

Assessing Feasibility of IT-enabled Networked Value Constellations: A Case Study in the Electricity Sector

Zsófia Derzsi^{1,3}, Jaap Gordijn¹, Koen Kok^{1,2}, Hans Akkermans¹, and Yao-Hua Tan³

¹Free University, FEW/Business Informatics, De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands. (derzsi, gordijn, elly)@few.vu.nl,

²ECN, Intelligent Energy Grids, PO Box 1, Petten, The Netherlands. j.kok@ecn.nl,

³Free University, FEWEB, De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands. ytan@feweb.vu.nl

Abstract. Innovative networked value constellations, such as Cisco or Dell, are often enabled by Information Technology (IT). The same holds for the Distributed Electricity Balancing Service (DBS), which we present in this case study. To explore *feasibility* of such constellations while designing them, we need at least to develop a *financial* and *technical* understanding of the constellation at hand. In this paper, we take a multiple perspective approach, by taking a *business value* perspective (using *e³-value*) and an *information system* perspective (using UML-deployment diagrams) on the case at hand. We present a novel, structured approach to relate both perspectives, thus enabling a financial and technical feasibility assessment of the constellation, using a real-life case study in the field of electricity supply and consumption.

1 Introduction

Companies increasingly form networked value constellations to jointly satisfy complex needs. Well known examples include the networked business model of Cisco Systems [17] -actually consisting of a series of well integrated companies-, and the virtual integration of Dell Computers [11]. In a networked value constellation, enterprises use each other core-competencies to offer a product or service that each individual enterprise could not offer on its own.

Such a constellation requires more than just a few enterprises alone, co-producing things of economic value. To coordinate their processes properly, information and communication technology is indispensable. Actually, for the distributed electricity balancing service case study, as to be presented in this paper, information technology is a prerequisite.

To our consulting experience [3, 7, 9], one of the issues in designing a networked value constellation is first to find a constellation (in terms of participating enterprises, cross-organizational processes, *and* interworking information systems) that seems to be *feasible*. Such an explorative feasibility assessment

track should be done in a light-weight fashion, to be able to develop a comprehensive, yet global, understanding of the constellation at hand, within a reasonable timeframe (time-to-market is typically just a few months). Such an understanding, while shared and agreed upon by the enterprises involved, can then provide further direction for a more detailed and focused requirements engineering and system design track.

In this paper, we consider two types of feasibility: (1) *economic* feasibility and (2) *technical* feasibility. Economic feasibility refers to the question whether *all* enterprises participating in a constellation can be economically sustainable over a reasonable period of time with respect to their participation in the constellation. It is then important to know *substantial* economical effects (in terms of expenses, investments, and revenues). Technical feasibility is about the question whether we can find an acceptable solution to put the value constellation into operation e.g. by deploying information technology (the focus of this paper) and inter-organizational business processes.

In this paper we combine two modeling techniques (e^3 -value and UML deployment diagrams) to reason about feasibility. An e^3 -value model has constructs for reasoning about financial feasibility by definition; however, since important financial effects can come from investments and expenses in IT, we feed the e^3 -value model by financial annotations of a UML deployment diagram. Another contribution of this paper is that we show that a value model and deployment model, if correctly related, can be used to reason about scalability issues, which are of importance while considering technical feasibility.

This paper is structured as follows. In section 2 we introduce how to explore feasibility of networked value constellations and in section 3 we present our case study-based research approach. A first step is to explore the networked constellation from an economic value perspective (section 4); a second step is to understand the information system perspective (section 5) of the case at hand. In section 6, both perspectives are structurally related with each other. Finally, in section 7 we present the lessons learned and conclusions.

2 Perspectives to understand feasibility of IT-enabled networked value constellations

To our experience [7], to assess feasibility of networked value constellations, multiple *perspectives* (e.g. strategic goals, value transfers, business processes, and information systems) need to be considered. We consider the following perspectives, amongst others inspired on frameworks such as TOGAF or Zachman [16, 1]. The *strategic goal* perspective represents the long-term objectives of enterprises, such as cost leadership or differentiation of products and services (see e.g. [15]). To explore feasible networked value constellations, it is important to know that the individual objectives of participating enterprises are aligned, and that no crucial conflicts exist. For the paper at hand, we do not elaborate on this perspective; instead the reader is referred to [13, 10, 8, 19]. The *economic value transfer* perspective explores *what* enterprises offer of *economic value* to each

other, and request *what* in return. The value transfer perspective shows how the strategy is put into operation. For feasibility understanding, this perspective is useful to assess *economic* sustainability, in terms of in-going and out-going money flows. In this paper, we employ the *e³-value* [7] approach for representing the value transfer perspective (see section 3.1); other possibilities are BMO, [12], or REA [6]. The *business process* perspective shows *how* value transfers are carried out (e.g. time ordering, parallelism), by processes, including coordinating processes between enterprises. For feasibility purposes, this perspective is usable to understand *economic* feasibility (since resources such as workers cost money). In this paper, we do not explore this perspective further; the reader is referred to [14, 18]. The *information system* perspective presents the software and hardware components, their communication, etc. In fact, this perspective may contain many sub-perspectives, depending on the modeling aim, and contributes to the understanding of *economic* feasibility (e.g. IS-components require investments and expenses for maintenance). Additionally, the perspective may give a clue regarding *technical* feasibility; whether it is possible to design an information system that satisfies the requirements expressed by e.g. the economic value transfer and business process perspectives. In this paper, we use UML [2], and as we will motivate later on, specifically deployment diagrams to capture the information system perspective.

3 Research approach

3.1 The *e³-value* methodology for economical feasibility

To evaluate feasibility in this paper, we employ *e³-value* and UML-deployment diagrams. To make this paper self-contained, we briefly introduce the *e³-value* modeling concepts below as well as the *e³-value* way of reasoning about economic feasibility (see for a more detailed explanation [7]). The *e³-value* methodology provides modeling constructs for representing and analyzing a network of enterprises, exchanging things of economic value with each other. The methodology is ontologically well founded and has been expressed as UML classes, Prolog code, RDF/S, and a Java-based graphical *e³-value* ontology editor and an analysis tool, which is available for download (see <http://www.e3value.com/>) [7]. In the following text, we use an educational example (see Figure 1) to explain the ontological constructs.

An *actor* is perceived by his/her environment as an economically independent entity. The Store and Manufacturer are examples of actors. Actors exchange *value objects* (e.g. Money). A value object is a service, a good, money, or even an experience, which is of economic value for at least one of the actors. An actor uses a *value port* to provide or request value objects to or from other actors. Actors have one or more *value interfaces*, grouping value ports, and showing economic reciprocity. So, in the example, Goods can only be obtained for Money and vice versa. A *value transfer* is used to connect two value ports with each other. In the example, a transfer of Good or Payment are both examples of value transfers. A *value transaction* groups value transfers that all should happen, or none at all.

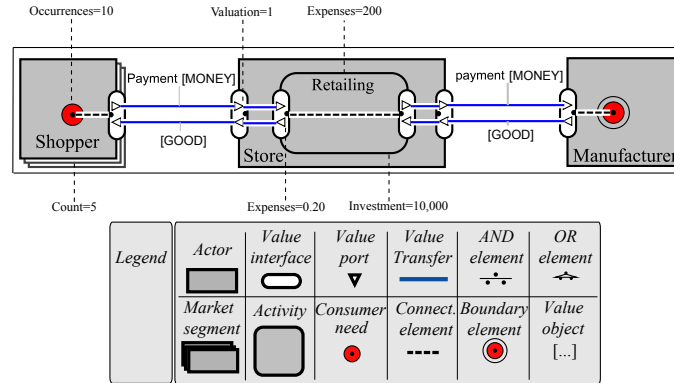


Fig. 1. Educational example

A *market segment* composes actors into segments of actors that assign economic value to objects equally. The Shopper is a market segment, consisting of a number of individual shoppers. An actor performs one or more *value activities*. These are assumed to yield a profit. In the example, the value activity of the Store is Retailing. A *dependency path* is used to reason about the number of value transfers as well as their economic value. A path consists of *consumer needs*, *connections*, *dependency elements* and *dependency boundaries*. A consumer need is satisfied by exchanging value objects (via one or more interfaces). A connection relates a consumer need to a value interface, or relates various value interfaces internally, of a same actor. A path can take complex forms, using AND/OR dependency elements taken from UCM scenarios [5]. A dependency boundary represents that we do not consider any more value transfers for the path. In the example, by following the path we can see that, to satisfy the need of the Shopper, the Manufacturer ultimately has to provide Goods.

An *e³-value* model can be attributed with financial numbers (e.g. the number of occurrences of consumer needs, the size (count) of a market segment, and the valuation of objects transferred), with which Net Value Flow Sheets (NVF) can be generated (for a free software tool see <http://www.e3value.com/>). Such sheets show the net cash flow for each actor involved and are a first indication whether the model at hand can be commercially successful for each actor. In the example, the Store has $5 * 10 = 50$ transfers with the Shoppers, so the incoming money stream = €50,-*1 =€50,-.

It is also possible to add various kinds of *expenses* and *investments*. These are cash-out flows that are significant, but for which it is not important to understand the actor receiving the expenses. Additionally, expenses are the hook to include financials obtained from other modeling perspectives (e.g. UML deployment diagrams) into the financial picture. In the example, the activity Retailing has a *fixed expense* (meaning independent from the number of transfers handled) of €200,-. Moreover, there is a *variable expense* of €0.20,- per outgoing Good

transfer (so for this case the expense is $50 * \text{€}0.20, = \text{€}10,-$. The e^3 -value ontology is capable of *assigning* the expenses related to activities to the *performing* actor automatically. Moreover, fixed and variable expenses can also be attributed to actors directly. It is also possible to include investments. For this purpose, it is important to understand that a *single* e^3 -value model represents the financial effects for a certain *timeframe*, say a day, month, or year. A *series* of potentially different e^3 -value models can be combined into an e^3 -timeseries model to consider a number of timeframes (say many years). An investment is then actually an one-time (often upfront) expense in a specific timeframe that does not occur in other timeframes. In this example, there is an upfront investment of $\text{€}10,000,-$. Summing up the financial effects of multiple timeframes is done via the Discounted Net Present Cash Flow (DNPC) technique [4]. This results in a, hopefully positive, financial number representing the net financial effects for an actor, thereby accounting properly for the time-value of money.

3.2 Explorative case study: Distributed Balancing Services

In this paper, we explore a model-based way of assessing feasibility of networked value constellations, by taking a value transfer and information system perspective (the other perspectives are also important but simply not in the scope of this paper due to space restrictions). Our ultimate goal is to arrive at a set of relevant, well integrated models that allows for feasibility studies. In this paper, we use a case-study on electricity supply and consumption (see [9]).

Due to the physical nature of electricity power, the amount of electricity supplied to the network must be *exactly equal* to the amount of electricity consumed, including inevitable transport losses. This balance has to be maintained at every instance otherwise power outages will occur. This requirement is at all time ensured by the Transmission System Operator (TSO) The TSO does so by asking large consumers and generators for their consumption/production *plans* a day ahead, matches these, and returns consumption/production plans that ensure consumption and supply balance. However, at runtime there are always deviations from the plans since it is impossible to *precisely* consume/produce the amount of electricity as planned. Since deviation from the plans causes imbalance, and adequate yet costly counter measures have to be taken, suppliers/consumers have to pay a penalty fee for causing imbalance to the TSO, who by default compensates for system imbalance.

The innovative idea for the case at hand is to create an IT-enabled service that reduces real-time imbalance in a *portfolio* of generators and consumers by allowing near-real time, distributed control over the electricity production and consumption of portfolio's participants: the Distributed Balancing Service (DBS). In case of imbalance, actors are asked to change their level of production and/or consumption. Because the imbalance is reduced for the portfolio, the penalties decrease also, and thus for the suppliers/consumers participating in the portfolio. Obviously, such near real-time control is only possible using advanced information technology, giving the time-scale (minutes) and the number of actors.

The aim of this case-study for us is to explore financial- and technical feasibility assessment of an IT-enabled value constellation by considering an economic value transfer and information system perspective. We have selected this specific case-study because: (1) The constellation and the information technology for imbalance reduction has already been built. We want to focus on the *conceptual constructs* required to understand feasibility issues and not yet on the *process* of assessment itself. (2) We have access to the financial data. (3) We have access to the developers of the DBS case. (4) The DBS case relies heavily on IT. (5) The case-study is of industrial strength. For the DBS case, we first study the available materials and do interviews with the domain experts, and we construct an e^3 -value model of the DBS case (see section 4). Also, we construct a UML deployment diagram of the DBS (see section 5). By annotating the UML deployment diagram, such that financials related to information systems can be represented, *and* by structurally relating these annotations to the elements in the value model, we derive comprehensive (discounted) net value flow sheets for both perspectives. For relating the value and deployment perspectives, we present a sub-ontology (see section 6). Finally, we reason about *technical feasibility* using the presented models with an emphasis on *scalability*.

4 An e^3 -value model for the distributed balancing service

Figure 2 shows an e^3 -value model for the DBS case study. The focus is on the participating enterprises and what they *transfer* of economic value, and not on the required soft- and hardware components yet.

There are ‘electricity generators’ in the form of ‘wind turbines’, ‘Combined Heat Power generators’ (CHPs) and ‘emergency generators’. All these generators offer ‘electricity’ and request ‘money’ in return. Different types of generators exist because, due the nature of the generator (volume of total electricity power, predictability of this volume), the pricing schemes may be different. Additionally, they offer ‘operational flexibility’, meaning that a portfolioholder (here the ‘supplier’) may influence the amount of electricity production, in return for ‘money’. There are also ‘consumers’ who buy ‘electricity’ and pay ‘money’ in return. Also, they offer ‘operational flexibility’ so that a portfolioholder can influence their amount of electricity consumption, and they request some ‘money’ in return for that. Normally, the ‘generators’ and ‘consumers’ must also pay a fee to the ‘Transmission System Operator’ (TSO), if their real-life production/consumption deviates from their forecasted production/consumption (which is always the case). This *balance-responsibility* is in the DBS e^3 -value model taken over by a ‘supplier’ of which we have one. The ‘generators’ and ‘consumers’ are all in the portfolio of the ‘supplier’. The ‘supplier’ pays a penalty (‘money’) to the TSO for the amount of imbalance caused. This amount can be reduced by controlling the ‘generators’ and ‘consumers’ near real-time. Finally, there is a ‘wholesale market operator’. The role of this operator is to sell electricity to the ‘supplier’ in case of shortage or to buy electricity from the ‘supplier’ in case of a surplus.

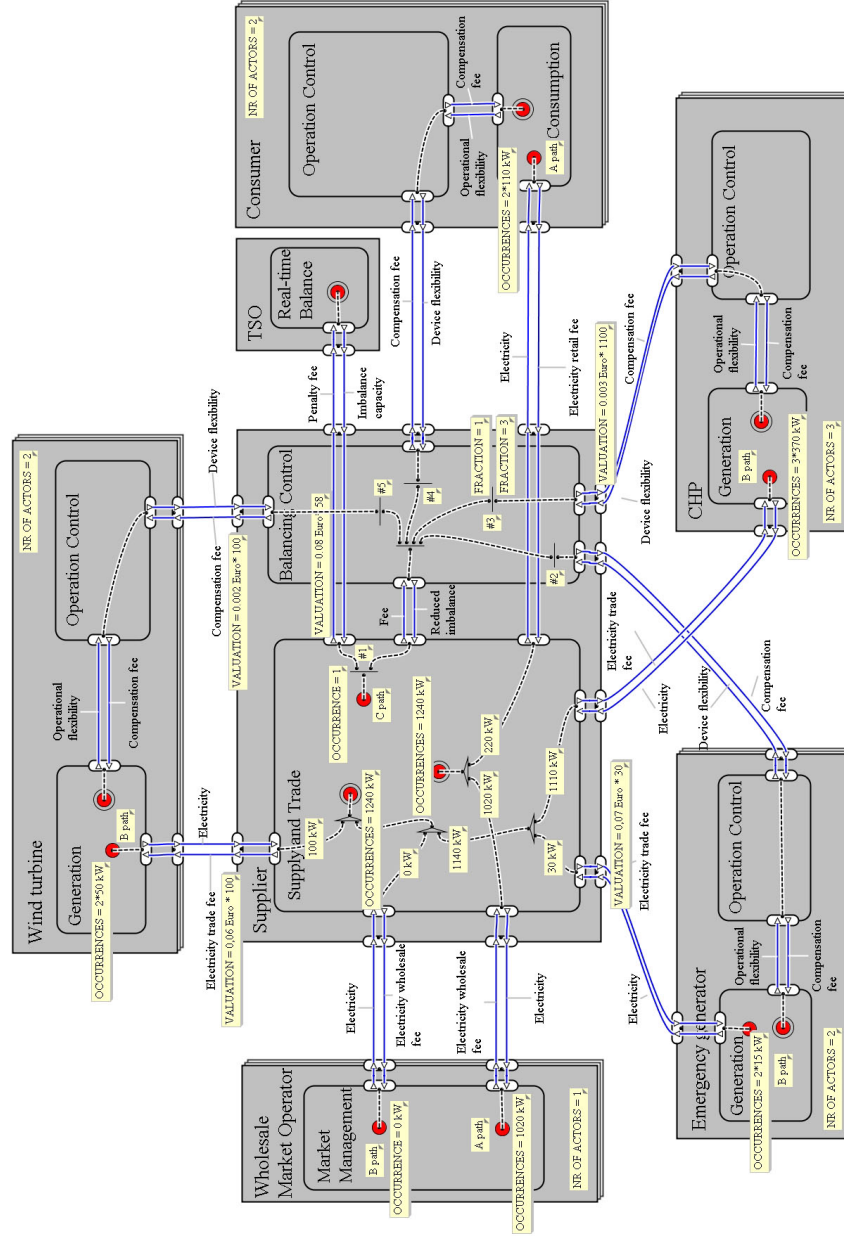


Fig. 2. e^3 -value model of the Distributed Balance System

An e^3 -value model actually provides a snapshot of the number of value transfers for a certain timeframe; here, for 15 minutes. In the DBS-case, this 15 minutes is used as a discrete interval to calculate fees, based on the actual production/consumption. In addition, all *needs* occur within this timeframe. Now, trace through the ‘A’ dependency path. As can be seen, the ‘consumer’ has a need for a certain amount of kilowatt-hours (kWh). The ‘wholesale market operator’ has also a need for electricity. These needs are satisfied by the ‘supplier’. He buys electricity from the ‘generators’ of his portfolio, and from the ‘wholesale market operator’ in case of a shortage, as can be seen from the ‘B’ path. From the ‘C’ path it can be seen that the ‘supply & trade’ activity requires ‘balancing control’, and so control of the ‘generators’ and/or ‘consumers’ in terms of operational flexibility. ‘Balancing control’ operates together with the ‘operation control’ activity, which is executed by consumers and generators. Since such a control moment is needed once per 15 minutes, for each timeframe, there will be precisely one occurrence, so one ‘operational flexibility’ transfer between the ‘supplier’ and the ‘generators’/‘consumers’. However, due to the fact that market segments aggregate actors, explosion elements are needed (fork (#2)-(#5)) in order to achieve one occurrence per actor in such a market segment. Despite the efforts of the ‘supplier’, there will always be some imbalance (because the ‘supplier’ can control *near* real-time). This is modeled by the *AND* fork (#1).

The e^3 -value model calculates, as shown, the occurrences for each dependency path element for the 15-minutes timeframe. We assume that investments in generators and in consumption control equipment were done earlier, so we do not consider these. Investments related to IT are explored in section 5. If we assign pricing schemes (valuation functions) to the model, assume an amount electricity power needed, assume a number of generators and consumers, and assume how much required electricity power can be satisfied by the portfolio’s participants (and the wholesale market), we can derive for each 15 minute timeframe net value sheets for each enterprise involved. With e^3 -timeseries, it is possible to concatenate a series of e^3 -value model snapshots, capturing many sequential timeframes of each 15 minutes. Then, a Discounted Net Present Cash Flow [4] sheet per actor can be derived to judge the financial attractiveness of the DBS, which we do not discuss in detail due to space restrictions. In table 1, such a sheet is (as an example) given for the ‘CHP generator’, including both the e^3 -value and UML-deployment perspective.

5 IS-perspective: A UML deployment diagram annotated with expenses and investments

5.1 Why a deployment diagram?

We now take an IS perspective on the DBS case. In this case study, we consider an already up-and-running system, for which the designed IS-models are available in UML. For our purpose, assessing economic and technical feasibility (in terms of scalability), we restrict ourselves to deployment diagrams for a

few reasons. (1) Deployment diagrams show, statically, *components* and *nodes* on which these components run. These components and nodes require *investments* (one-time upfront expenses) and regularly occurring *fixed expenses* (e.g. for maintenance). So, for annotating UML with financials, deployment diagrams provide sufficient handles. (2) Components themselves have *interfaces* which offer or request *services* via *ports*. Both offering and requesting services may result in *variable expenses*. ‘Variable’ means here that the expense relates to the number of service invocations; in case of a ‘fixed’ expense we have to do the expense always -even if there are zero service invocations. Service invocations are related to value transfers on the economic value transfer perspective (see section 6.1). (3) Ports offering and requesting services are annotated, e.g. with the *maximum* number of service invocations per timeframe. If the previously mentioned value transfers are related to service invocations, we can reason about scalability issues. (4) Deployment diagrams are sufficiently course-grained, so they are of use in a light-weight feasibility assessment approach.

5.2 A DBS deployment diagram

Figure 3 shows a deployment diagram (with components) for the case at hand. The ‘generators’ and ‘consumers’ all have the same, complex components derived from the value activity ‘operation control’. They consist of several sub-components, namely (a) a *computation* component, (b) a *database*, and (c) a *measuring&control* component. The computation component computes for each 15-minute timeframe a *pricing-function* that can be used to calculate, given the amount of electricity supplied/required, the price willing to obtain/pay. For this calculation, historical data from the local database is used. The measuring&control component directly influences the generation/consumption device, e.g. by adjusting the produced/required electricity power.

The ‘supplier’ managing the portfolio operates the ‘balancing control’ component, which consist of (a) a *computation* component, and (b) a *database* component. These components are used to collect the forementioned pricing-functions from each ‘generator’ and ‘consumer’. Then, supply and demand is balanced, and the ‘generators’ and ‘consumers’ are each reported back the required/consumed electricity power.

There are two services: (1) the generators and consumers offer a service that returns the forementioned pricing-function, (2) the supplier offers a service that tells the generator/consumer how much electricity they must produce/consume the coming 15-minutes timeframe by using the above pricing functions. Effectively, this controls the generator/consumer behavior.

Components are assigned to *devices* and eventually to *nodes*, being physical resources. The nodes, devices and components are *classes*, which have one or more *instances*. So, the deployment diagram tells that *one* supplier node (instance) is associated with six device nodes (instances) (consumer and generator PC’s) connected via ADSL, and with three device nodes (instances) (in this specific case, the CHP nodes) connected via wireless-UMTS. The latter

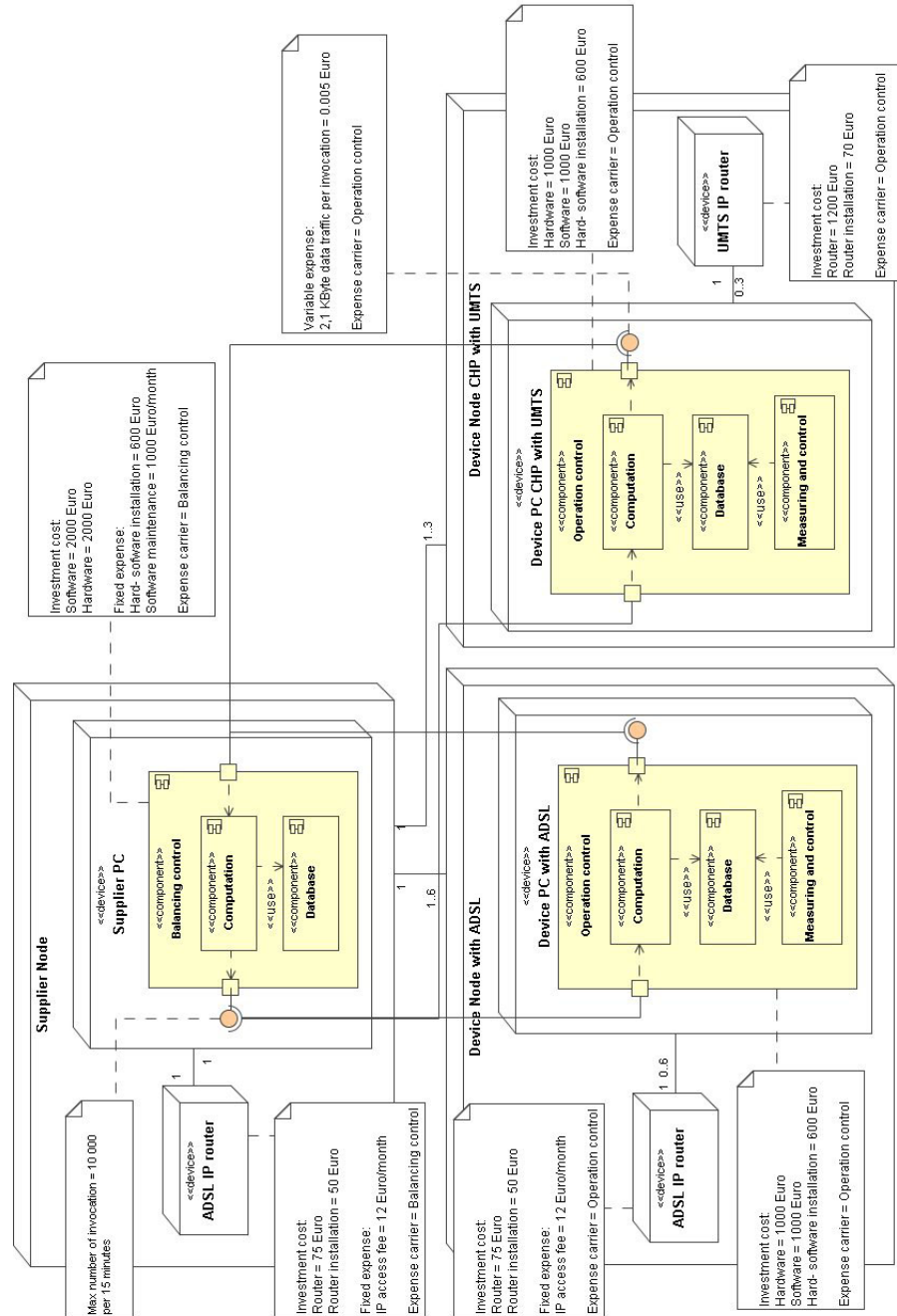


Fig. 3. UML deployment model of the DBS case.

sub-classification, based on connection technology, is motivated by the very different expense-profiles of these technologies. The number of required instances are *derived* from the e^3 -value model, by counting the number of generators and consumers.

5.3 Annotating the deployment diagram for feasibility reasoning

Figure 3 shows financial annotations to various elements in the deployment diagram. These annotations are structured along the lines of Figure 4, as an *extension* to the UML 2.0 metamodel [2] and the e^3 -value ontology [7]. Moreover, Figure 4 shows how the UML *relates* to e^3 -value. Figure 4 distinguishes various kinds of IT-expenses. *Fixed expenses* are expenses that occur once per timeframe in an e^3 -timeseries sequence of value models; *investments* occur only once per timeseries (typically these are upfront investments in equipment, software, etc., to enable future cash flow). A fixed expense is *for* precisely one *Asset*, which in UML terminology is a *Device* or a software *Artifact*; an asset *has* one or more *fixed expenses*. An expense is *assigned-to* to one *Expense Carrier*. An expense carrier is an e^3 -value Actor, *Value Activity* or *Market Segment*. This way, expenses can be assigned to business entities that create revenues to pay these expenses.

In a UML deployment diagram *Ports* attached to components are used to offer and request services from the environment. Requesting or offering a service via a port may result for each invocation in expenses themselves. The connection between requested and offered services via ports is in UML stated as an *Assembly Connection*. Such a connection *is caused by* one or more *value transfers* in an e^3 -value model. Conversely, a *value transfer causes* one or more invocations (as represented by an Assembly Connection showing the invoked and invokee port). This model-fragment allows for modeling *variable expenses*; the amount of expense is based on the number of service invocations, which in turn depends on the number of value transfers in the e^3 -value business model.

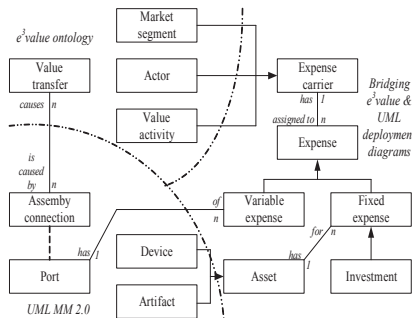


Fig. 4. Relating an e^3 -value model and a UML deployment diagram

As an example consider *investments*. The financial annotations of Figure 3 show the hard- and software costs related to components that are, as mentioned before, derived from value activities. By executing value activities (i.e. the ‘Operation control’ activity) e.g. hard- and software investments (€1,000.- and €1,000.- respectively) are required. This can be fed into the e^3 -value model for the appropriate value activity (being an *expense carrier*). The use of wireless-UMTS routers results in data-traffic accounted on a per KByte basis and thus in extra *variable expenses* each time services (using the UMTS connection) are invoked (see Figure 3). In this example, each invocation results in sending of 2,1 KB, resulting in an expense of €0.005,-. For this expense, an *expense carrier* is identified (here the ‘operation control’ activity of the ‘CHP’).

6 Relating the e^3 -value and UML deployment perspective

6.1 Financial feasibility

Financial feasibility is assessed by summing up the net cash flow (*revenues – expenses – investments*) for each actor involved over a series of timeframes. From a value transfer perspective, we consider a series of e^3 -value models, each describing a (here 15 minutes) *timeframe*, together forming an e^3 -timeseries. Each e^3 -value model, or each timeframe, contributes revenues, expenses, and possibly investments for each actor.

From an information system perspective, each timeframe may result in (fixed or variable) expenses and investments. Fixed expenses and investments of IT are directly assigned to *expense carriers* (along the lines of of Figure 4). Based on the number of value transfers (per timeframe, per actor), the amount of variable expenses is calculated that stem from IT-service invocations.

The result of the above calculation is exemplified in table 1 for the ‘Operation control’ activity that is executed by a CHP. The table normally lists all considered timeframes (here only period 0 -showing the initial investment-, and period 1 -in which the first value transfers are done- are shown, for brevity reasons). Since many sequential timeframes can be considered as equal, the number of timeframes with *different* financials is often much less. For each timeframe, first the cash transferred (both to- and from an actor) is shown as a result of doing value transfers according to the stated e^3 -value model for that timeframe. Then, expenses and investments are shown for that timeframe that result from the information system perspective. Hereafter, the net cash flow is calculated for each timeframe, just by subtracting expenses and investments from revenues. Finally, all net cash flows for all timeframes are summed up using the Discounted Net Present Cash Flow method [4], thereby accounting properly for the time value of money, cost of capital, and risk associated with participating in the constellation.

6.2 Technical feasibility

The technical feasibility assessment may contain various perspectives; here we explore *scalability* only. More specifically the question is: what happens if from

Table 1. Net value flow sheet for 'Operation control' activity of one CHP

Actor/Activity:	CHP- 'Operation control'					
Timeframe:	period 0					
					Economic Value	Total
INVESTMENT					3,870	
Timeframe:	period 1					
Value Interface	Value Port	Value Transfer	Occurrences	Valuation	Economic Value	Total
Device flexibility, MONEY			1		0.735	
	out: Device flexibility	(EXPENSES)	1	0.005	-0.005	
	in: MONEY (Compensation fee)	MONEY	1	0.002	0.74	
Operational flexibility, MONEY			1		-0.666	
	out: MONEY (Compensation fee)	MONEY	1	0.0018	-0.666	
Net Cash Flow:						0.069
Timeframe:	period 103,680 + 1					
	...					
Discounted Net Cash Flow:	1,715.12					

an economic value transfer perspective things scale up (e.g. a significant increase in consumers, or generators).

To facilitate such reasoning, we have annotated the ports in the deployment diagram with a *maximum* number of *invocations* per timeframe (here 15 minutes). We have already explained that by calculating the number of value transfers (for the economic value transfer perspective), we can calculate the number of port-invocations per timeframe. If this number is larger than the maximum number of invocations, there is a scalability issue. Perhaps it can be solved by using different hardware, but at some point, it can be possible that an entirely different architecture should be selected.

Given the numbers for the case at hand, problems occur if the size of a market segment increases since it effects the number of value transfers. As an example, let the number of consumers increase to 15,000. This results in 15,000 value transfers and so in 15,000 port-invocations of the 'balancing control' component initiated by consumers. As can be seen from the annotation of the component port (see Figure 3), this is already larger than the maximum number of invocations that can be handled by the current IS design. Obviously, this is just an example how we can reason about scalability, but it shows that for addressing a scalability issue, an integrated view on the value transfer and IS perspective is useful.

7 Lessons learned and conclusions

In this paper, we have shown how an e^3 -value model, taking an *economic* perspective on networked value constellations, can be structurally related to a corresponding UML deployment diagram, representing a *technical* perspective. As a result, a comprehensive discounted net value flow sheet can be produced for each actor involved, for the purpose to assess *economic* feasibility. Additionally, scalability, being an aspect of the larger construct, *technical* feasibility, can be reasoned about.

While doing this case study, we experienced some learnings, of which we articulate here two (due to space considerations). First, with respect to the notion of time, it is important that the timeframes as considered by the different perspectives are *indeed* about the *same* timeframe. Specifically, if the timeframe is determined by IT (e.g. timeframe of invocations), the e^3 -value model should possess the same timeframe as well. This puts certain requirements on the models constructed. Second, we have seen that -for the purpose of *feasibility* assessment-, the selection of the relevant components and nodes as distinguished by the UML deployment diagrams are influenced by the size of expenses of these components and nodes. We experienced that these deployment diagrams are sufficient for our purposes; obviously, they need to be detailed if stakeholders really decide to develop the case at hand further.

Many continuing research lines are possible. Deployment diagrams are typically constructed if information system requirements and design are already somewhat clear. Consequently, it is important to understand how e^3 -value models influence other UML-type of diagrams and vice versa (i.e. use-cases, activity / state transition diagrams, class diagrams) that are usually built in an earlier stage of requirements engineering and system design. Additionally, other aspects, specifically of *technical* feasibility need to be addressed, in conjunction with the e^3 -value model (think of flexibility, maintainability, etc.). Another line of research is the development of *guidelines* that help to make architectural decisions, which are then expressed using the integrated models we have proposed.

On the short term, we continue our research by developing a DBS to be used in Woking/UK. As the current DBS is developed for The Netherlands, we expect changes in the e^3 -value model and IS architecture for the Woking case, due to specific UK-regulations about electricity supply. We intend to use this case study to develop an integrated view on flexibility, by considering to what extent the Dutch system is usable for the UK. Additionally, we will work on an entirely different case, a ship container tracking system, to validate our proposed model-based feasibility assessment approach for IT-intensive networked value constellations.

Acknowledgements. This work has been partly sponsored by the EESD-IST funded project FENIX (518272), the JACQUARD/NWO funded project VITAL (838.003.407), and the VUA funded project VUBIS.

References

1. Togaf enterprise edition, version 8.1. <http://www.opengroup.org/architecture/togaf8-doc/arch/>, accessed november 2006. accessed november 2006.
2. Unified modeling language: Superstructure. <http://www.omg.org/docs/formal/05-07-04.pdf>, accessed november 2006. accessed november 2006.
3. J.M. Akkermans, Z. Baida, J. Gordijn, N. Peña, A. Altuna, and I. Laresgoiti. Value webs: Using ontologies to bundle real-world services. *IEEE Intelligent Systems*, 19(4):57–66, 2004.
4. Richard Brealey, Stewart Myers, and Franklin Allen. *Corporate Finance*. McGraw Hill Higher Education, 2005.
5. R. J. A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.
6. Guido Geerts and William.E. McCarthy. An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems and Their Applications*, pages 89–94, 1999.
7. J. Gordijn and J.M. Akkermans. Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.
8. J. Gordijn, E. Yu, and B. Van der Raadt. e-Service design using i^* and e^3 value modeling. *IEEE Software*, 23(3):26–33, 2006.
9. Jaap Gordijn and Hans Akkermans. Business models for distributed energy resources in a liberalized market environment. *The Electric Power Systems Research Journal*, 2007. Accepted by the Electric Power Systems Research Journal. Preprint available. doi:10.1016/j.epsr.2006.08.008.
10. Jaap Gordijn, Michael Petit, and Roel Wieringa. Understanding business strategies of networked value constellations using goal- and value modeling. In Martin Glinz and Robyn Lutz, editors, *Proceedings of the 14th IEEE International Requirements Engineering Conference*, pages 129–138, Los Alamitos, CA, 2006. IEEE CS.
11. J. Magretta. The power of virtual integration: an interview with dell computers michael dell. *Harvard Business Review*, 76(2):72–84, March-April 1998.
12. A. Osterwalder and Y. Pigneur. An ontology for e-Business models. In W.L. Currie, editor, *Value Creation From e-Business Models*, chapter 4, pages 65–97. Elsevier Butterworth-Heinemann, Oxford, UK, 2004.
13. Vincent Pijpers and Jaap Gordijn. Bridging business value models and business process models in aviation value webs via possession rights. Accepted by HICSS 2007, see <http://docs.e3value.com/bibtex/pdf/PijpersBridging2007.pdf>, 2007.
14. Vincent Pijpers and Jaap Gordijn. e3forces: Understanding the environment of networked value constellations for strategic goal and business model analysis. Submitted, see <http://docs.e3value.com/bibtex/pdf/PijpersStrategy2007.pdf>, 2007.
15. M. E. Porter. Strategy and the Internet. *Harvard Business Review*, (march):63–78, 2001.
16. J. Sowa and J. Zachman. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, 31:590616, 1998.
17. D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Nicholas Brealy Publishing, London, UK, 2000.
18. H. Weigand. On the notion of value object. volume 4001 of *LNCS*, pages 321–335, Berlin, D, 2006. Springer.
19. H. Weigand, Paul Johannesson, Birger Andersson, Maria Bergholtz, Ananda Edirisuriya, and Tharaka Ilayperuma. Strategic analysis using value modeling - the c3-value approach. Accepted by HICSS 2007, 2007.