Multi-perspective Assessment of Scalability of IT-enabled Networked Constellations

Zsófia Derzsi
VU University Amsterdam
De Boelelaan 1081
1081 HV, Amsterdam, The Netherlands
zderzsi@few.vu.nl

Jaap Gordijn
VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam, The Netherlands
gordijn@cs.vu.nl

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

Abstract

Networked constellations are often formed to provide commercial IT services by leveraging the Internet technology. The provision of IT services should remain scalable, meaning that they should remain both economically and technically deployable in different business configurations. The same requirements hold for the Distributed Balancing Service (DBS), which is the subject of our analysis. In this paper we assess potentials of system scale of a constellation during its early design phase. We address a multi-perspective analysis approach, by taking a business value (using e^3-value *) and an information system perspective (using UML) on the case at hand. We present a structured approach to relate both perspectives, thus enabling a cross-perspective assessment of system scale.*

1. Introduction

We consider commercial IT services built on the top of Internet technology as *commercial* deeds of a mostly intangible nature [7]. They rely on a substantial amount of IT for production and provision. IT services are seldom offered fully by one enterprise, rather by *networked* constellations of enterprises. Constellations consist of both *suppliers* and *consumers* of commercial IT services who jointly work on the realization of a complex, IT-driven consumer need by using each other core competencies to offer a product or service that each individual enterprise could not offer on its own. [11]. As an example for such service one can consider the wireless Internet access in hotels, which can be realized as a joint work between hotels and telecommunication companies. Well-known examples for networked constellations include Cisco Systems [11] -actually consisting of a series of well integrated companies-, and the virtual integration of Dell Computers [10].

The design of networked constellations is a non-trivial task. Different perspectives (e.g. business value perspective, information system perspective) and viewpoints (e.g. suppliers, consumers) should be taken into account during the design phase to prove their long-term sustainability [4, 6]. Constellations offering commercial IT services depend also on the underlying information system, which puts them into operation. They often harvest the advantages of Internet technology by using this medium as a communication and distribution channel. The required software and hardware components are often located at multiple, different places, which are interconnected by e.g. web service technology. Business requirements influence the design of the underlying information system. To remain financially feasible, often a certain *scale* in the number of consumers is targeted. Such a *business-driven scalability requirement* can affect the technical feasibility of the information system. A system architecture that should support five concurrent customers might look quite different from a system that must support hundreds or thousands of concurrent customers besides keeping its performance constant. In addition, requirements on scale should be matched by sufficient capacity in soft- and hardware components, resulting in additional financial consequences. In a networked business setting, the *allocation* of these costs among enterprises is vital, since it directly influences the potentials of financial feasibility of the enterprise involved.

Requirements regarding system scale can be addressed from different perspectives, and the resulting financial and technical effects of scale are interrelated. To *explore* the multi-perspective effects of scale and to *support* the design of both financially and technically feasible constellations, we address their design in this paper from two different perspectives. First, from the *perspective of business*, it is essential to show that the *commercial* IT service provision remains *economically sustainable* for *each* stakeholder in the long-term. As such, it is important to understand the business value proposition of enterprises, i.e. what enterprises offer of economic value to each other and what they request in return. Second, from the *perspective of information technology* it is then important to design a *technically feasible* information system architecture that enables the service provision and keeps a constant level in system performance independently from any changes in the business configuration [8].

In this paper we combine two modeling techniques (e^3-*value* [6] and UML [1]) to address *cross-perspective* effects of scalability. Constructs of e^3-*value* are used to describe the characteristics of the business setting. We analyze to what extend the e^3-*value* methodology can be used to articulate business-driven requirements on system scale. UML deployment and sequence diagrams are employed in order to analyze whether the underlying information system architecture fulfills these requirements. We define a *conceptual bridge* between e^3-*value* and UML to explore and pinpoint interrelated effects of system scale. The e^3-*value* methodology is suitable to reason about financial feasibility. Since important financial effects are caused by the underlying IT (investments and expenses), as another contribution of the paper, we use the developed bridge to feed the e^3-*value* model with these financials. The remainder of this paper is structured as follows. Section 2 elaborates on our model-based approach by introducing e^3-*value* . We show how e^3-*value* and UML can support our scalability analysis. Section 3.1 presents an industry-strength case study, which we use to analyze effects of scale from the business and from the information system perspectives. We explore the networked constellation providing an IT-intensive service (subject of our analysis) from its economic value perspective (Section 3.2). Then, we analyze the IT service provision from the information system perspective (Section 3.3 and Section 3.4). Section 4 shows how both perspectives can be structurally related with each other in order to explore cross-perspective effects of scale. As a result of our analysis, we show how effects of scale influence the financial and technical feasibility of constellations and, as a consequence, their design. Section 5 concludes and gives an outlook for further research.

2. Model-based research approach

To articulate and to relate the effects of system scale from different perspectives, and to explore resulting potentials of financial and technical feasibility of the system-under-study are complex problems. A comprehensive, yet global, understanding of the constellation at hand is needed, within a reasonable time frame (time-to-market of a product is typically just a few months). Such an understanding, while shared and agreed upon by each stakeholder involved, can then provide further direction for a more detailed and focused requirements engineering and system design track. Thus, an explorative feasibility assessment during the design of constellations should be done in a light-weight fashion [5, 6]. To provide a light-weight yet efficient approach we use modeling techniques that describe the networked constellation from its *business value* (using e^3-*value* methodology [6]) and its *information system* (using UML) perspectives. Modeling these two perspectives is a necessary first step if one wants to reason over the feasibility of networked constellations in the early phase of design [4]. Modeling tools, relying on sound conceptualization, support the aimed light-weight, exploration phase. Relating concepts of modeling techniques help to understand the cross-perspective effects of scale. To make this paper self-contained, we briefly introduce the e^3-*value* method. We also articulate how e^3-*value* and UML can support our multi-perspective scalability assessment.

2.1. The e^3-*value* methodology

The e^3-*value* methodology provides modeling constructs for representing and analyzing a network of enterprises, by exchanging objects of economic value with each other. The methodology is ontologically well-founded and supports the financial feasibility assessment of system scale. A Java-based graphical e^3-*value* ontology editor and an analysis tool is available for download at http://www.e3value.com/ [6]. We use the e^3-*value* methodology to visualize a snapshot of the business setting of the IT service provision for a defined *time frame*. An educational example demonstrates the ontological constructs and describes how the methodology can support our feasibility assessment (see Figure 1).

An *actor* is perceived by her environment as an economically independent entity. The 'Store' and 'Manufacturer' are examples of actors. 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. Actors perform one or more *value activities*. These are assumed to yield a profit. In the example, the value activity of the 'Store' is 'Retailing'. Actors exchange *value objects*. A value object can be

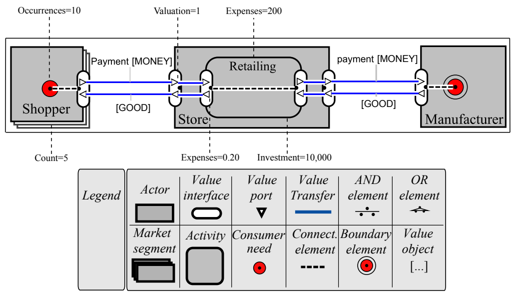

Figure 1. Educational example

a service, good or money, which is of economic value for at least one of the actors. Actors use a *value port* to provide or request value objects to/from other actors. Actors have one or more *value interfaces*, grouping value ports, and showing economic reciprocity. Goods can only be obtained for money and vice versa. A *value transfer* is used to connect two value ports with each other. It represents one or more potential trades of value objects. In the example, a transfer of 'Good' or 'Payment' are both examples of value transfers.

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 connection relates a consumer need to a value interface, or relates various value interfaces internally, of the same actor. A path can take complex forms, using AND/OR dependency elements taken from UCM scenarios [3]. A dependency boundary represents that we do not consider any more value transfers for the path. A consumer need is satisfied via value transfers, which are aggregated into the dependency path they initiate. By following the path in Figure 1, we can see that, to satisfy the need of the 'Shopper', the 'Manufacturer' has to provide 'Goods'. The dependency path does not show any time ordering with respect to value transfers. The dependency paths only show *which* value transfers are triggered if a connected value activity is executed, not the *order* nor the *behavior* of the value transfers. The number of transfers is determined by the occurrence of the modeled *consumer need* that triggers the value transfer and by the number of actors involved in the transfer. The 'Store' realizes 50 value transfers during the modeled time frame derived from the cardinality (equal to 5) of the 'Shopper' market segment and the occurrence of the consumer need (equal to 10).

When modeling constellations, the *number* of value transfers can be used to address a *requirement* on system scale that the underlying information system should accommodate. Value transfers often represent trade of commercial IT services (e.g. domestic Internet access for a monthly payment). Such value transfer is maintained by the underlying information technology, thus it forms the linking pin toward IT-supported interactions (based on e.g. web service invocations) and communication ports that realize their value. In addition, a value activity often labels an IT service provision, which is assumed to remain profitable (e.g. wireless Internet access provision in hotels). Such value activity is maintained by the underlying information technology, and forms the linking pin toward IT-supported operational activities and system components that realize their value. The e^3*-value* method enables *quantitative value assessment* by calculating the net cash flow of actors. If we assign economic value to the value objects transfered, and we count the number of value transfers, it is possible to calculate for each actor involved a net value flow sheet. Such a sheet helps to assess *financial feasibility*, since a positive net value flow is the crucial indication for economic sustainability. The e^3*-value* modeling tool possesses a built-in function to generate Net Value Flow Sheets (NVF) (for a free software tool see http://www.e3value.com/).

2.2. UML deployment and sequence diagrams

We assume, that the reader is familiar with UML deployment and sequence diagrams [1], we only identify here the benefits these techniques provide. UML deployment diagrams describe the information system architecture. They are conceptually well-grounded to provide a *structural overview* over the underlying system. We use UML sequence diagrams to characterize the interactions that occur during the joint service provision. These interactions should be then accommodated by the underlying information system. A sequence diagram shows *how* different objects interact during operation, emphasizing the *time ordering* and the *occurrence* (e.g. parallel, sequential) of interactions. As shown in Sections 3.3 and 3.4, UML modeling constructs can be annotated by different attributes in order to support a refined feasibility reasoning. Concepts taken from the UML meta model can be related to concepts of e^3*-value* methodology, enabling us to bridge the business value and information system perspectives. This facilitates our cross-perspective scalability assessment and helps to interrelate effects of system scale (see Section 4.1).

3. Multi-perspective assessment of scalability: a model-driven analysis

3.1. Case study: a commercial IT service to reduce imbalance in electricity supply

The study focuses on the imbalance reduction of electricity supply in the Netherlands (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. This balance has to be maintained continuously; otherwise power outages will occur. This requirement is ensured by the Transmission System Operator (TSO), who compensates imbalance real-time and charges an imbalance fee for the parties, who caused the imbalance.

To ensure system balance, TSO asks large suppliers and generators for their *day-ahead* consumption/production *plans* to match these and to return consumption/production plans that support balance of electricity consumption and supply. However, during 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 compensates for system imbalance.

The analyzed Distributed Balancing Service (DBS) in this case study is used to perform near real-time, distributed control over the electricity supply and consumption of commercial portfolios, consisting of a series of small-scale electricity generators (wind turbines, emergency generators) and consumers (industrial heat pumps and cooling facilities) [9] in order to reduce imbalance, and so, the penalty fee paid to TSO. In case of imbalance, consumers and/or producers of the commercial portfolio are asked to change their level of production and/or consumption. Such near real-time control is only possible using advanced, distributed information technology. All stakeholders of the portfolio have to employ certain software and hardware for execution of the DBS at their production and consumption sites.

To introduce DBS on the market for *domestic* electricity suppliers, certain requirements are needed to take into consideration. First of all, the number of aggregated domestic users is crucial to achieve the desired level of imbalance reduction. The domestic electricity consumption profile of a single household is too low compared to industrial users and shows stochastic behavior. Therefore, it is a crucial business requirement to *scale up* the number of consumers per portfolio in order to achieve a desired level of imbalance reduction and to realize the return on necessary investment. Parallel to the scale, it is important to decrease initial investment costs as well. Such business-driven scale requirement has to be supported by the underlying system architecture. Therefore, it is important to assess whether the domestic application of DBS remains both financially and technically feasible, or modifications are required in the system design.

We selected this specific case study because the information technology for imbalance reduction has already been built and applied to reduce imbalance of a portfolio aggregating industrial stakeholders. The system, however, is not yet applied on domestic level. For the study, we have access to documentations. We performed interviews with domain experts that helped to develop the models. In the following we use this case to present our model-driven analysis.

3.2. The e^3-*value* model to describe the characteristics of constellation

We have constructed an e^3-*value* business model for the case at hand (see Section 2.1 and [6]); Figure 2 shows a simplified extraction of it. The model represents the *one-time execution* of the DBS. Such execution happens in every 15 minute due to legal requirements on balance responsibility in the Netherlands, which we also selected as the time frame of the model. We now explain the e^3-*value* model by focusing on the participating stakeholders and what they *transfer* of economic value to each other.

Tracing through the ‘A’ dependency path, it shows that a ‘Consumer’ has a need for electricity (see Figure 2). The ‘Wholesale market’ has also a need for electricity. These needs are satisfied by the ‘Supplier’, who receives ‘Retail fee’ and ‘Wholesale fee’ in return for the sold amount of electricity. For supply, electricity is bought from a ‘Producer’ and from the ‘Wholesale market’ as the ‘B’ dependency path shows. The ‘C’ dependency path focuses on the execution of DBS. The ‘Supplier’ maintains ‘Balancing control’ activity, which provides ‘Reduced imbalance’ to ‘Supply’. It operates together with the ‘Operation control’ activity maintained by a ‘Consumer’, which controls his ‘Consumption’. A ‘Consumer’ offers his ‘Device flexibility’ value object as a result of ‘Operation control’ and receives ‘Compensation fee’ in return. Despites all efforts, imbalance can still occur. TSO provides ‘Imbalance capacity’ to reduce the real-time imbalance of ‘Supplier’, and receives a ‘Penalty fee’ in return (see Section 3.1).

The ‘C’ dependency path helps to count the *number* of value transfers between the ‘Supplier’ and the ‘Consumer’ market segment. Since such a control moment is needed once per 15 minutes, there will be precisely *one* occurrence, so one ‘Device flexibility’ transfer between the ‘Supplier’ and a ‘Consumer’. Due to domestic scale (see Chapter 3.1), however, in this study we aggregated 100,000 consumers forming a market segment, as it is also expressed in the e^3-*value* model. Explosion element (see fork (#1) in Figure 2) sizes up the number of value transfers according to the number of consumers in order to achieve one occurrence per consumer per market segment. As a result, the e^3-*value* model shows that there will be 100,000 value transfers initiated by the ‘Supplier’ during the modeled time frame.

If we assign *valuation functions* to value objects embedded in value transfers, assuming an amount electricity power sold and bought, assuming the amount device flexibility received and compensated, as well as the size of the remaining imbalance reduction and the penalty fee, we can

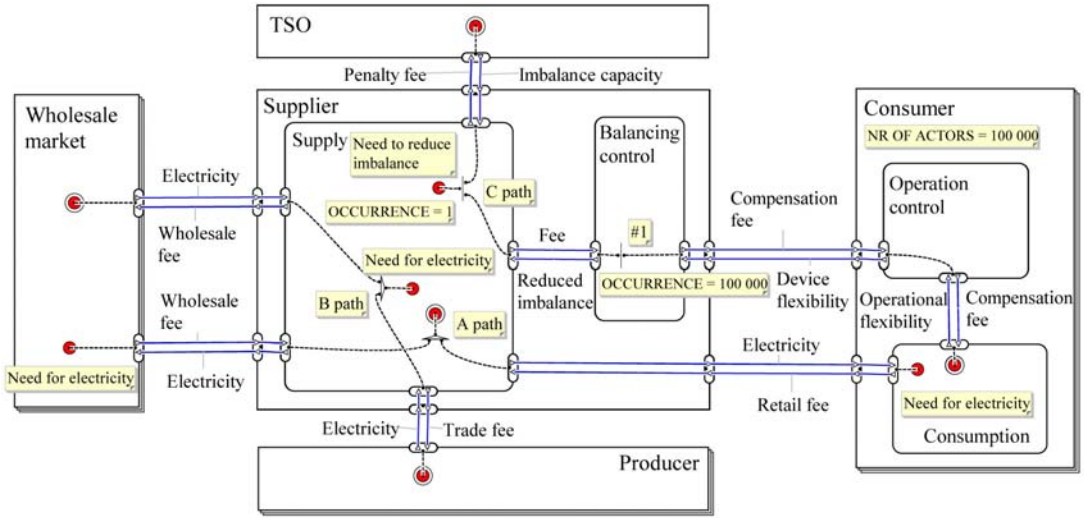

Figure 2. Business value model of the DBS case, represented by the e^3-*value* **modeling technique**

derive for each 15 minute time frame the net value sheets for each stakeholder involved. With e^3-*timeseries* (see [4]), it is possible to concatenate a series of e^3-*value* model snapshots, capturing many sequential time frames of each 15 minutes. Then, a Discounted Net Present Cash Flow [2] sheet per actor can be derived to judge the financial attractiveness of the DBS. Such a qualitative value assessment based on the detected cash flows helps to address *financial feasibility* of each stakeholder of the constellation. We elaborate on the value assessment in Section 4.4.

Based on the e^3-*value* model, business-driven requirements toward the underlying information system can be articulated. The execution of DBS runs at multiple, different sides as 'Operation control' and 'Balancing control' activities are assigned to different actors. As a consequence, the underlying information system has to maintain interactions between actors while executing DBS embedded in *value transfers* between stakeholders. The number of value transfers, which is a function of domestic scale, then translates into a system requirement that the underlying information system should handle, and can indicate concerns of system scale. In the followings, we explore the underlying information system focusing on its architecture and the interactions that occur during the execution of DBS. We assess whether the architecture design conforms the business-driven requirements, or whether modifications are needed due to concerns of system scale.

3.3. Analysis of interactions using UML sequence diagram

We drew a UML sequence diagram to describe the *characteristics* of interactions that take place between the 'Supplier' and a 'Consumer' during one-time execution of DBS. The diagram is constructed based on the analysis of the up-and-running DBS, which would be adopted to domestic setting to support imbalance reduction (see section 3.1). Figure 3 presents our model.

Interacting objects, modeled by white rectangles in the top of Figure 3, correspond to a 'Consumer' and the 'Supplier' of the study. Horizontal arrows correspond to *interactions* that occur between actors during executing DBS. Full black arrows show *synchronous* interactions implying that the sender waits for the response before continuing its operation. We use arrows pointing back to the sender to model the *operational tasks* he executes locally. The enumeration of the arrows indicates the order of interactions.

As Figure 3 shows, 'Consumer' first creates a *Demand function*, which presents his consumption profile for the next 15 minutes (arrow 1). 'Consumer' then 'sends' this demand function to 'Supplier' (arrow 2). The synchronous message passing indicates that 'Consumer' then waits for a response in order to continue his operation. 'Supplier', based on the received 'Demand function' and on the forecasted one day-ahead plan (see Section 3.1), creates a *Control bid* (arrow 3), which is used to control the actual consumption profile of 'Consumer' for the next 15-minute time frame. 'Supplier' sends this control bid to 'Consumer' (arrow 4), who then sets his actual consumption of electricity (arrow 5).

Figure 3 shows a sequence of interactions that occur between 'Consumer' and 'Supplier' while jointly executing DBS. Based on the e^3-*value* model (see Figure 2), the business setting aggregates 100,000 consumers. Thus, an arrow labels not one, but a *series* of interactions that occur between the 'Consumer' market segment and the 'Supplier'

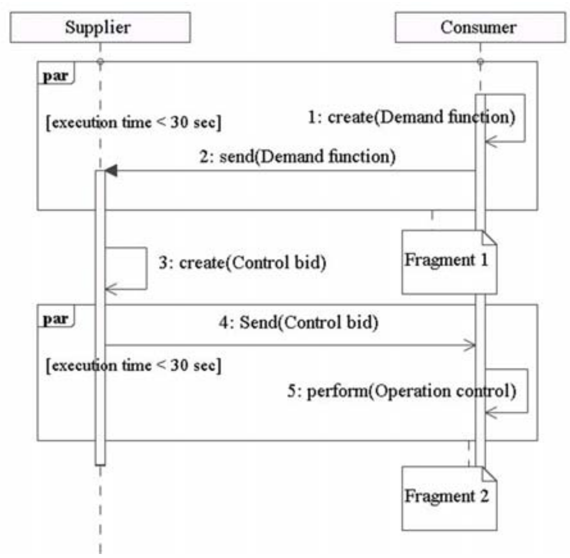

Figure 3. Interactions during the execution of the analyzed IT service

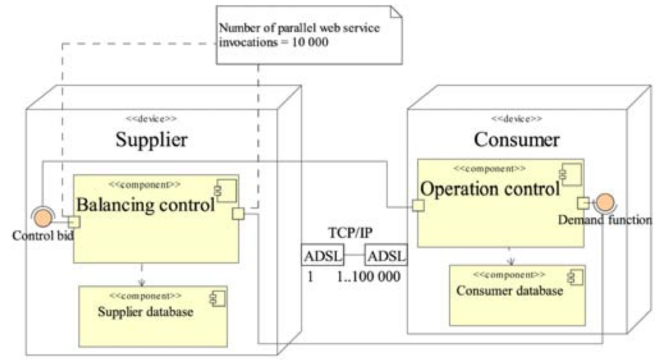

Figure 4. Information system architecture

at the *same time*. To model *how* the series of interactions occur, we use the *combined fragment* modeling construct taken from UML sequence diagrams. Time constraints are assigned to fragments in order to emphasize the time limit of interactions they aggregate. ‘Fragment 1’ shows that all 100,000 consumers create and send their ‘Demand function’ *parallel* and at the *same time* to ‘Supplier’, limited by a given execution time. Similarly, ‘Fragment 2’ tells that ‘Supplier’ sends the ‘Control bid’ *parallel* and at the *same time* to all consumers so they can perform their ‘Operational control’ operation. The nature of interactions is dictated by the requirements of the business setting. Execution of DBS, and the successful imbalance reduction requires numerous, parallel message passing and locally executed operations at the same time, within a certain time interval.

3.4. Analysis of the information system architecture using UML deployment diagram

Figure 4 presents a high-level overview of the current architecture of the information system, which needs to be adopted for domestic setting. The depicted UML deployment diagram shows *allocated soft- and hardware components*, which realize the provision of DBS. It aggregates instances of physical nodes into classes that host the same soft- and hardware structure. ‘Supplier Node’ and ’Consumer Node’ represent the execution environment of actors, who are involved in the service provision (see Figure 2).

The value activities of the e^3-*value* model appear as *components* assigned to nodes. They embed certain operational tasks as was described in Section 3.3 and in Figure 3. To calculate the ‘Demand function’, a locally stored ‘Consumer database’ is used that stores the operational data. This ‘Demand function’ can be seen as a *service* provided by ‘Consumer’. In the UML deployment diagram, it is offered via a *service port* attached to ‘Operation control’ component. Similarly, to calculate the ‘Control bid’, the ‘Balancing control’ activity relies on a locally stored ‘Supplier database’, which stores the reported day-ahead plan. The ‘Control bid’ can be seen as a *service* provided by ‘Supplier’. It is offered via a *service port* attached to ‘Balance control’ component. Component ports that offer and request the same service are connected via *assembly connections*. Figure 4 also shows that the exchange of services take place via a TCP/IP based communication path connecting the ‘Supplier Node’ with each ‘Consumer Node’ individually using ADSL routers; there is no relation between any ‘Consumer Node’. The drawn architecture highlights the centralized manner of data sharing. The cardinality of these nodes in the UML diagram matches to the cardinality of stakeholders they represent, and is derived from the settings of the e^3-*value* model. Figure 4 indicates that ‘Supplier Node’ is connected to each 100,000 ‘Consumer Node’.

As learned from the sequence diagram (see Figure 3), service ports assigned to components have to support a series of parallel interactions at the same time. The designed, centralized communication architecture however suggests that the number of parallel service invocations might cause a bottleneck of system scale. Based on real-life experiences and simulations [9], a maximum number of parallel service invocations is assumed that the service port can handle, as Figure 4 shows. This number limits the scale of this specific architecture design. If the number of invocations that occur due to the business setting is bigger than the maximum that the port can handle, there is a concern of system scale, and the revise of the architecture design is needed.

The modeling constraints of the UML deployment diagram provide sufficient handles to annotate them with financials. *Initial investments* and *fixed expenses* can be assigned to modeling constraints. By articulating the *expense carrier* of these financials, and relating these financials to modeling constructs in the e^3*-value* model, it becomes possible to *allocate* the financial effects of the deployed information system to stakeholders (see Section 4.4).

4. Reflective learning

4.1. Meta model to relate business and information system perspectives

Based on the performed model-based analysis of the study, a conceptual model is developed that bridges the employed modeling techniques in order to address *cross-perspective effects* of system scale from the business value and information system perspectives. Derived from the model-based analysis, reasoning about financial and technical effects of system scale can be structured among the lines of Figure 5. It is an *extension* to the UML 2.0 meta model [1] and to the e^3*-value* ontology [6]. In addition, the conceptual model *bridges* the UML meta model to the e^3-*value* ontology, so that the business value and information system perspectives, which they represent, are related, too.

Based on the e^3*-value* ontology, figure 5 shows that an *Actor* performs a *Value activity* in order to gain certain *Benefit* he aims to maximize. Actors might *collaborate* with other actors to support the realization of these benefits. Executing value activities can result in *Value transfers* between collaborating actors that embed various kinds of *Value objects* (e.g. services, money) that eventually realize the benefit.

Collaboration between actors can occur as a consequence of a joint execution of IT services. Value activities and transfers will then be realized by means of the underlying information technology. From the perspective of information systems, joint execution of IT services can cause several *Interactions* (e.g. web service invocations) between collaborating actors. A *Value transfer* thus *embed* one or more *Interactions*, and *Interactions* between actors can be *embedded in Value transfers*. Behavior of interactions is described via their *Occurrence specifications*. Since interactions can result from service invocations between actors, they are maintained via service *Ports* assigned to components of the information system. *Interactions* thus *relate* service *Ports* offering and requesting services, which, in return, are related via one or more *Interactions*.

Defining such conceptual relation between elements of e^3*-value* ontology helps to articulate *business-driven requirements* that translate into *system requirements*. Concluded from the case study, in order to reduce the imbalance of the 'Supplier', a certain scale in the number of participating consumers is needed. Each actor performs certain value activities to support the realization of benefits. It results in numerous value transfers embedding the 'Device flexibility' and 'Compensation fee' value objects to realize the benefit of collaboration between the 'Supplier' and a 'Consumer'. The defined *conceptual link* between *Value transfers* and *Interactions* proved to be helpful to pinpoint a possible bottleneck in system scale. The characteristics of interactions, given by their occurrence specifications, influence the *maximum number* of service invocations that the assigned service port can conform. Business requirements determine the number of value transfers, which at the same time influence the number of interactions. If the number of value transfers exceeds the maximum number of invocations, there is a concern on system scale.

The e^3*-value* model of the study (see Figure 2) shows that 100,000 value transfers occur during one-time execution of DBS. Value transfers embed several interactions depicted in Figure 3. These interactions occur *parallel*, at the *same time* and within a certain *time limit*. Based on the behavior of interactions, and on system-specific characteristics, a maximum number of service invocations is determined that a corresponding service port can handle (see Figure 4). The number of value transfers, which embed these interactions, is determined by the e^3*-value* model. Since this number exceeds the maximum, suggestions are given below to *revise* the information system architecture to conform the business requirements.

Figure 5 also provides a structured way to reason over *financial effects* of the employed information system. To provide an IT service, different *Assets* are required, which in UML terminology can be a *Device* or a software *Artifact*, which manifest different *Components* in UML. *Assets* often call for *Investments*, as type of *Expenses*. They are *assigned to* one *Expense carrier*, which corresponds to an *Actor* in the e^3*-value* model. This way, financial effects of the deployed information system can be fed into the e^3*-value* model, and a more detailed profitability analysis of stakeholders becomes possible (see Section 4.4).

4.2. Information system architecture revised based on requirements on scale

To conform the number of interactions that component ports should handle, and to resolve the technical bottleneck, suggestions were given to shift the system architecture toward a hybrid, hierarchical solution, as shown in Figure 6. As one possible solution to reduce the overload, an intermediate level of 10 *intermediary* nodes is introduced. It is positioned between the 'Supplier' and each 'Consumer'. As cardinalities show, one 'Intermediate' node *clusters* 10,000 consumer nodes. Due to their positioning, their operational role is also intermediate. They host a 'Balancing and op-

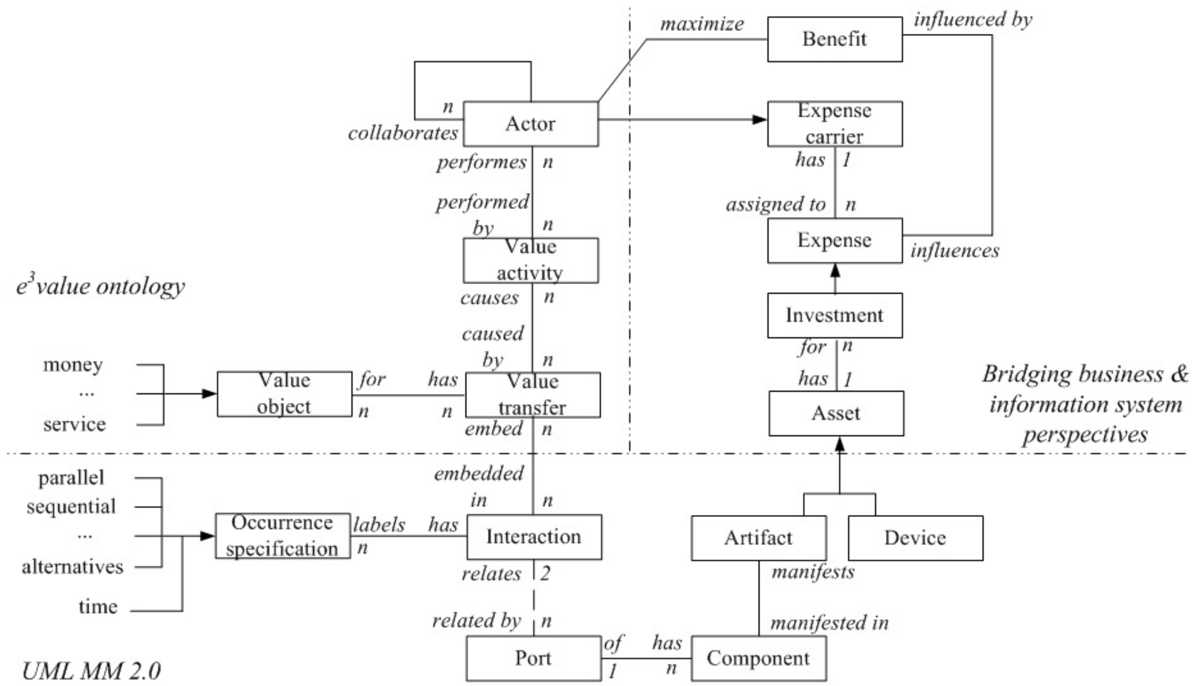

Figure 5. Meta model to understand cross-perspective effects of system scale

erational control' component. Each 'Consumer' provides his 'Demand function' to the 'Intermediary', which clusters him. Each 'Intermediate' node aggregates these functions and communicate further an aggregated 'Demand function' of the nodes they represent to the 'Supplier' on the top of the hierarchy. These interactions between 'Consumer' and 'Intermediate' nodes are handled parallel, and as a consequence, the operation of *all* 100,000 consumers is controlled at the *same time*. The role of the 'Supplier' remains unchanged. Based on the received demand functions and on the reported day-ahead plan, it provides a 'Control bid' to each 'Intermediary' node. The 'Control bid' is further communicated to each 'Consumer' to eventually control the actual operation. The introduction of these intermediaries resolve the bottleneck caused by the centralized nature of communication. The added 'Balancing and operational control' components imply additional interactions via their service ports, thus imply modifications in the UML sequence diagram. Due to space limitations, in this paper we do not elaborate on this aspect. We only state that the changed behavior of invocations does not affect the technical feasibility of the solution.

4.3. Changes in information system architecture affect business model

From the perspective of business, the added intermediary level implies a new business role within the value network. As a consequence of the information system redesign, suggestions were given to *extend* the value constellation represented in Figure 2. As a possible solution to align the changed architecture to the value network, a new actor, the 'DBS service provider' was introduced as an intermediary between the consumer market segment and the supplier.

Figure 7 shows the extended value constellation. The 'DBS service provider' executes 'Balancing and operation control' activity in order to support the 'Supplier' to reduce his imbalance. He controls the operation of a 'Consumer', and pays 'Compensation fee' for the provided 'Device flexibility'. Based on the performed control, he offers the 'Balancing service' for the 'Supplier' and receives 'Fee' in return. 'DBS service provider' maintains and operates all the components of the information system that enables this intermediary role.

4.4. Assessing financial sustainability of the constellation

Based on the characteristics of the revised business model (see Figure 7), and on the predicted level of imbalance reduction derived from simulated performance measures, we performed calculations concerning the financial feasibility of DBS as a potential commercial service for domestic electricity suppliers. After assigning values to all relevant financial variables, a profitability sheet was generated for a *one-time* execution of DBS, so for a 15-minute

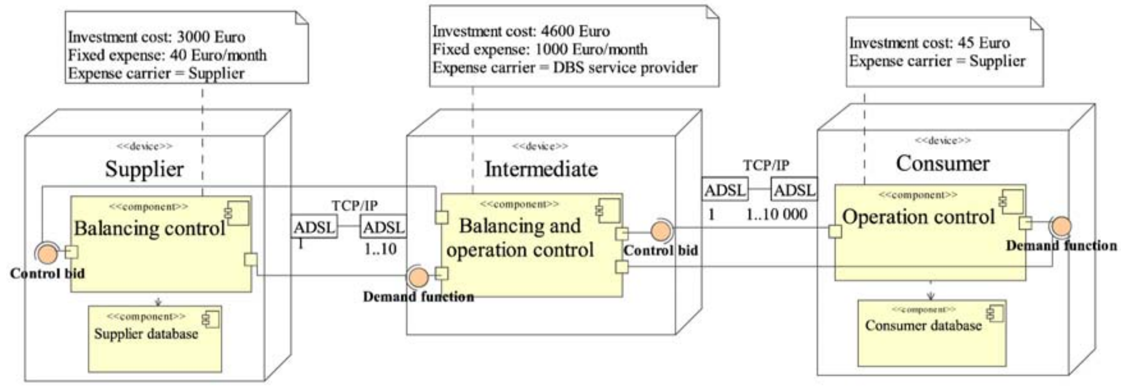

Figure 6. Modified information system architecture to accommodate requirements on scale

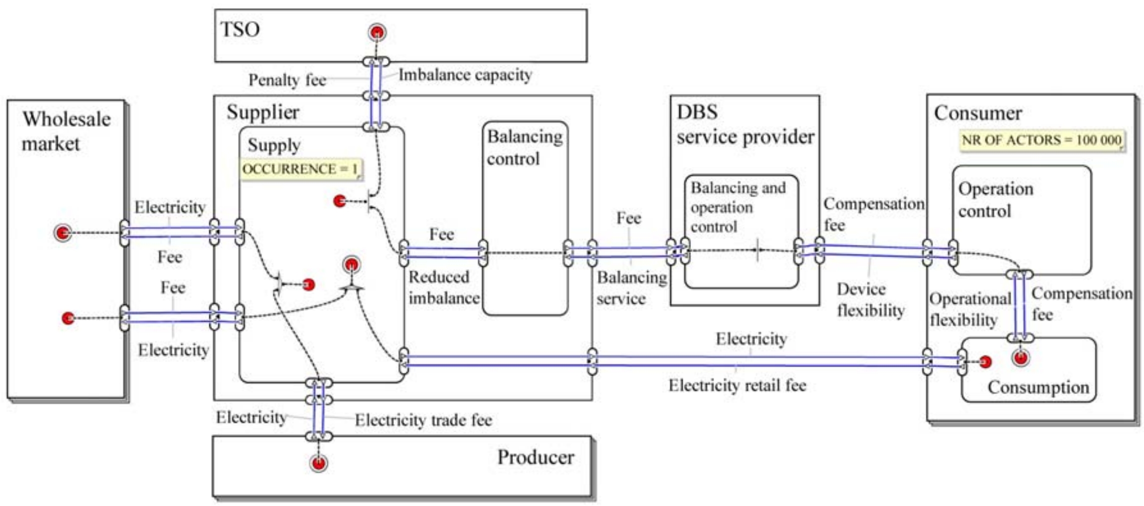

Figure 7. Modified business model to conform the changed structure of system architecture

time horizon. The revised information system architecture helped to articulate financial effects (investments and fixed expenses), which were allocated to actors in the e^3-*value* model (see Figure 6). To assess the attractiveness of a given business scenario, a series of similar profitability calculations were performed using e^3-*timeseries* technique [4]. Due to the domestic scale, the initial investments are high; the return over investment is expected in 2 years time.

The result of the described calculation is exemplified in Table 1 for a 'Supplier' who executes 'Balancing control' activity. The table usually lists all considered time frames, for brevity reasons, we only show the first period. The cash in- and outflows are listed based on value transfers related to executing 'Balancing control', as shown in the revised e^3-*value* model. Period 0 shows investments that a 'Supplier' has to pay for, based on the analysis of the underlying architecture. After calculating the net cash flow for all time frames, they are summed up using the Discounted Net Present Cash Flow method [2], reflecting on a two-year operation of DBS, including estimates on investments, accounting properly for the time value of money, cost of capital, and risk associated with participating in the constellation. The chosen method suits our light-weight exploratory approach. The initial calculations remain at a low level of complexity, and offer good insights on the financial attractiveness of the employment of DBS. Due to space considerations, we here only mention that similar, positive conclusions were drawn for other stakeholders.

5. Conclusions and future work

In this paper we introduced a *model-based* analysis combining two different (e^3-*value* and UML) modeling techniques that assisted to explore *technical* and *financial* effects of scale. Our approach proved to be helpful to support the design of networked constellations providing commer-

Table 1. Net value flow sheet for 'Supplier'

Actor:	Supplier - 'Balancing control'					
Timeframe:	period 0					
						Total
INVEST-MENT						4 503 000
Timeframe:	period 1					
Value Interface	**Value Port**	**Value Transfer**	**Occur-rences**	**Valua-tion**	**Eco-nomic Value**	**Total**
DBS,MONEY			X		X	
	in: DBS		X	X	X	
	out: MONEY (Fee)	MONEY	1	-20	-20	
Reduced imbal-ance,MONEY			1		-X	
	in: MONEY (Fee)	MONEY	1	90	90	
	out: Reduced imbalance		1			
EXPENSES	X	MONEY			-0.014	
NCF:						X
Timeframe:	70 080 + 1					
			...			
DNCF:						67 900

cial IT services. From the perspective of business, scalability of IT services is often driven by revenue-driven concerns, i.e. to accommodate the desired profit. As demonstrated, such business requirements translate into system requirements, which naturally should be accommodated by the underlying information system. The bottleneck of system scale is however often caused by the mismatch between business-driven requirements and technology-given capabilities, which negatively influences the technical and financial feasibility of the desired business configuration.

To address the technical and financial potentials of the desired value constellation it is therefore important to understand the business-driven scalability requirements already during the design phase. An explorative feasibility assessment during the design of constellations should be done in a light-weight fashion, should avoid complexity, and should provide further direction for a more detailed and focused requirements engineering and system design track. In line with this philosophy, we propose in this paper a model-based technique and combined different modeling techniques (e^3-*value* and UML) to support the design of feasible constellations during the early explorative phase. The aim of our approach is to articulate possible shortcomings that influence the technical and financial potentials of IT-intensive business ideas.

As the case study showed, the employed modeling techniques combined with simulations and real-life experiences help to analyze and pinpoint concerns of system scale and, as a consequence, assist system designers to align IT with business requirements already during the early design phase. Furthermore, the introduced method supports the assessment of *cross-perspective* effects of system scale, which in turn helps to detect how concerns of scale interrelate and affect the financial and technical feasibility of the system under design. As future research, we aim to further refine our cross-perspective scalability assessment, with the emphasis on business-driven requirements toward system scale. We seek to understand the cause of scale from multiple viewpoints as well, emphasizing the influencing role of network effects on information system requirements.

References

[1] Unified Modeling Language: Superstructure. *http://www.omg.org/docs/formal/05-07-04.pdf*, accessed May 2007.

[2] R. Brealey, S. Myers, and F. Allen. *Corporate Finance*. McGraw Hill Higher Education, 2005.

[3] R. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.

[4] Zs. Derzsi, J. Gordijn, K. Kok, H. Akkermans, and Y. Tan. Assessing feasibility of IT-enabled networked value constellations: A case study in the electricity sector. In *Proceedings of the CAiSE*, volume 4495:66-80 of LNCS, 2007.

[5] J. Gordijn and H. Akkermans. Business models for distributed energy resources in a liberalized market environment. *The Electric Power Systems Research Journal*, 2007.

[6] J. Gordijn and J. Akkermans. Value-based requirements engineering: Exploring innovative e-Commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[7] Christian Grönroos. *Service management and Marketing: A Customer Relationship Management Approach*. John Wiley & Sons, Chichester, UK., 2000.

[8] P. Jogalekar and M. Woodside. Evaluating the scalability of distributed systems. *IEEE Trans. Parallel Distributed Systems*, 11(6):589–603, 2000.

[9] K. Kok, C. Warmer, and R. Kamphuis. The PowerMatcher: Multiagent control of electricity demand and supply. *IEEE Intelligent Systems*, 21(2):89–90, March/April 2006.

[10] J. Magretta. The power of virtual integration: An interview with Dell Computer's Michael Dell. *Harvard Business Review*, 76(2):72–84, March-April 1998.

[11] D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Nicholas Brealy Publishing, London, UK, 2000.