fraud constraints (as they are called at the business level) or security requirements (as they are called on the software architecture level) can either be addressed during business model design by moderating potential conflicts of interest, during business process design by creating a fraud-prevention protocol or by a software architecture design by applying secured components based on encryption technology, or all of these [7].

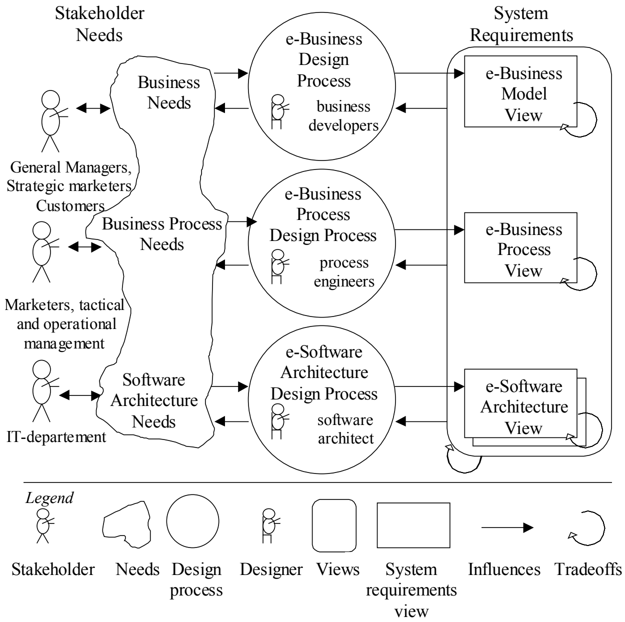

Figure 1 E-business design processes produce different stakeholder views on system requirements

Design of business models is, therefore, a process which needs guidance by itself. However, there is hardly any scientific consensus or sound method how a business model should be represented. How to do this is a main focus in our paper. A key idea of our approach is that structured *value* analysis is a crucial activity in business model design, and that this enables one to properly address various design requirements tradeoffs that exist across the three viewpoints depicted in Figure 1. In modelling value, we suggest that a good starting point is found in business administration literature, in particular work on value creation in micro-economic pricing theory, the value-chain concept [16] or, better, the value-constellation notion [14]. However, there is a need for a more formal representation of business models, and we propose such a representation below.

The *business process view*, the middle level in Figure 1 shows how activities should be performed and by whom. Also messages exchanged between activities performed by actors are represented. Stakeholders are managers on tactical and operational level since they are responsible for carrying out most processes, and marketers regarding detailed buy flows. Business process engineers are the most important designers. To represent a business process view a number of techniques are suitable, for instance UML activity diagrams with swimming lanes to represent actors [6], high-level Petri Nets [10], [20], or role-based process-modelling techniques [15]. The CommonKADS approach for knowledge engineering and management [18] has interesting facilities to model multiple-actor aspects of the business process view.

The *software architecture views*, the bottom of Figure 1, show the information system and its constitutive software components. In [2], a software architecture is defined as the structure or structures of a system which comprises software components, the externally visible properties of those components, and the relationships among them. Multiple architectural views are used to represent different types of structures. Four commonly used views are the logical view, process view, physical view and development view [13]. Representation techniques for software architectures are in an early phase of development. Stakeholder is the IT-department in its role of managing and maintaining systems. The software architect is the most important designer. The focus in this paper, however, is the representation of business models and how this representation can be exploited to derive architectural requirements.

2.1. Business model view: core concepts

We propose that the central concept in any business model is that of a value activity. A value activity is performed by actors and aims at producing material or immaterial objects that are of value to others. This notion of value activity is recognised in, e.g., [16], [14], [11]. Value activities as specified in [16] can be connected to form a value chain. At the macro-level, we can use these concepts to specify a business model. However, from micro-economics theory, interesting concepts can be borrowed in the field of pricing theory [5], [9], [19]. In particular, these authors consider extracting the maximum price a customer is willing to pay as one of the challenges of electronic commerce applications. They propose to do this by offering each customer a specific tailored version of a product to each customer. We use these macro- and micro-concepts, as well as our consulting experience in designing electronic commerce applications, to derive a small set of core concepts needed to represent a semi-formal business model.

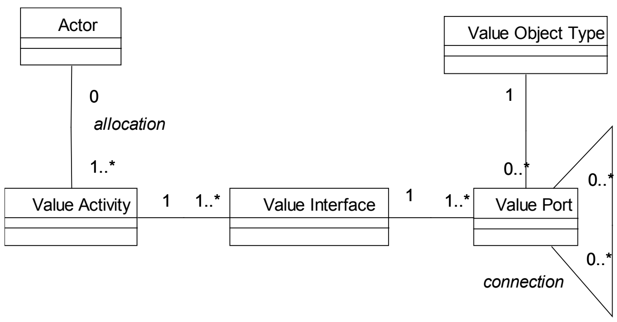

Figure 2 Core concepts to represent a business model in a semi-formal way.

Also, we define a six-step approach that provides a practical way to work with these concepts.

Our core concepts are represented by abstract UML classes (Figure 2). For a specific case, these concepts are specialised using the UML generalisation mechanism.

Actor. An actor is an independent entity such as a company or a person. An actor corresponds to company A,B,C or person X,Y,Z. Actors perform one or more value activities.

Value activity. A value activity represents a process which adds value. Actors perform these value activities. An actor can perform multiple value activities, but a particular value activity is performed by one actor only. When developing business models, we are primarily interested in finding chunks of activities that add value and in studying the various possible assignments of these activities to different actors. These reflect important business decisions. However, we are not yet interested in the actual way of performing these activities: this reflects the separation of concerns, and is the main concern in business process design (e-business process view in Figure 1). Value activities for a specific case are represented as specialisations of the value-activity concept. The granularity of defining value activities should be such that they can be performed technologically and economically independently from other value activities [16], *and* that they cannot be further decomposed into smaller activities that can be assigned to different actors. Instances of leaf value activities have a unique mapping onto the set of actors. Constructing these value activities and mapping of its occurrences onto actors is an important part of the electronic commerce design problem.

Value object and value object type. A value object is what is produced or consumed by a value activity. Value objects are the things that are exchanged between value activities. A value object type denotes the type of asset which is created or used by a value activity. A value object type refers to a type of (digital) good, a service type, or type of money [5], for instance token-based or notational money [4]. A value object has one value object type.

Value port. We further need a formal way to indicate how value activities can be connected to each other in a component-based and (re)configurable manner. Here, we introduce the concept of ports, a notion known from general and technical systems theory (as a helpful analogy, think of a wall outlet for electricity; it has two ports). A value port, then, denotes a connection point of a value activity that defines how it may be connected to the external world of other value activities. On a value port, value objects are exchanged. A value port has exactly one value object type. Value objects can flow into a value activity or away from a value activity via a port. This direction is modelled as a property of the value port. A value port can have various properties such as a price or price range for the value object. Note that a property such as a price is seen as a property of the port and not of the value object, because other actors may offer the same value object for a different price.

Value interface. Value ports are grouped into value interfaces. A value interface represents a commerce service offered to or requested from a value activity. It consists of at least one value port. A value interface having only one value port can be used to model a value activity which produces value objects for free. In other cases, we have two ports; one value port for the outgoing good or service to be sold and one value port for the incoming payment (not necessarily money, for instance in some cases one can pay with privacy information). Finally, one can think of more than two ports in an interface, to model the business concept of bundling [5], [19]. Bundling refers to the situation that a customer buys a number of products or services (the bundle) as a whole and pays for this bundle as a whole. A value activity may have multiple value interfaces. There are two motivations for having multiple interfaces. Firstly, a value activity typically requests (buys) value objects from actors and uses these objects to create and sell other value objects, mostly to other actors. The value activity has in this case two faces to its environment: one as a buyer and one as a seller. For each, a value interface is available defining the commerce service requested or offered. Secondly, multiple versions of equally typed value objects can be sold against different terms and in different bundles to address price and product differentiation [19], [9], [5]. Versioning, bundling and different terms are ways to implement value-based pricing. With value-based pricing, a seller tries to extract as much value from the buyer as possible, by making an offer that is targeted to the specific customer. We employ different value interfaces to model the situation that a value object is offered in different versions, bundles and with different terms since they are different commerce services. A value interface also prescribes the value ports of value activities which can be interconnected. A connection between two ports of different value interfaces can only be made if these value interfaces match. Interfaces match if for each value in-port in an interface, a corresponding value out-port in the other value interface can be found and vice-versa, and, for each set of connected value ports, the value ports have the same type. On a value interface a number of rules and constraints can be defined. For example, consider a time-ordering rule stating that a customer has to pay on a value port first and subsequently receives the good (pre-payment) or vice versa (post-payment) via another value port.

2.2. A Six-Step Approach to e-Business Model Building

The above set of value-based concepts constitutes a concise and generic starting point to develop e-business models in a semi-formal and structured fashion. To further support this, we also propose a six-step approach to business model building. These steps need not necessarily to be performed sequentially.

- ***Step 1:*** identification of the actors/stakeholders in the e-commerce process in hand.
- ***Step 2:*** construction of the list of relevant value activities.
- ***Step 3:*** definition of the associated value ports, interfaces, and value object types.
- ***Step 4:*** allocation of the value activities to the actors, including sensible alternative ways to do this.
- ***Step 5:*** analysis of the tradeoffs occurring in the alternative business models ensuing from the first four steps.
- ***Step 6:*** tracking down the associated implications for requirements on the information systems architecture.

How value-based business model development step-by-step works in practice is illustrated in the next sections that present two different industrial case studies.

3. Case A: The Ad Association

The Ad Association is a company which co-ordinates more than 150 local free ad papers called FAPs. FAPs produce (non-electronic) papers with ads. They are independent, often privately owned organisations. A FAP serves a geographical region, for instance a large city or a county. The handling of ads is as follows. A customer submits an ad to a FAP. The FAP checks the ad (e.g. for absence of dirty language and for style) and places the ad in its next issue. It is possible to place an international ad. In this case, the FAP to which the ad was submitted distributes the ad to other FAPs (serving different geographical regions). These other papers publish the ad as soon as possible. Placement of an ad is for free. However, a person who wants to read an ad has to pay a FAP by buying its paper. The exchange of international ads between FAPs is nearly for free. FAPs are only charged for the use of a common infrastructure which is offered by the Ad Association. The Ad Association carefully analysed the international ads. They concluded that international ads are mostly contact ads, in which one person is searching for another person. The Ad Assocation is considering an Internet-based service for international contact ads. There are a number of business objectives which are important. First, FAPs want to protect the current market share of world-wide (paper-based) contact ads. FAPs are afraid of new parties entering the arena of international contact ads. They are especially afraid of competitors which are capable of setting up a world-wide Internet-based contact service. Ad papers want to exploit their local trusted brand names now to establish a trustworthy Internet based contact ad service before someone else does. Thus, the development of a contact service has rather defensive objectives. Second, FAPs want to enlarge the market share of ads by exploiting yet another communication channel. Third, FAP wants to attract customers to their existing ad papers by offering a full service spectrum, amongst others, placement of an ad on the Internet.

We present two business models: (1) a FAP-centred business model, and (2) an Ad Association-centred business model. In both models, we assume that one has to pay for reading an ad, whereas placement of an ad is for free. Note that it is also possible to create business models which assume that one pays for placing an ad.

3.1. A FAP centred business model

Step 1: **identification of the actors/stakeholders.** The following actors participate: contact searcher, FAPs and, Ad Association.

Table 1. Value activities, value interfaces, value ports and value object types

Value activities	Value interfaces with value ports of value object type
Place Ad	In port: Placed ad (ad) Out port: Submitted ad (ad)
Read Ad	In port: Read ad (ad) Out port: Payment for reading ad (money)
Ad intake	In port: Submitted ad (ad) Out port: Placed ad (ad)
	In port: Checked ad (ad) Out port: Payment for checking (money)
	In port: Payment for sending ad (money) Out port: Sent ad (ad)
Check ad	In port: Payment for checking ad (money) Out port: Checked ad (ad)
Publish ad	In port: Received ad (ad) Out port: Payment for receiv. ad (money)
	In port: Payment for reading ad (money) Out port: Read ad (ad)
Redist. ad	In port: Received ad (ad) Out port: Payment for receiv. ad (money)
	In port: Payment for sending ad (money) Out port: Sent ad (ad)

Step 2: **construction of the list of the relevant value activities.** Ads have to be placed and read by contact searchers. This results in the value activities *place ad* and *read ad*. Value activity *ad intake* ensures that an ad is placed internationally. Value activity *check ad* checks an ad for correct use of language. Value activity *publish ad* offers a reading service of ads to contact searchers. Value activity *redistribute ad* receives an ad from a FAP and redistributes this ad to other FAPs.

Step 3: **definition of the associated value ports, interfaces, and value object types.** In Table 1 value activities, their value interfaces, value ports and value object types are presented. Dashed lines separate different value interfaces of a value activity. Note that a value interface is defined by enumerating its value ports and their value object types.

Value activities, value interfaces and value ports are represented as specialised classes in the UML class diagram. Also relations between core concepts are specialised. For reasons of brevity, we present the extension of the core-UML diagram (Figure 2) for the value activity *publish ad* only (Figure 3). For other value activities, similar diagrams can be drawn.

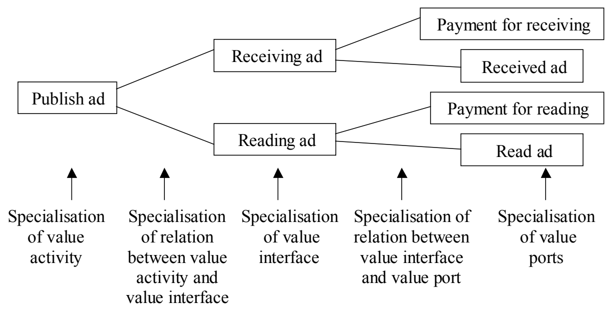

Figure 3 Specialization of core concepts into concepts for a specific business model.

Step 4: **allocation of the value activities to the actors, including sensible alternative ways to do this.** In this business model, FAPs are performing as much value activities as possible. The Ad Association only redistributes ads. These design decisions are concisely presented in Figure 4.

There are three types of contact searchers. The first type only places ads, the second type only read ads and the third type does both. In a business model, many similar actors can exist. Similar actors have (1) the same number of instances of specific value activity classes and belonging value interfaces and value ports, and (2), value ports are interconnected in the same way. Such actors are called *stacked actors* and are presented as stacked rectangles. In this business model, contact searchers as well as FAPs are examples of such actors. However, at the individual actor level, there might be differences. For instance, each FAP can have its own price for the ad to be read by searchers. Not all connections between value ports of actors are shown, especially in the case of stacked actors.

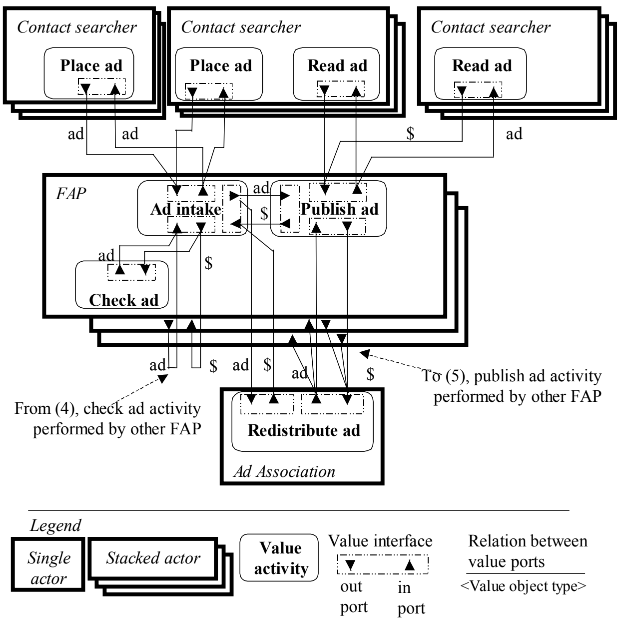

Figure 4 A FAP centred business model.

For instance, only the value ports of the topmost stacked contact searcher and the first stacked FAP are shown while each contact searcher is connected to each FAP. In some cases, value ports of stacked actors themselves are connected. For instance, a FAP can ask another FAP to check an ad. An connection between the second stacked FAP and the topmost FAP represents this. Finally, note that value activity *publish ad* has two instances of the same type of value interface. An ad can be obtained from the *ad intake* activity performed by the FAP itself, and from the *redistribute ad* activity performed by another actor. These sources can have different pricing schemes. Therefore, we use two value interfaces.

3.2. Alternative business model: Ad Association centred

This business model assumes that the Ad Association performs most value activities. The Ad Association exploits brand names of FAPs by offering an international contact ad service using brand names of each FAP. Contact searchers use the brand name of their local FAP (e.g www.localfap.com) to *find* the service *offered* by the Ad Association. The Ad Association pays each FAP a fee for the usage of brand names. Such a fee can be fixed but preferably depends on the number of ads placed and read mediated by the brand name of a FAP.

Step 1: **identification of the actors/stakeholders.** The actors/stakeholders are the same as identified in the previous section.

Step 2: **construction of the list of the relevant value activities.** A new value activity, *maintain brand*, is added to already identified value activities. This activity models that FAPs try to increase the value of their brand name by amongst others publishing a paper.

Step 3: **definition of the associated value ports, interfaces, and value object types.** In Table 2, value activities, value interfaces, value ports and value object types are presented. For brevity we only report changes.

Table 2 Specific value activities, value interfaces, value ports and value object types

Value activities	Value interfaces with value ports of value object type
Ad intake	…
	In port: Brand (brand) Out port: Payment brand name (money)
Publish ad	…
	In port: Brand (brand) Out port: Payment brand name (money)
Maintain brand	In port: Payment brand name (money) Out port: Brand (brand)

The value activity *maintain brand* has a value interface with a value-out port representing the brand is of value and a value-in port representing payment for using the brand. The value activities *ad intake* and *publish ad* have an additional value interface with value ports for using the brand name and paying for it.

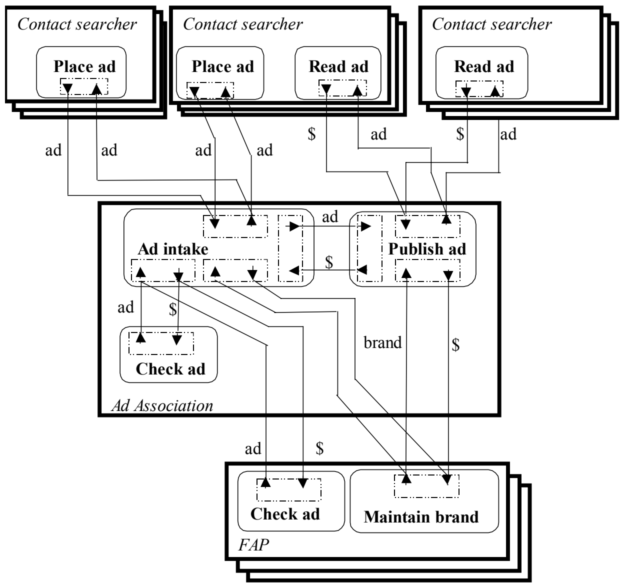

Figure 5 Ad Association centred business model

***Step 4:* allocation of the value activities to the actors, including sensible alternative ways to do this.** For reasons of brevity we skip the UML class model and show the allocation of value activities to actors directly (Figure 5).

Note that value activity *redistribute ad* is not allocated to any actor, since redistribution of ads is not necessary anymore.

3.3. Trade-offs

***Step 5:* analysis of the tradeoffs occurring in the alternative business models ensuing from the first four steps.** The previously discussed business models have nearly the same value activities, value interfaces and value ports. The most important design trade-off shows up when allocating value activities to actors. In the first business model, FAPs perform the most important value activities *ad-intake*, *check ad*, and *publish ad* themselves and only use other FAPs for checking ads expressed in foreign languages. The Ad Association has a limited task in offering a redistribution infrastructure, for which FAPs pay a small fee. In the second business model, the Ad Association is the most dominant actor. FAPs add value by (1) offering a brand name to the Ad Association which is trusted and known by searchers and, by (2) checking ads expressed in a language the Ad Association can not handle. Since the Ad Association adds far more value in the second business model, we expect that FAPs will receive fewer revenues compared to the first model. However, the amount of work for FAPs also decreases compared to the first business model. Note that if the business model is implemented well, the contact searcher does not notice anything from the choice made for the first or second business model.

***Step 6:* tracking down the associated implications for requirements on the information systems architecture.** A number of implications for the information system architecture can be drawn from the business models.

- The FAP centred business model requires a component that redistributes ads to other FAPs while the Ad Association business model does not need such a component.
- The Ad Association centred business model needs an accounting system which administrates the use of brand names of individual FAPs. If FAPs are paid on a per-ad placed/read basis, such a system should be able to relate brand names to ads placed and read. Moreover, such an administration should be trusted by the FAPs.
- The FAP centred business model supposes a number of components which should be implemented by each FAP, such as a web server, a databaseserver containing ads, application components for reading, placing and checking ads, and a high quality connection to the internet to be reachable for searchers. In the Ad Association business model, these components can be centralised. Especially, it is possible to invest in a high bandwidth/low latency internet link.
- In the FAP centred business model, each FAP should be capable of handling payment. Setting up a payment infrastructure can be rather costly in some countries. In the Ad Assocation centred business model, only one payment component is necessary.

A number of alternative business models exist which, due to space limitations, have not been considered. However, the core concepts as well as the six steps provided guidance in charting various business models and trade-offs. Also, it is possible to identify some system architectural requirements.

4. Case B: Value-Added Services in the Energy Business

Until recently, energy utilities mainly acted as regional monopolies. Their main process has been to deliver energy and electricity for a fair price. In many countries in Europe as well as the US, utilities are being privatised, the energy market is being deregulated, so that in time customers will have a free supplier choice, mergers and acquisitions have become commonplace, and international competition is increasing. The old and simple business model of effective energy delivery in a monopolistic regional market therefore is quickly becoming obsolete, and, as a consequence, utilities are investigating new business models and strategies. Both cost and differentiation strategies [16] are considered. A cost- and product-oriented strategy is attractive for some companies, because it remains close to the business-as-usual situation before the deregulation, and is therefore well-known and perceived as not very risky. However, margins related to this strategy have historically been low (2-3%), and deregulation-driven price competition will further erode these margins.

The alternative, a differentiation strategy, must be customer-oriented, requires significant innovation, and thus leads into uncharted and uncertain waters. However, recent developments in information and communication technology enable a wide range of new utility, customer, and telecom services based on two-way communication with the customer [1]. Relevant new technologies include communication over the power line as one competitor in local-loop telecom access technologies, associated new ways of fast internet access (also directly over the power line), home networking and home automation leading to the “smart” home [8], and advanced software technology enabling value-added electronic energy services, varying from automated metering and billing, home security at a distance, energy load management [22], and agent-based climate control [21]. There are many opportunities as well as uncertainties here, leading to a clear need for e-business model design and analysis. In this paper, we will select one example of a possible new electronic service in the energy area, namely, what one might call *"e-managed comfort"*.

4.1. Business models for e-managed comfort

The new service idea behind e-managed comfort is the following. Rather than just delivering electricity and energy for a fixed tariff, a utility may define its offering as delivering comfort in the home at the lowest possible cost for the customer. Production prices for energy strongly fluctuate over the day, so the real-time/time-of-use price of distributed energy is not constant (as suggested by fixed tariffs), but actually fluctuates as well. Hence, there is a financial incentive to manage the energy consumption of appliances over time if possible (a concept known as power load management). Indeed, quite some appliances are manageable in the sense that it is possible to shift their energy use in time (away from periods with peak prices) within certain tolerance limits without compromising their function. This holds, for example, for heating and cooling equipment (slow distributed processes, responsible for most of the energy use in households), battery-charging electrical equipment, dish and laundry washers, and so on. Moreover, due to new technology this load management can nowadays be done automatically and from a distance, combining microprocessor and software (e.g., agent) facilities within "intelligent" equipment, and IP-based telecommunication between equipment over the power line and other media: the *smart home*. This time-of-use energy load management can lead to significant savings both for the utility and the customer, on average in the order of 10% to 15%. So, e-managed comfort is a potentially interesting concept, but, compared to energy business as usual, obviously requires new business models as well as a new utility-home information architecture and technology. Using the ontology [3] sketched in Sec. 2 for value-based business modelling and the associated *e^3-VALUE* six-step approach, we will now study some of the possible new e-business models in energy.

Step 1: **identification of the actors/stakeholders.** There are different situations to consider in different countries, but to keep things simple in the context of this paper, we only consider two actors here: (a) the utility; (b) the (household) customer.

Step 2: **construction of the list of the relevant value activities.** Business as usual essentially involves two value activities: *energy delivery* by the utility; *energy use and payment* by the customer. Managed comfort requires additional value activities that all deal with some form of (valuable) information: *energy usage forecasting and planning*; *delivering real-time energy price information*; *home comfort control actions*. One may add: *utility-customer contract design*, as this is a non-trivial activity setting the context, boundaries and constraints of all other value activities.

Step 3: **definition of the associated value ports, interfaces, and value object types.** In the present case, this is relatively simple. Most value activities have one value-in port and one value-out port, but the type of the associated value objects differs. In the *energy delivery* activity, there is a value-out port with energy as its value object type and a value-in port with money as its value object type. In the *energy usage and payment* activity, it is the same with the in and out port direction being switched. Together, these two value activities provide the traditional bottom-line business model of an energy utility. So the other value activities are indeed value-added activities with respect to the bottom-line business model. Their in- and out-ports all refer to different forms of information as value objects. For example, the *home comfort control actions* value activity, has an out-port with control signals (a specific type of information) as value object types, and an in-port with an equipment load management schedule (another specific type of information) as its value object type. This schedule is the value object type at the value-out port of the *energy forecasting and planning* value activity. Analogous considerations hold for the other value ports and objects. For brevity, we further refer to Figure 6.

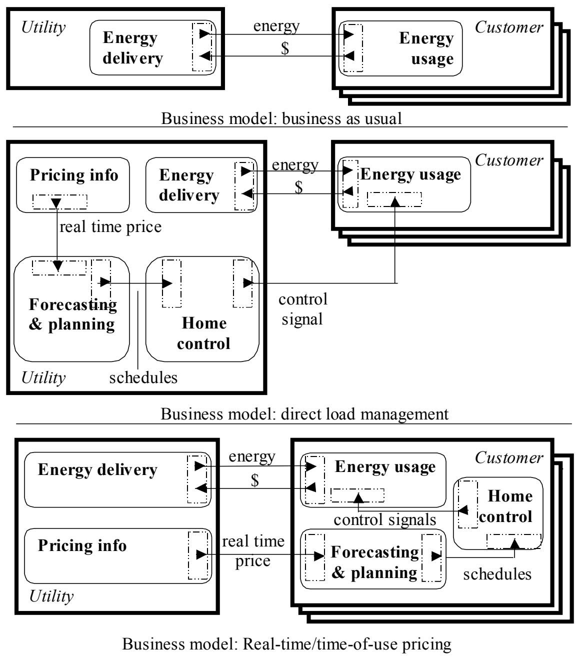

Figure 6 Business models for e-managed comfort.

Step 4: **allocation of the value activities to the actors, including sensible alternative ways to do this.**

This simple set of six value activities leads to at least three interesting and very different business models, depending on how activities are allocated:

1. *Business as usual:* this business model contains only the two traditional value activities, energy delivery and usage/payment.
2. *Direct load management:* the added-value activities handling information and control are predominantly allocated to the utility. In particular, the utility carries out the control actions.
3. *Real-time pricing:* these added-value activities are predominantly allocated to the customer. The customer remains "behind the steering wheel", the utility supports this by providing pricing information.

The associated value-configuration business models are depicted in Figure 6.

Step 5: **analysis of the tradeoffs occurring in the alternative business models ensuing from the first four steps.** This is concisely done in Table 3.

Table 3 Trade-offs between business models.

Business as usual	Direct load management	Real-time pricing
+: customer convenience	+: customer convenience	+: max. customer latitude
-: no pain, no gain	+: flat rate contract with rebate	+: simple contract with own control over savings
	-: contract conditions design	-: customer has process burden
	-: external control over customer: privacy/security	-: home automation technology investments

Step 6: **tracking down the associated implications for requirements on the information systems architecture.** The different business models of Figure 6 already show different requirements for the information architecture. Briefly:

- The direct load management scheme requires major functional components in the architecture, such as forecasting/planning functionality, to be at the utility, whereas in the real-time pricing business model these functions must reside at the customer's premises. The latter in turn requires a certain mature level of home networking and automation that, for example, would not have been technically feasible ten years ago. The business models also differ with respect to the real-time requirements they generate (real-time price vector handling vs. off-line contract design, for example).
- In addition, the sub architecture of a forecasting and planning function is very different at a utility or a customer, because it refers to very different situations and handles very different information. So, these different business models lead to different architectures at points already identifiable at the business level. Here, the allocation of value activities to actors directly influences the architecture.
- An interesting conclusion follows from looking at the utility-customer interface in Figure 6 in the various business models. In contrast to the real-time pricing business model, direct load management involves control from the utility over customer equipment. It is known from market studies that this is an important point for household customers. Thus, it requires special privacy and security facilities in the architecture of direct load management that are not needed in the real-time pricing scheme.

Although this case study is too brief to cover all relevant considerations, it does indicate that a value-based requirements approach (1) helps to make initial requirements creation into a more structured process, and (2) supports business decision-making by clarifying various possible value propositions to customers and their tradeoffs.

5. Conclusions

Requirements engineering for electronic commerce applications involves a first stage of requirements creation, rather than that system requirements can be elicited or acquired. The reason for this is that in e-commerce the business models and processes that underlie the supporting information systems are also new and have to be designed simultaneously.

We have put forward an approach to requirements creation for e-business that integrates both business and technology considerations. Discussed key aspects of our e^3-*VALUE* approach are:

- Requirements creation should start from structured value-based analysis at the business level, within the relevant actor/stakeholder network in electronic commerce application.
- To this end, we have proposed a novel, semi-formal way to define a business model. We have identified a limited and generic set of core concepts (i.e. the basis of an ontology) to construct a business model, including the notions of actors, value activities, objects, ports, and interfaces. These concepts and their relationships can be defined with the help of modelling techniques such as UML class diagrams.
- In addition to this value-based ontology, we have also proposed a practical six-step procedure for constructing alternative business models in a piecemeal and structured fashion.
- The resulting value-configuration models enable one to analyse tradeoffs between different business models, and at the same time to generate high-level

requirements regarding the electronic commerce systems architecture.

How our *e^3-VALUE* approach practically works has been demonstrated by means of two different real-life case studies, one from web-based advertising and one from new e-services in the energy industry. In sum, there is a definite need for an integral design view upon relevant business and technology matters in electronic commerce applications. Value-based business modelling, as developed in this paper, is a powerful tool to achieve such an integrated approach to business model decision-making and systems requirements engineering.

References

[1] J.M. Akkermans, D. Healey, and H. Ottosson, *The Transmission of Data over the Electricity Power Lines, Advisory Report to the European Commission*, 1998.

[2] L. Bass, P. Clements, and R. Kazman, *Software Architectures in Practice*, Addison-Wesley, Reading, Massachusetts, 1997.

[3] W.N. Borst, J.M. Akkermans, and J.L. Top, "Engineering Ontologies", *International Journal of Human-Computer Studies,* vol. 46, 1997, pp. 365-406.

[4] L.J. Camp, *Privacy & Reliability in Internet Commerce*, Carnegie Mellon University, School of Computer Science, Pittsburgh, 1996.

[5] S.-Y. Choi, D.O. Stahl, and A.B. Whinston, *The economics of doing business in the electronic marketplace*, Macmillan Technical Publishing, Indianapolis, 1997.

[6] M. Fowler and K. Scott, *UML Distilled - Applying the Standard Object Modeling Language*, Addison-Wesley, Reading, Massachusetts, 1997.

[7] J. Gordijn and J.C. van Vliet, "On the Interaction between Business Models and Software Architecture in Electronic Commerce", *in Addendum to the proceedings of the 7th European Software Engineering Conference/Foundations of Software Engineering*, ed. M. Lemoine, Toulouse, 1999.

[8] R. Gustavsson, "Agents with Power", *Communications of the ACM ,* vol. 42, Mar, 1999, pp. 41-47.

[9] J. Hagel III and A.G. Armstrong, *Net Gain - Expanding markets trough virtual communities*, Harvard Business School Press, Boston Massachusetts, 1997.

[10] W.J. Hofman, *A conceptual model of a business transaction management system*, UTN Publishers, 's Hertogenbosch, 1994.

[11] R.S. Kaplan and D.P. Norton, *The Balanced Scorecard*, Harvard Business School Press, Boston, Massachusetts, 1996.

[12] G. Kotonya and I. Sommerville, *Requirements Engineering - Processes and Techniques*, John Wiley and Sons Ltd, Chichester, England, 1998.

[13] P.B. Kruchten, "The 4+1 view model of architecture", *IEEE Software,* vol. 12, Nov, 1995, pp. 42-50.

[14] R.R.R. Normann, *Designing Interactive Strategy - From Value Chain to Value Constellation*, John Wiley & Sons, Chichester, 1994.

[15] M.A. Ould, *Business Processes - Modelling and Analysis for the Re-engineering and Improvement*, John Wiley & Sons Ltd., Chichester, England, 1995.

[16] M.E. Porter and V.E. Millar, "How information gives you competitive advantage", *Harvard Business Review,* Jul-Aug 1985, pp. 149-160.

[17] C. Potts, "Software Engineering Research Revisited", *IEEE Software,* vol. 10, Sep, 1993, pp. 19-28.

[18] A.Th. Schreiber, J.M. Akkermans, A.A. Anjewierden, R.d. Hoog, N.R. Shadbolt, W.v.d. Velde, and B.J. Wielinga, *Knowledge Engineering and Management - The CommonKADS Methodology*, MIT Press, Boston, MA, In press.

[19] C. Shapiro and H.R. Varian, *Information Rules*, Harvard Business School Press, Boston, Massachusetts, 1999.

[20] K.M. van Hee, *Information systems engineering - A formal approach*, Cambridge University Press, Cambridge, 1994.

[21] F. Ygge and J.M. Akkermans, "Decentralized Markets vs. Central Control: A Comparative Study", *Journal of Artificial Intelligence Research,* 1999.

[22] F. Ygge, J.M. Akkermans, A. Andersson, E. Boertjes, and M. Krejic, "The HOMEBOTS System and Field Test: A Multi-Commodity Market or Predictive Power Load Management", pp. 363-382, *in Proceedings of the Fourth Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology PAAM'99*, The Practical Application Company Ltd., Blackpool, UK, 1999.