Towards a Multi-perspective Assessment of Scalability of Distributed IT Services

Zsófia Derzsi and Jaap Gordijn

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

Abstract. Scalability is an important issue in distributed IT service design and should be addressed at least from *technical* and *economic* perspectives. We present a conceptual framework that addresses scalability from these two perspectives. To explore scalability of distributed services, our framework employs a value model and UML deployment model to describe scalability concerns of a distributed, commercial IT service. We illustrate our approach by a case study.

1 Introduction

We consider commercial and distributed IT services built on the top of Internet technology as *commercial* deeds of a mostly intangible nature [4]. They are operated by *networked* constellations of enterprises (suppliers and costumers), who, using each other core competencies, jointly work on the satisfaction of an IT-intensive consumer need. The underlying information system architecture, which puts the constellation into operation, shows a distributed landscape as software and hardware components to realize such a service are typically distributed among a number of these enterprises.

The term 'distributed IT service' requires two different perspectives. From the *perspective of business* it is a *commercial* concept, which is offered by a network of enterprises rather than one enterprise to satisfy an IT-intensive consumer need. A well-known example is the need to surf on the web, which can be satisfied by an Internet Service Provider (ISP). From the *perspective of information technology* it means that the required software and hardware components of the information system are located at multiple, different places, which are interconnected e.g by web service technology leveraging the Internet [7]. To avoid confusion, in the rest of this paper we use the term 'service' to label the commercial perspective, while 'web services' refer to the information technology perspective.

Ideally, distributed services should remain both technically and economically feasible in different business settings. Many configurations are possible, caused by e.g. varying the participating enterprises or by the increase of number of consumers. One desirable requirement for the underlying distributed system is that it must be *scalable*, meaning that it should handle different business settings while at the same time provide a constant output in performance [7, 5]. Additionally, business and information technology perspective on a distributed system should match; a business setting should match with its supporting distributed information system with respect to scale.

Assessing the scalability of distributed services is a complex task. Many scientific papers addressed the problem and proposed different analytical and performance measures, mostly from the information system point of view [7, 1]. System scale should be matched by sufficient capacity in soft- and hardware components resulting in financial consequences (e.g. investments) for its stakeholders. Moreover, assumptions for designing a scalable system (e.g. the expected number of customers) are important; a system that should support 5 concurrent customers looks often quite different from a system that must support millions of concurrent customers. Several examples of system scalability research focus on the importance of cost effects [8, 5] in addition to performance. In a commercial, networked business setting, however, the *allocation* of these costs among enterprises is of an importance, too, because such an allocation directly influences potential profitability of an enterprise involved.

In this paper, we propose a conceptual framework to relate scalability concern from a business value perspective (using e^3*-value* models) and information technology perspective (using UML deployment diagrams). Additionally, we discuss how to address scalability analysis from a business *and* technical perspective. We illustrate our framework by a small case study.

2 Perspectives on Evaluating Scalability of Distributed IT Services

Our framework is motivated by the work of Neumann [7], where scalability of distributed information systems is addressed from three aspects: *(a) size*, the increase or decrease of objects or users of the system, *(b) allocation of computational resources*: hard- and software components, as the execution of subtasks can happen by different stakeholders at geographically different locations, and *(c) administrative control*, as systems leveraging the Internet technology can cross multiple, independent administrative domains and conflicting policies can occur with respect to e.g. resource usage or security.

Ideally, the performance of distributed information systems should remain constant as any change occur along these three aspects (e.g. increase of users). Analyzing the effects of these changes purely from the information technology perspective would only judge whether the provision of the distributed service it supports is technically feasible. However, scalability has its price, and thus its financial consequences. It is of an importance to examine the resulting *economic value* effects in order to assess whether the provision of the distributed service remains financially sustainable.

Figure 1 summarizes our framework built on the above articulated concepts and shows their relations. Technical scalability is ultimately about *system performance* and addresses whether it remains constant if there are changes in the three forementioned scalability aspects (users/objects, resource allocation, administrative control). Economic scalability is about *financial performance* of the enterprises of networked constellation, and evaluates the resulting financial effects of these changes (e.g. additional investments). The scale as supported by the distributed information system (technical perspective) should correspond to its financial effects as indicated by the business perspective.

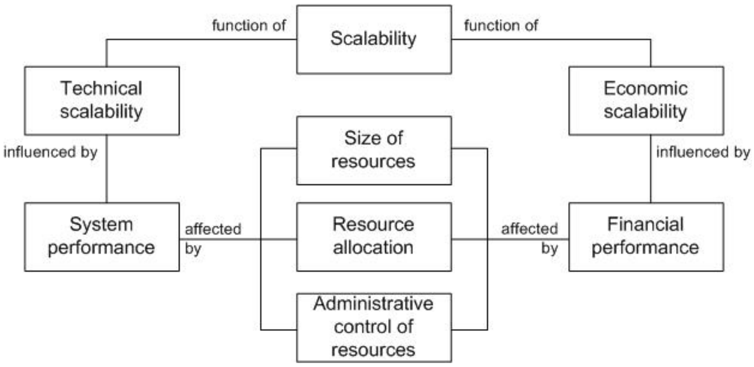

Fig. 1. Conceptual framework to assess scalability of commercial, distributed IT services supported by distributed information systems

To guide scalability analysis, our research employs modeling techniques to describe the information system and the business setting focusing on size, resource allocation, and administrative control. In addition, our research investigates how constructs of these modeling techniques can be related to support reasoning over scalability. As a first step, we employ the e^3*-value* technique (for a detailed description please consult [3]) to describe the business model and we use the UML deployment modeling technique to describe the information system perspective. We explore to what extent these modeling tools are applicable to guide the *technical scalability* assessment, and we also show how constructs of a value model and a deployment model can be related to support our analysis. We do not elaborate on economic scalability.

3 Case Study: A Commercial Distributed Service to Reduce Imbalance in Electricity Supply

We now introduce a case-study on electricity supply and consumption (see [2] for details). 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 imbalance fee for the parties, who caused the imbalance.

The analyzed distributed service (Distributed Balancing Service (DBS)) is used to perform near-real time, distributed control over the electricity supply and consumption of commercial portfolios (consisting of a series of electricity generators and consumers) [6] in order to reduce imbalance. In case of imbalance, consumers and/or producers of the commercial portfolio are asked to change their level of production and/or consumption. Obviously, such near real-time control is only possible using advanced, distributed information technology. All stakeholders of the portfolio (producers, consumers and supplier) have to employ certain software and hardware for execution of the DBS at their production and consumption sites.

Portfolios may vary in size (i.e. the number of consumers) and in geographic location (i.e. due to the liberalized market in the electricity domain [6]). In most cases, portfolios aggregate stakeholders with a different operational profile (i.e. wind turbines, generators), thus the administrative complexity while executing DBS can easily increase. It is thus important to assess the scalability of DBS to assure its deployment in different portfolio settings. In the following we use this case to exemplify our model-driven analysis, focusing, due to space limitations, on technical scalability.

4 Analysis of Technical Scalability: a Case Study

4.1 Characteristics of the Business Design Affecting Technical Scalability

We have constructed an e^3*-value* business model (we assume that the reader is familiar with e^3*-value* , otherwise please consult [3]), figure 2 shows a simplified extraction of it as presented in [2]. We now assess how the e^3*-value* method supports the analysis of technical scalability (see Figure 1).

The model represents the *one-time execution* of the functionality offered by DBS, namely keeping the balance of supply and consumption. In the e^3*-value* methodology such an execution is shown by a *dependency path*, which models how a consumer need is satisfied by performing 'value activities' and 'value transfers' by different entities. In our example, the path connects the business parties of the networked constellation (represented by 'actor' and 'market segment' modeling constructs) who jointly execute the distributed service satisfying the occurring business need: the imbalance reduction. 'Value activities' demonstrate *who* executes which activity with respect to the distributed service. 'Value transfers' encapsulate exchanged 'value objects' resulting from performed 'value activities'.

The dependency path of Figure 2 thus depicts that a supplier executes a 'Balancing control' activity to decrease imbalance of 'Supply'. It operates together with the 'Operation control' activity maintained by consumers, which controls their 'Consumption'. Consumers offer their 'Device flexibility' as a result of their 'Operation control' and receive 'Compensation fee' in return.

The path first provides information concerning the *size* of actors of the networked constellation executing the distributed service. An 'Actor' models by definition one (business) entity. Cardinality of the market segment is equal to

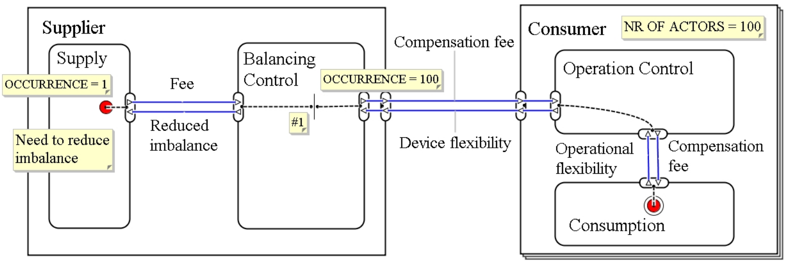

Fig. 2. Structure of the business design, represented by e^3-*value* modeling technique

the number of actors it aggregates. By summing up the number of actors along the path, the number of actors of the constellation can be found for the need at hand (in this example: 100 consumers + 1 supplier = 101). This number of actors is a first indication of scale to be supported by the information system.

The dependency path also helps to determine the number of value transfers between actors. This is given by the number of occurrences of the need (in this example, 1, since the dependency path demonstrates one-time execution), and influenced by the number of actors. In this example, the number of value transfers is sized up according to the number of consumers involved in the service execution, expressed by the explosion element (#1).

In terms of our earlier mentioned scalability aspects, the value model helps to articulate the following characteristics of the business setting, which influence technical scalability:

- *Size:* the value model shows the number of business actors of the networked constellation and the number of value transfers.
- *Resource allocation:* the value model demonstrates the distributed nature of the service by allocating value activities to different entities, but does not provide any specific information over the employed hard- and software resources, nor how these activities are executed. In addition, it does not show how the business actors of the constellation are geographically distributed.
- *Administrative control:* the value model only shows what value activity is executed and what value object is exchanged but gives no operational insight.

To assess technical scalability of the DBS in more detail, it is necessary to explore the business processes of actors within the networked constellation.

4.2 Characteristics of Information System Design Affecting Technical Scalability

Below, we present a UML deployment diagram to describe the structure of the distributed information system supporting the provision of the DBS. The diagram is constructed based on interviews with domain experts. Figure 3 shows an extraction of it (a more detailed model and explanation can be found in [2]). We

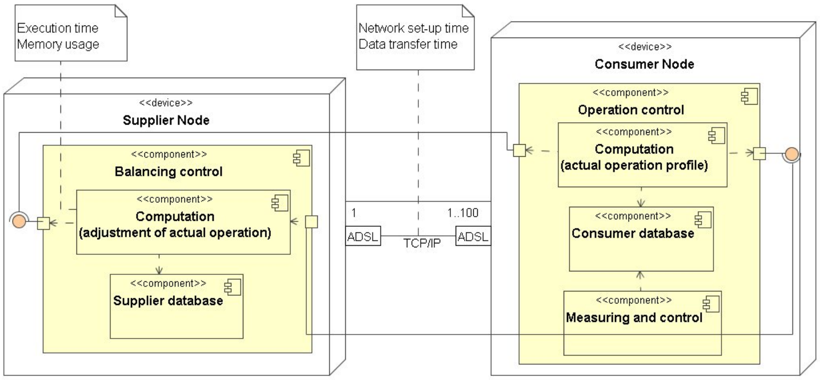

Fig. 3. Structure of the system design, represented by UML deployment class diagram

now assess how the UML deployment diagram supports the analysis of technical scalability (see Figure 1).

The depicted deployment diagram gives a better structural insight by showing the *allocated soft- and hardware resources* and *web service ports* offering and requiring web services. The class diagram aggregates instances of physical nodes into classes that host the same software and hardware structure, yet independently from geographic location. The cardinality of these classes thus provides information about the *size* of employed hardware and software resources of the information system.

The modeled ports attached to components help to understand the invocation of web services of the distributed information system. These web services are invoked via a TCP/IP based communication path connecting consumer and supplier nodes. Each node communicates via ADSL router.

The diagram also represents the idea of centralized communication in this specific case. ‘Consumer nodes’ do not exchange data with each other, they only communicate with the ‘Supplier node’, highlighting the centralized manner of data sharing. It is the ‘Balancing control’ component, which possesses all the information needed to adjust actual operation profiles of consumers and thus to perform distributed control. ‘Operational control’ components are responsible only for the local device control based on the adjustments provided by ‘Balancing control’. Such a centralized organization of communication and web service exchanges suggests that the ’Balancing control’ component of ‘Supplier node’ may form a potential performance bottleneck that can limit the technical scalability of the distributed service [9]. We assume that local device control is performed satisfactory.

The structure of the information system thus suggests that the communication network and the computation task of the ‘Balancing control’ influence the system performance, thus the technical scalability. *Functional parameters* are attached to corresponding modeling constructs (see attached comments in Figure 3). We employ the following metrics: *(a) network set-up time*, the time needed to initialize and to build up automatically the communication network between supplier and consumer nodes, respectively, *(b) data transfer time*, the time needed to transfer data from one point to another, as a function of the type of the end connection (i.e. ADSL), *(c) execution time*, the time needed to perform the required tasks, *(d) memory usage* for computation and for maintenance of communication channels.

We assume that these metrics together determine the *maximum number* of web service invocations that the 'Supplier node' can handle. To determine the value and significance of these performance metrics, however, the analysis of operational processes and of the behavior of web service invocations (e.g. regularity) is essential. To this end, the static modeling approach, as the UML deployment diagram suggests, is not sufficient. In addition, 'network set-up' and 'data transfer time' is location dependent, yet the deployment diagram does not provide insight over the geographic allocation of nodes. Moreover, the analysis of operational processes is essential to assess how the administrative control aspect of scalability would influence system performance, since devices might have diverse operational profiles, yet their operation has to be controlled equally.

In terms of our earlier mentioned scalability aspects, the static, structural constructs of the UML deployment class model help to articulate the following characteristics of the information system, which influence technical scalability:

- *Size:* the cardinality of UML constructs shows the size of employed hard- and software components.
- *Resource allocation:* the deployment class diagram gives better structural insight, yet independently from geographic distribution. Web service ports display the structure of web service exchanges.
- *Administrative control:* the deployment diagram shows the components of the distributed service, but gives no insight to the behavior of these components.

Further refined analysis of the operation of components and of web service invocations is needed to get better insight to technical scalability.

4.3 Relating e^3-*value* and UML deployment diagrams

In the following we show that the e^3-*value* technique and UML deployment class diagram, if correctly related, may contribute to the technical scalability assessment. *Value activities* of the value model that are required for the one-time execution of DBS appear as *components* in the UML deployment diagram. Value activities result in *value transfers* between actors exchanging value objects. This is maintained by offered and received web services between *web service ports* of these components on the information system level.

As a consequence, *value transfers* of the value model encapsulate *web service invocations* needed to execute the distributed service (i.e. the offered 'Device flexibility' is supported by these invocations). The e^3-*value* technique is capable to determine the number of value transfers occurring between actors, which can be used to estimate the number of web service invocations between software components that the underlying information system should handle. As the maximum number of web service invocations that can be handled during the one-time execution of DBS is known for the information system at hand, the number of value transfers may indicate whether technical scalability is violated.

5 Conclusions and future work

In this paper, we proposed a conceptual framework to support the technical scalability assessment of commercial, distributed IT services offered by networked constellations of enterprises. The framework addresses the evaluation of scalability among two - *technical* and *economic* - perspectives taking different aspects (size, resource allocation, administrative control) of scale into account.

This paper focuses on technical scalability. To guide the analysis we employed and related e^3*-value* technique and UML deployment class diagram, and we explored how our model-driven approach can support our aim. As a next step, we extend our analysis by employing behavioral modeling techniques for the scalability assessment.

Another line of research focuses on aspects of economic scalability. Coupling the e^3*-value* and UML techniques seems as a suitable candidate to support our assessment [2], however, further expansion of analysis toward business processes is needed in order to gain better insight on financial consequences of scale.

References

1. Gunnar Brataas and Peter Hughes. Exploring architectural scalability. In *WOSP '04: Proceedings of the 4th international workshop on Software and performance*, pages 125–129, New York, NY, USA, 2004. ACM Press.
2. Zsófia Derzsi, Jaap Gordijn, Koen Kok, Hans Akkermans, and Yao-Hua Tan. Assessing feasibility of it-enabled networked value constellations: A case study in the electricity sector. Conditionally accepted by CAiSE 2007, 2007.
3. J. Gordijn and J.M. Akkermans. Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.
4. Christian Grönroos. *Service management and Marketing: A Customer Relationship Management Approach.* John Wiley & Sons, Chichester, UK., 2000.
5. Prasad Jogalekar and Murray Woodside. Evaluating the scalability of distributed systems. *IEEE Trans. Parallel Distributed Systems*, 11(6):589–603, 2000.
6. Koen Kok, Cor Warmer, and René Kamphuis. The PowerMatcher: Multiagent control of electricity demand and supply. *IEEE Intelligent Systems*, 21(2):89–90, March/April 2006.
7. B.C. Neuman. Scale in distributed systems. In T. Casavant and M. Singhal, editors, *Readings in Distributed Computing Systems*, pages 463–489. 1994.
8. Maarten Van Steen, Stefan Van der Zijden, and Henk J. Sips. Software engineering for scalable distributed applications. In *COMPSAC '98: Proceedings of the 22nd International Computer Software and Applications Conference*, pages 285–293, Washington, DC, USA, 1998. IEEE Computer Society.
9. Maarten van Steen and Gerco Ballintijn. Achieving scalability in hierarchical location services. Technical Report, IR-491, Vrije Universiteit, Department of Mathematics and Computer Science, 2001.