

# A Model of View-Based Enterprise Business Service Quality Evaluation Framework

Audrone LUPEIKIENE\*, Jolanta MILIAUSKAITE,  
Albertas CAPLINSKAS

*Institute of Mathematics and Informatics, Akademijos 4*

*LT-08663 Vilnius, Lithuania*

*e-mail: audrone.lupeikiene@mii.vu.lt, jolanta.miliauskaite@mii.vu.lt,*

*albertas.caplinskas@mii.vu.lt*

Received: April 2013; accepted: December 2013

**Abstract.** The symbiosis between an enterprise architecture and service-oriented architecture results in so-called service-oriented enterprise architecture and brings up new problems for service-oriented enterprise systems engineering. One of the most important is a business service quality definition, specification and evaluation. The paper proposes a formal model of enterprise business service quality evaluation framework to encompass and balance all the viewpoints and perspectives on an enterprise business service quality.

**Key words:** service-oriented enterprise systems, quality of service, view-based modelling, view balancing.

## 1. Introduction

The symbiosis between an Enterprise Architecture (EA) and Service-Oriented Architecture (SOA) results in so-called Service-Oriented Enterprise Architecture (SoEA) and brings up new problems for Service-Oriented Enterprise Systems (SoES) engineering. A number of different stakeholders have different, often conflicting understandings of a SoES service quality and how it should be defined, specified, and evaluated. Thus, despite a large number of QoS models and ontologies, the question “What does QoS mean?” still has no final answer, at least, in the context of SoES. The open issue also remains a quality evaluation of the overall SoES.

The term QoS was introduced in telecommunication where it was focused on the service performance measures from the network perspective. Later it was extended including even hardly related to quality characteristics such as a service requestor’s satisfaction or service cost.<sup>1</sup> Currently the term QoS refers to several different things. As stated in Benbernou *et al.* (2010), “*This set of quality attributes does not characterize only the service*

---

\*Corresponding author.

<sup>1</sup>Note that in our paper service requestor and service consumer are different entities. The first one is a person or an organisation while the second one is a piece of software.

but any entity used in the path between the service and its client. Such an entity may exist in any of the three possible service levels. Thus, different QoS attributes may be used to define the QoS of a service in the application, service, and infrastructure levels". There the term client refers to the service requestor.

In the context of SoES, the quality of an enterprise business service (EBS) is typically addressed by the term QoS for web services (WS QoS) that, unfortunately, causes some confusion. One of the most popular WS QoS definitions is presented in Kritikos and Plexousakis (2012), where this term is defined as "a set of nonfunctional attributes of the entities used in the path from the WS to the client that bear on the WS's ability to satisfy stated or implied needs in an end-to-end fashion". The definition speaks about EBS end-to-end quality but calls it web service quality. However, in many contexts it is important to distinguish between these qualities, i.e. to differentiate between the extent to which an EBS satisfies stated or implied service requestor's needs and the manner in which this EBS is delivered. Using the terms proposed by Grönroos (1982), it is important to differ between EBS *technical quality* and EBS *functional quality*. As pointed out in Parasuraman et al. (1985), a variety of factors, including resource constraints, market conditions, and incompetence of available developers, may result in a discrepancy between the real needs of end user and the actual Service Level Agreement. Consequently, the gap often appears between the service requestor's needs and service owner's treatment of those needs. From service requestor point of view, often it is important not only to satisfy his or her needs but also to take into account some additional factors. Therefore, it is confusing to use the term WS QoS to address an end-to-end quality experienced by an EBS requestor. It also is unclear whose viewpoint (service requestor's, service owner's, etc.) it represents.

Besides, the definition presented in Kritikos and Plexousakis (2012) and similar ones emphasise only one aspect of the concept, namely, non-functional attributes. Thus, there exists a need to develop a framework that encompasses and balances all viewpoints and perspectives on EBS quality at a higher abstraction level.

This paper aims to cope with the conflicts in the enterprise business service quality evaluation given by different stakeholders. It proposes a model of view-based framework to describe, relate to each other, to balance the different viewpoints and perspectives on QoS, and to evaluate the QoS in web-based SoES.

The rest of the paper is organised as follows. Section 2 discusses the related works. Section 3 describes the research methodology. Section 4 discusses the differences between SOA and SoES. Section 5 presents the formal model of view-based enterprise business service quality evaluation framework. Finally, Section 6 concludes the paper.

## 2. Related Works

A lot of QoS models and ontologies have been proposed. There are also many good and exhaustive surveys on the QoS models (Benbernou et al., 2010, among others), QoS specification and evaluation. However, there are only a few works (Wang et al., 2004; Farroha and Farroha, 2007; Wu and Li, 2010; Usmani et al., 2011) surveying QoS models

devoted to SoES. Besides, these surveys focus on some specific aspects of the problem, and to the best of our knowledge no one investigates this problem systematically.

Most of the proposed QoS models can be classified into taxonomy, activity-based, and ontology-based ones.

Taxonomy-based quality models structure a quality along the characteristics (e.g. security, interoperability, reliability, usability, efficiency, maintainability and portability). In other words, these models are more or less exhaustive taxonomies of the QoS characteristics. A typical example of taxonomy-based models is the S-Cube Reference Model (SCRM) (Benbernou *et al.*, 2010). Another example is the SQuaRE based Web Services Quality Model (SQuaRE-WSQM) (Abramowicz *et al.*, 2008). It follows the similar classification scheme established by ISO/IEC SQuaRE model. OASIS Quality Model for Web Services (WSQM 2.0) (Kim and Lee, 2005) also defines a hierarchical taxonomy of quality attributes, referred to as quality factors. However, as pointed out in Döweling *et al.* (2012), the classification approach used in taxonomy-based models often lacks clear semantics regarding the relationship between super- and sub-characteristics.

To eliminate this shortcoming the activity-based quality models have been proposed. Originally the activity-based quality models were introduced to model software maintainability (Deissenboeck *et al.*, 2007). Currently this approach is successfully applied to model SOA systems quality. In activity-based QoS models the quality is described along the activities performed on or with a SOA system. However, the proposed activities, roles and view dependencies are only mentioned, but not described in detail. The model is too much technology as well as product oriented and encompasses only three viewpoints. Nevertheless, our research to some extent is inspired by the philosophy beyond this model.

There are also ontology-based QoS modelling approaches, for example (Bianchini *et al.*, 2004; Papaioannou *et al.*, 2006). In these approaches, QoS models – usually taxonomy-based ones – are accompanied by QoS ontologies, which define QoS characteristics in a formal way. Although there are no obstacles to use appropriate QoS ontology to define QoS characteristics and even viewpoints and perspectives in the conceptual view-based framework proposed in our paper, the ontology-based QoS modelling approach is not directly related to our research.

Close related to our research are works on view reconciliation methodology, which in the field of Computer Science was originated by SADT methodology (Ross, 1977) and Leite (1988) PhD thesis, and was further developed mainly in software requirements engineering.

Close related to our research also are works on application of the  $i^*$  framework (Yu, 1995) for the enterprise (Rolland and Prakash, 2000; Stirna and Persson, 2007) and software requirements (Kavakli and Loucopoulos, 2004; van Lamsweerde, 2001) modelling and reasoning. We adopted this framework to model interdependencies between viewpoints and between perspectives, and to reason about alternative configurations of QoS characteristics' values in order to compromise requirements stated on the basis of different viewpoints. For this aim, systematic view-based QoS models analysis procedures are required. A number of analysis procedures for analyzing goal models have been introduced in the  $i^*$  framework related literature (Giorgini *et al.*, 2004; Amyot *et al.*, 2010;

Franch, 2006; Letier and Lamsweerde, 2004). Majority of these procedures emphasize automated reasoning over goal models ignoring the interactive nature of such analysis. As pointed out in Horkoff and Yu (2009), “*The full automation in these procedures does not give the evaluator freedom to make decisions in the presence of conflicting, partial or unknown information*”.

Pioneering works on the design of automated reasoning procedures were published by Giorgini and co-authors (Giorgini *et al.*, 2004; Sebastiani *et al.*, 2004). Authors proposed a number of qualitative and quantitative procedures for goal model analysis which separately propagate negative and positive evidence, are fully automated, and work in a forwards and backwards direction. The proposed algorithms are sound and complete. The algorithms take as input labels for some of the lower goals of the model and infer other labels higher up. In other words, given a formal axiomatic goal model and labels for some of the goals, the algorithms propagate these labels forward, towards root goals. If the graph contains loops, this is done until a fix point is reached. An axiomatisation of goal models also was proposed by Giorgini and co-authors (Giorgini *et al.*, 2003, 2002). These works have strong impact on our approach for formalisation of QoS-oriented goal models.

An interactive qualitative approach allows one to narrow the number of alternatives and further to test the feasibility of remained alternatives using some automated quantitative procedures. Stirna and Persson (2007) developed one of the first procedures of this type. It was developed as a part of the NFR Framework (Chung *et al.*, 1995) and based on the notion of goal “satisficing”. The procedure pretended to be extensible for  $i^*$  framework, however, it emerged that its interactivity level is too restrictive to be effectively applied to  $i^*$  models. Horkoff and Yu (2010, 2009) developed other interactive qualitative procedure for goal- and agent-oriented models. This procedure is applicable to  $i^*$  models and allows evaluator to compare alternatives in the domain by asking “what if?” type questions. It could also be applied to the NFR Framework (Amyot *et al.*, 2010) and GRL (Liu and Yu, 2004) because they both are syntactic subsets of  $i^*$  framework. The process starts by assigning the initial values to labels expressing the degree of satisfaction or denial to intentions related to the analysis question. Using the preliminary defined rules, these values are propagated through the model links. Human judgements is required in cases when it is necessary to combine multiple conflicting or partial values. Evaluator analyzes the final values taking into account the original “what if?” question.

### 3. Research Methodology

The research was done in two steps. First of all, using the methodology proposed by Webster and Watson (2002), a concept-centric literature survey on the QoS in SOA with the special emphasis on SoES was carried out. In second step, the results of the survey were generalised and juxtaposed against the results of empirical analysis of observations made by the authors working in a research project for the development an innovative Advanced Planning and Optimisation system for make-to-order production environments.

#### 4. SOA and SoEA

The most important innovation of service orientation is the manner in which the separation of concerns is done. SOA is an architectural style that implements this approach. Various authors define SOA differently. We adopt the following definition proposed by Bieberstein *et al.* (2005):

*“A service-oriented architecture is a framework for integrating business processes and supporting IT infrastructure as secure, standardized components – services – that can be reused and combined to address changing business priorities”.*

SoEA is a substyle of SOA. Therefore, SoEA introduces two new high-level abstractions, namely, enterprise business services (EBS) and enterprise business processes (EBP). Enterprise business services are the abstractions of existing application capabilities, which are aligned with the enterprise business functions. Enterprise business processes are the abstractions of the overall business functioning.

The conceptual, technological and technical differences between SOA and SoEA are presented in Table 1.

Table 1  
SOA vs SoEA.

SOA	SoEA
Internet-wide open system. Developed in a bottom-up manner	Relatively closed enterprise-wide system controlled on an enterprise-wide level. Developed in a top-down manner. Enterprise service inventory
Any business services. No ability to define global data types and normalize <sup>a</sup> business services	Normalized enterprise business services aligned with the enterprise business functions, the use of global data types.
Not purported to support a particular business strategy and to implement predefined business processes.	Business-driven, i.e., support enterprise’s business strategy and objectives. Enterprise business process coordinates a set of interacting EBSs
No guide on the set of services, on how they are built and deployed. No control over changes in services	EBSs are designed, developed and deployed in compliance with the enterprise-wide standards. All changes are under control
The structure of messages is standardized (e.g. by SOAP) but not unified. ESB interfaces are standardized (by WSDL), but not clearly defined, not stable. No ability to use global data types in the interfaces	The structure of messages is unified. EBS interfaces are clearly defined, stable, and make use of global data types
SLA is negotiated between provider and consumer at the run time	Mandated (mostly) at the enterprise-wide system at the design time
Direct peer-to-peer communication between consumer and provider. UDDI for service registration and discovery	Enterprise service bus as a mediator between consumers and providers
Neither service providers nor consumers can control the SOA infrastructure and communication networks	Intranet, extranet, and the whole infrastructure, including enterprise service bus, servers and other elements, are under control by the enterprise
Recommended security and safety standards	Mandatory security and safety standards
Some services are situation-aware but only in rare cases are context-aware because the context as a rule is ill-defined	All services are context-aware because they run in the well-defined enterprise context

<sup>a</sup> Normalisation means that each EBS should be designed with the intent to avoid similar or duplicate bodies of service logic.

## 5. EBS Quality Model: A View-Based Approach

### 5.1. Technical Preliminaries

In the first order predicate logic formalism, the quantifiers  $\exists$  and  $\forall$  are “unrestricted” in the sense that  $(\exists x) P(x)$  means that there is some entity in the universe of discourse  $U$  which has the property  $P$ .  $(\forall x) P(x)$  means that all entities in the universe of discourse  $U$  have the property  $P$ . In this formalism the sentence “Each element of a set  $A$  has a property  $P$ ” should be described by the formula  $(\forall x)(A(x) \Rightarrow P(x))$  and the sentence “Some element of a set  $A$  has a property  $P$ ” should be described by the formula  $(\exists x)(A(x) \& P(x))$ . In this paper, the quantifiers  $\exists$  and  $\forall$  are “restricted” and have the following set theoretic semantics:

$$\begin{aligned} (\forall x : A) P(x) &\equiv_{\text{def}} A \subseteq P, \\ (\exists x : A) P(x) &\equiv_{\text{def}} A \cap P \neq \emptyset, \\ (\neg x : A) P(x) &\equiv_{\text{def}} A \cap P = \emptyset. \end{aligned} \quad (1)$$

In the cases when a quantifier is restricted to range only a finite set  $A = \{a_i \mid 1 \leq i \leq N, N \in \mathfrak{N}\}$  we will write  $\exists_A$  and  $\forall_A$ . These quantifiers have the following semantics:

$$\begin{aligned} (\forall_A x) P(x) &\equiv_{\text{def}} \bigcap_{i=1}^N \{P(a_i)\}, \\ (\exists_A x) P(x) &\equiv_{\text{def}} \bigcup_{i=1}^N \{P(a_i)\}, \\ (\neg_A x) P(x) &\equiv_{\text{def}} \bigcap_{i=1}^N \{\neg P(a_i)\}. \end{aligned} \quad (2)$$

Further, let  $A = \{a_i \mid 1 \leq i \leq N, N \in \mathfrak{N}\}$  is a finite set of elements,  $<_A \subseteq A \times A$  is a partial order relation on  $A$ ,  $\check{R} = \{\check{r}_i \mid 0 \leq i \leq N_{\check{R}}, N_{\check{R}} \in \mathfrak{N}\}$  is a finite set of numbers called *ranks*,  $F^{\check{r}} : A \rightarrow \check{R}$  is a ranking function which for each element of  $A$  assigns a unique rank, i.e.  $(\forall_A x) (\exists_{\check{R}} \check{r}') (F^{\check{r}}(x) = \check{r}') \& \forall_{\check{R}} \check{r} ((F^{\check{r}}(x) = \check{r}) \Rightarrow (\check{r} = \check{r}'))$ . Several elements of  $A$  can have the same rank, all elements of  $A$  with the same rank  $\check{r}$  form a subset denoted by  $A^{(\check{r})} = \{x \mid x \in A, F^{\check{r}}(x) = \check{r}\}$ . The dependencies between the relation  $<_A$  and the ranking function  $F^{\check{r}}$  are described by the following formulas:

$$\begin{aligned} (\forall_A x, y) (\forall_{\check{R}} \check{r}) ((\check{r} > 0) \& (F^{\check{r}}(x) = \check{r} - 1) \Rightarrow (\exists_A x_1) ((F^{\check{r}}(x_1) = \check{r}) \& (<_A(x_1, x) \& \\ & (\forall_A x_2) (F^{\check{r}}(x_2) = \check{r} - 1) \& (x_2 \neq x_1) \Rightarrow \neg <_A(x_1, x_2))))), \\ (\forall_A x, y) ((x \in A^{(\check{r})}) \& (y \in A^{(\check{r})}) \Rightarrow \neg <_A(x, y) \& \neg <_A(y, x)), \\ (\forall_A x, y) ((x \in A^{(0)}) \Rightarrow \neg <_A(x, y)). \end{aligned} \quad (3)$$

It means that in a set  $A$  any element  $x$  with the rank  $\check{r} > 0$  has an element with the rank  $\check{r} - 1$  which is greater than  $x$  with respect to the partial order relation  $<_A$ , two elements with the same rank are not comparable to each other with respect to the relation  $<_A$ , and elements with rank 0 are maximal with respect to the relation  $<_A$ .

DEFINITION 1.  $A' =_{\text{def}} \langle A, <_A, F'' \rangle$  is a *ranked set* produced by relation  $<_A$  and the ranking function  $F''$  on a finite set  $A$ .

DEFINITION 2. If  $T \subseteq A$  and

$$\begin{aligned} & (\exists_T x_1)((x_1 \in A^{(0)}) \& (\forall_T x)((x \neq x_1) \Rightarrow \neg(x \in A^{(0)}))), \\ & (\forall_T x)((x \in A^{(i)}) \& (i > 0)) \Rightarrow (\exists_T x_1)((x_1 \in A^{(i-1)}) \& <_A(x, x_1) \& \\ & (\forall_T y)((y \neq x_1) \& (y \in A^{(i-1)})) \Rightarrow \neg <_A(y, x_1))), \end{aligned}$$

then  $T' =_{\text{def}} \langle T, <_A, F'' \rangle$  is a connected acyclic single-rooted *tree on a ranked set*  $A'$  with the *set of nodes*  $T$ , the *set of edges*  $E = \{(x, y) \mid x \in T, y \in T, F''(x) = i - 1, F''(y) = i, <_A(y, x)\}$ , the single *root*  $x_1 \in T \cap A^{(0)}$ , and the set of *terminal nodes* (leaves)  $L = \{x \mid x \in T, (\forall_T y) \neg <_A(x, y)\}$ .

Let  $x \in \{T - L\}$  be a node of the tree  $T'$  for which  $F''(x) = i, 0 \leq i \leq N - 1$ . The elements of set  $\text{Ch}_x\{y \mid y \in T, F''(y) = i + 1, <_A(y, x)\}$  are called *children* of the node  $x$  and the node  $x$  is called *parent* of the nodes  $y$ .

Let  $T_{\text{and}} \subset T, T_{\text{or}} \subset T, T_{\text{and}} \cap T_{\text{or}} = \emptyset, T_{\text{and}} \cup T_{\text{or}} \cup L = T, |T_{\text{and}}| = n, |T_{\text{or}}| = m, m, n \in \aleph$  is a partition that partitiones a set of nodes of a tree  $T$  into three disjoint subsets: set of *AND* nodes  $T_{\text{and}}$ , set of *OR* nodes  $T_{\text{or}}$  and set of leaf nodes  $L$ . Let  $P$  is some property defined on a set  $T$ . In a particular node  $t \in T$  this property can be *satisfied* or *unsatisfied*. We will write  $P(t)$  if a property is satisfied in the node  $t$  and  $\neg P(x)$  otherwise. In other words,  $P$  is a linguistic variable with a set of linguistic terms  $P_{lr} = \{\text{satisfied}, \text{unsatisfied}\}$  and with the following semantics:

$$P = \{(P(t), \mu_P(t) \mid t \in T, P(t) \in P_{lr}, \mu_P : T \rightarrow \{0, 1\}\}, \quad (4)$$

where  $\mu_P$  is a membership function. The semantics of the *AND* and *OR* nodes is defined by the following formulas:

$$\begin{aligned} & (\forall_T x : T_{\text{and}})((\forall_T y : \text{Ch}_x) P(y) \Rightarrow P(x)) \& ((\exists_T y : \text{Ch}_x) \neg P(y) \Rightarrow \neg P(x)), \\ & (\forall_T x : T_{\text{or}})((\exists_T y : \text{Ch}_x) P(y) \Rightarrow P(x)) \& ((\forall_T y : \text{Ch}_x) \neg P(y) \Rightarrow \neg P(x)), \end{aligned} \quad (5)$$

It means that in an *AND* node a property  $P$  is *satisfied* iff it is *satisfied* in all children nodes and that this property in an *OR* node is *satisfied* iff it is *satisfied* at least in one children node. We define an implication relation  $r^{(P)} \subset T^N$  which in each node  $t \in T_{\text{and}} \cup T_{\text{or}}$  in a forward manner infer a value of property  $P$  from the values of  $P$  of its child nodes,

i.e.

$$\begin{aligned}
& (\forall_T x : T_{\text{and}}) ((\forall_T y : \text{Ch}_x) P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow P(x)), \\
& (\forall_T x : T_{\text{or}}) ((\exists_T y : \text{Ch}_x) P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow P(x)), \\
& (\forall_T x : T_{\text{and}}) ((\exists_T y : \text{Ch}_x) \neg P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow \neg P(x)), \\
& (\forall_T x : T_{\text{or}}) ((\forall_T y : \text{Ch}_x) \neg P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow \neg P(x)).
\end{aligned} \tag{6}$$

This inference is based on the *modus ponens* rule.

DEFINITION 3.  $T_{\text{and/or}}^{(P)} =_{\text{def}} \langle T', P, r^{(P)} \rangle$  is an AND/OR tree with respect to the property  $P$ . If  $T_{\text{or}} = \emptyset$ , the tree  $T_{\text{and/or}}^{(P)}$  becomes  $T_{\text{and}}^{(P)}$  AND tree with respect to the property  $P$ , if  $T_{\text{and}} = \emptyset$ , the tree  $T_{\text{and/or}}^{(P)}$  becomes  $T_{\text{or}}^{(P)}$  OR tree with respect to the property  $P$ .

Using the relation  $r^{(P)}$  the satisfiability (or deniability) of the property  $P$  is propagated across the whole  $T_{\text{and/or}}$  tree. We say that in  $r^{(P)}(x, y_1, \dots, y_n)$  the nodes  $y_1, \dots, y_n$  are *source nodes* and the node  $x$  is a *target node*. In other words, the relation  $r^{(P)}$  is directional, directed from the source nodes to the target node.

Let us fuzzify the  $T_{\text{and/or}}$  tree fuzzifying the set of nodes  $T$  and the set of edges  $E$ :  $\tilde{T} = \{(x, \mu_T(x) \mid x \in T, \mu_T : T \rightarrow [0, 1])\}$ , where  $\mu_T$  is a membership function;  $\tilde{E} = \{(x, y, \mu_E(x, y)) \mid (x, y) \in E, \mu_E : E \rightarrow [0, 1]\}$ , where  $\mu_E$  is a strength of edge  $(x, y)$ .

Let us fuzzify further the property  $P$  assuming that the strength of the property  $\tilde{P}$  in the node  $t$  is identical with the value of the membership function in this node:

$$\tilde{P} = \{P(t), \tilde{\mu}_P(t) \mid t \in T, P(t) \in P_{tr}, \tilde{\mu}_P(t) \equiv \mu_T(t)\}. \tag{7}$$

Defining fuzzy implication relation  $\tilde{r}^{(P)}(t, t_1, \dots, t_n)$  between source nodes  $t_1, t_2, \dots, t_n$  and target node  $t$  with respect to the property  $\tilde{P}$  on the basis of *generalized modus ponens rule to fuzzy logic*, Zade compositional rule of inference (Zade, 1975) and Mamdani (1977) or Lukasiewicz (1967) implication, the relation  $\tilde{r}^{(P)}$  in each node of the fuzzified  $T_{\text{and/or}}$  infers in a forward manner the value of property  $\tilde{P}$  starting from the given values of  $\tilde{P}$  in the leaf nodes of this tree. In a similar way, the relations can also be defined for backward and even bidirectional inferences.

DEFINITION 4.  $\tilde{T}_{\text{and/or}}^{(\tilde{P})} =_{\text{def}} \langle \tilde{T}', \tilde{P}, \tilde{r}^{(P)} \rangle$  is a fuzzy AND/OR tree with respect to the property  $\tilde{P}$ . If  $T_{\text{or}} = \emptyset$ , the tree  $\tilde{T}_{\text{and/or}}^{(\tilde{P})}$  becomes fuzzy  $\tilde{T}_{\text{and}}^{(\tilde{P})}$  AND tree with respect to the property  $\tilde{P}$ , if  $T_{\text{and}} = \emptyset$ , the tree  $\tilde{T}_{\text{and/or}}^{(\tilde{P})}$  becomes fuzzy  $\tilde{T}_{\text{or}}^{(\tilde{P})}$  OR tree with respect to the property  $\tilde{P}$ .

## 5.2. Basic Concepts

We advocate that the consideration of QoS in the context of SoES should be performed applying view reconciliation methodology. Our claim is motivated by the fact that QoS requirements in some extent are akin to software quality requirements. Basic terms in the



view reconciliation methodology are *viewpoint*, *perspective*, and *view*. In our approach, the term *viewpoint* refers to an attitude used to define EBS QoS. Following (Sommerville and Sawyer, 1997), we define two kinds of viewpoints: viewpoints associated with a particular role and viewpoints reflecting a particular standpoint. Our hypothesis is that, despite many differences between the tangible products (e.g., software products) and intangible services, the definitions of QoS may be grounded by the same attitudes to quality as the definitions of a quality for products (Garvin, 1984). In line with this hypothesis, below we define the sets of EBS quality goals (or *qualities*)  $\Gamma$ , *viewpoints*  $\Omega$  and *perspectives*  $\Pi$ . The meaning of qualities cannot be defined by formally defined properties, because they are vague concepts. For example, the meaning of “*reliable*” depends on a particular viewpoint  $\omega' \in \Omega$  and the meaning of “*highly*” cannot be defined precisely at all. The concepts a viewpoint and a perspective have precisely definable meaning, but they are evaluated using uncertain information. Thus, we consider qualities, viewpoints and perspectives as linguistic variables with a common set of linguistic terms:

$$L_{tr} = (unsatisfied, \dots, satisfied), \tag{8}$$

$$\Gamma_{tr} = \left\{ (lt, \mu(lt)) \mid lt \in L_{tr}, \mu_{\Gamma} : L_{tr} \rightarrow \mathbb{P}([0, 1]), \bigcup_{lt \in L_{tr}} \mu_{\Gamma}(lt) = [0, 1] \right\}, \tag{9}$$

where  $\mathbb{P}([0, 1])$  is a set of all possible subintervals of the interval  $[0, 1]$ . It means that membership function  $\mu_{\Gamma}$  is convex because it relates linguistic terms with the corresponding subintervals of  $[0, 1]$ . *Satisfied* and *unsatisfied* are primary linguistic terms. For a particular EBS, they are usually renamed. Other linguistic terms are generated using linguistic modifiers, which characterize the degree in which corresponding quality goal is satisfied. The number of qualities, number of linguistic terms and their labels depends on a particular EBS. For example, the labels *below low quality* (synonym to *unsatisfied*), *low quality*, *average quality*, *high quality*, *perfect quality* (synonym to *satisfied*) can be defined.

DEFINITION 5.

$$\Gamma = \{(\gamma_i, \mu_i(\gamma_i)) \mid \mu_i : \Gamma \rightarrow L_{tr}, 1 \leq i \leq N_{\Gamma}, N_{\Gamma} \in \aleph\}, \tag{10}$$

is a set of qualities. The elements of  $\Gamma$  model high-level business-oriented EBS quality requirements from a particular viewpoint  $\omega'$  (e.g., “*a service under consideration should be highly reliable*”).

DEFINITION 6.

$$\Omega = \{(\omega_i, \mu_{\Omega}(\omega_i)) \mid \mu_{\Omega} : \Omega \rightarrow [0, 1] \times L_{tr}, 1 \leq i \leq 6\}, \tag{11}$$

is a set of viewpoints, where the intuitive meaning of the viewpoints is as follows:

- **$\omega_1$  is the metaphysical viewpoint.** According to this viewpoint, a quality of EBS is a degree of excellence where excellence is defined as an abstract ideal, which shows the direction where services are heading to but will never get there.

- **$\omega_2$  is the cost-based viewpoint.** According to this viewpoint, a quality of EBS is a degree of excellence at an acceptable price.
- **$\omega_3$  is the value-based viewpoint.** According to this viewpoint, a quality of EBS is service fitness for requestor's values and preferences. It differs depending on a service requestor for whom it is defined.
- **$\omega_4$  is the pragmatic viewpoint.** According to this viewpoint, a quality of EBS is the balance of features and characteristics of service that bear on its ability to satisfy stated or implied needs of service requestor. It depends on a particular context, in which the service is consumed or, in other words, the judgment about the quality of a service depends on the aims and goals for which this service is intended to be consumed.
- **$\omega_5$  is the provider's viewpoint.** According to this viewpoint, a quality of EBS is a compliance with the stated requirements, which are mostly formulated in business and technical terms.
- **$\omega_6$  is the designer's viewpoint.** According to this viewpoint, a quality of EBS is something that is defined by the values of quantifiable and measurable internal characteristics of a service. This viewpoint assumes that the greater the amount of a desired attribute is possessed by a service, the higher is the quality of this service.

DEFINITION 7.

$$\Pi = \{(\pi_i, \mu_\Pi(\omega_i)) \mid \mu_\Pi : \Pi L_{\text{tr}}, 1 \leq i \leq 8\} \quad (12)$$

is a set of *perspectives*, where the intuitive meaning of the perspectives is as follows:

- **$\pi_1$  is the presentation perspective.** This perspective describes EBS quality from the point of view of the presentation of information produced by an EBS for service requestors. To some extent it overlaps with the data quality mentioned in Benbernou *et al.* (2010). The description includes: the relevance, granularity, and level of detail of presented information; its accuracy, consistency, completeness, and timeliness; appropriateness of its visualisation, perspicuity, and transparency for a service requestor; etc. One of the problems arising in the context of SoEA in describing QoS from this perspective is the separation of concerns between EBS itself and software, which is implemented by a service consumer. The latter describes rather the quality of the whole SoEA system than the quality of a service.
- **$\pi_2$  is the transportation perspective.** This perspective describes EBS quality from the network point of view. The description includes the response time, maximal throughput, service availability, networks reliability, etc. In SoEA context, the problem of the separation of system and service concerns arises again.
- **$\pi_3$  is the infrastructure perspective.** This perspective describes EBS quality from the implementation platform point of view. The description includes the performance, reliability, security and other related issues. In SoEA the implementation platform usually is shared among many or even all services. So, the problem of the separation of system and service concerns arises again.

- **$\pi_4$  is the web service perspective.** Web service is the software that exposes EBS. Thus, the quality of implementation of the software affects the quality of whole EBS. However, it is even more important to evaluate the software, first of all, as a service but not as a software product. It means that the description of this perspective along with the reliability, security and other product related issues must describe messaging, responsiveness, courtesy (politeness, respect for service requestor, friendliness, etc.).
- **$\pi_5$  is the application perspective.** As a rule, the functionality of an EBS service is implemented by some application, i.e. by some software component or some legacy software. In various implementation platforms the components are named differently, for example in Microsoft Windows Communication Foundation (Chapell, 2007) the components are addressed to using term *service classes*. The non-functional properties of the application affect the quality of the whole EBS. The application perspective describes the non-functional properties of a component as a software product or EBS technical quality.
- **$\pi_6$  is the data perspective.** According to (Bean) and many other sources, in SoES there is a special kind of services – SoES data service. In such services, web services encapsulate data and the supported behaviour, for example, the operations that manipulate the data. The term *data* is used here to address data stored in the enterprise's data bases as well as XML documents and various contents: "... a single data service will usually only expose or manipulate a core set of data, rather than all data for the entire enterprise" (Bean). It is obvious that the quality of encapsulated data essentially affects the QoS of the whole SoEA service as well as QoS of components processing these data. Thus, the data perspective describes EBS quality from the point of view of the quality of encapsulated data.
- **$\pi_7$  is the domain perspective.** The domain-dependent EBS quality attributes capture those properties which are specific to a particular business domain, for example, for online banking services or for online streaming multimedia services. In addition, for practical reasons, even the attributes that are defined in all business domains can be treated differently in different domains (Kritikos and Plexousakis, 2012). The specific nature of a particular domain may effect weights assigned to the values of such EBS quality attributes. For example, in online streaming multimedia services the attribute *bits-per-second* is more important than the *security*. In online banking services, vice versa, the *security* is more important than the *bits-per-second* (Pathak et al., 2005). *Inter alia*, despite the fact that media applications, including video-oriented ones, also emerge in SoEA systems, up to date they are rather marginal there (an exhaustive discussion on the QoS of video-oriented services can be found in Cisco, 2008). In SoEA system, the domain perspective describes EBS quality from the point of view of a particular (internal) enterprise domain, for example, manufacturing or human resource management.
- **$\pi_8$  is the socio-economic perspective.** This perspective describes EBS quality from the point of view of business, economic and social issues. The price of service, payment mode (e.g., kinds of accepted bank cards), legal constraints, and other similar

issues should be described. *Business effect of the service* is one of the most important socio-economic characteristics for the EBS.

In this paper, the term *perspective* refers to some aspect of the EBS quality on which any viewpoint can be focused in principle. Above listed EBS quality perspectives were chosen on the basis of the concept-centric literature survey on the QoS, which was performed in the process of writing this paper, and on our experiences in various projects. These perspectives are identical for all viewpoints, but it does not mean that all perspectives necessary cut across all viewpoints. Some viewpoints, for example, metaphysical one, may ignore some perspectives.

There is one more perspective that describes EBS quality from a service composition point of view (Abramowicz *et al.*, 2008). However, due the limited space of this paper, we ignore it. We also ignore the fact that the socio-economic perspective can be splitted into several finer-grained perspectives.

**DEFINITION 8.** A *view* is a result of integration of all perspectives taken from the same viewpoint  $\omega' \in \Omega$ .

### 5.3. Views Construction and Balancing

Let we define a relation

$$\rho^\Psi = \{(\psi, \mu_\Psi) \mid \mu_\Psi : \Psi \rightarrow [0, 1] \times L_{tr}, \rho^\Psi \subseteq \Omega \times \Pi \times \Gamma\}, \quad (13)$$

which relates viewpoints, perspectives and qualities. For each fixed viewpoint  $\omega_k \in \Omega$ ,  $1 \leq k \leq 6$ , the projection of  $\rho^\Psi$  to  $\Pi \times \Gamma$  produces a family of relations

$$\rho^{\Psi_k} = \{(\psi_{i,j}^k, \mu_{\Psi_k}(\psi_{i,j}^k)) \mid \mu_{\Psi_k} : \Psi_k[0, 1] \times L_{tr}, \psi_{i,j}^k = (\pi_i, \gamma_j), \pi_i \in \Pi, \gamma_j \in \Gamma, 1 \leq k \leq 6, 1 \leq i \leq 8, 1 \leq j \leq n, n \leq N_\Gamma\} \quad (14)$$

which relates perspectives observed from the viewpoint  $\omega_k$  and qualities on which this viewpoint focuses. In other words, for each viewpoint have a matrix of linguistic variables

$$\Psi_k = \begin{pmatrix} \psi_{1,1}^k & \cdots & \psi_{1,n}^k \\ \vdots & \ddots & \vdots \\ \psi_{8,1}^k & \cdots & \psi_{8,n}^k \end{pmatrix}, \quad (15)$$

where  $\psi_{i,j}^k = (\pi_i, \gamma_j)$ , where  $\pi_i \in \Pi$ ,  $\gamma_j \in \Gamma$ ,  $1 \leq k \leq 6$ ,  $1 \leq i \leq 8$ ,  $1 \leq j \leq n$ ,  $n \leq N_\Gamma$ .

**DEFINITION 9.**

$$X = \{(\chi_i, \mu_X(\chi_i)) \mid \mu_X : X \rightarrow L_{tr}, 1 \leq i \leq N_X, N_X \in \aleph\} \quad (16)$$

is a finite set of linguistic variables called *EBS quality characteristics*.

DEFINITION 10.

$$\rho_{\text{eqib}}^X = \{((\chi_1, \chi_2), \mu_{\rho_{\text{eqib}}^X}(\chi_1, \chi_2), \text{label}) \mid \mu_{\rho_{\text{eqib}}^X} : X \rightarrow [0, 1], (\chi_1, \chi_2) \in X \times X, \mu_{\Gamma}(\chi_1) + \mu_{\Gamma}(\chi_2) \leq C_{\text{eqib}}^{(\chi_1, \chi_2)} \leq 1, \text{label} \in \{\ll, \gg, \sim, \llsim, \sim\gg\}\} \quad (17)$$

is a labelled *equilibrium relation* defined on a set  $X$ .  $C_{\text{eqib}}^{(\chi_1, \chi_2)}$  is an equilibrium constant, which means that a sum of the lengths of subintervals  $\mu_{\Gamma}(\chi_1)$  and  $\mu_{\Gamma}(\chi_2)$  cannot exceed the length defined by this constant, which, in turn, cannot exceed the length of interval  $[0, 1]$ .

The label of this relation tells how, if it is necessary, the lengths of subintervals  $\mu_{\Gamma}(\chi_1)$  and  $\mu_{\Gamma}(\chi_2)$  should be changed in order to preserve the equilibrium defined by  $C_{\text{eqib}}$ : (1)  $\ll$  means that the length of subinterval  $\mu_{\Gamma}(\chi_1)$  should be changed; (2)  $\gg$  means that the length of subinterval  $\mu_{\Gamma}(\chi_2)$  should be changed; (3)  $\sim$  means that the lengths of both subintervals should be changed proportionally; (4)  $\llsim$  means that the lengths of both subintervals should be changed taking preference to  $\mu_{\Gamma}(\chi_2)$ ; and (5)  $\sim\gg$  means that the lengths of both subintervals should be changed taking preference to  $\mu_{\Gamma}(\chi_1)$ . The number of *EBS quality characteristics* under consideration and their nature depend on a particular EBS under consideration.

In an analogous way we define total equilibrium relations  $\rho_{\text{eqib}}^{\Omega}$  on a set  $\Omega$  and  $\rho_{\text{eqib}}^{\Psi_k}$  on each set  $\Psi_k$ :

$$\rho_{\text{eqib}}^{\Omega} = \{((\omega_1, \omega_2), \mu_{\rho_{\text{eqib}}^{\Omega}}(\omega_1, \omega_2)) \mid \mu_{\rho_{\text{eqib}}^{\Omega}} : \Omega \rightarrow [0, 1], (\omega_1, \omega_2) \in \Omega \times \Omega, (\omega_1)_{\text{prj}} + (\omega_2)_{\text{prj}} \leq C_{\text{eqib}}^{(\omega_1, \omega_2)} \leq 1\}, \quad (18)$$

$$\rho_{\text{eqib}}^{\Psi_k} = \{((\psi_{i,j}^k, \psi_{l,r}^k), \mu_{\rho_{\text{eqib}}^{\Psi_k}}(\psi_{i,j}^k, \psi_{l,r}^k)) \mid \mu_{\rho_{\text{eqib}}^{\Psi_k}} : \Psi_k \rightarrow [0, 1], (\psi_{i,j}^k, \psi_{l,r}^k) \in \Psi_k \times \Psi_k, (\psi_{i,j}^k)_{\text{prj}} + (\psi_{l,r}^k)_{\text{prj}} \leq C_{\text{eqib}}^{(\psi_{i,j}^k, \psi_{l,r}^k)} \leq 1\}, \quad (19)$$

where  $(x)_{\text{prj}}$  stands for the projection of a relation  $x$  to  $L_{\Gamma}$ . In this relation the lengths of intervals should be changed by balancing. Balancing takes in account the weights defined by the projection of relation  $x$  to  $[0, 1]$ . It is a cyclic process, which proceeds until the equilibrium among all elements is reached. If it is impossible to do, the decision should be taken by decision maker using an interactive procedure.

Let  $X' = \langle X, <_X, F^{<} \rangle$  is a ranked set on  $X$ . For each element  $\psi_{i,j}^k$  of matrix  $\Psi_k$  we define a fuzzy AND tree

$$\tilde{T}_{\text{and}}^{(\psi_{i,j}^k)} = \langle \tilde{X}', \tilde{\psi}_{i,j}^k, \tilde{r}^{(\psi_{i,j}^k)} \rangle, \quad \text{where } 1 \leq k \leq 6, 1 \leq i \leq 8, 1 \leq l \leq 6, 1 \leq j \leq n, n \leq N_{\Gamma} \quad (20)$$

of EBS quality characteristics. The value of a linguistic variable  $\tilde{\psi}_{i,j}^k$  describes a  $\gamma_j$  aspect of EBS (e.g., reliability or security) from the perspective  $\pi_i$  of the viewpoint  $\omega_k$ . The

equilibrium relation  $\rho_{\text{eqlb}}^X$  can relate any two leaf nodes of the same or different trees. Using the relations  $\tilde{r}^{(\psi_{i,j})}$  and  $\rho_{\text{eqlb}}^X$ , the given linguistic values can be propagated from leaf nodes up to the root of the tree  $\tilde{T}_{\text{and}}^{(\tilde{\psi}_{i,j}^k)}$ .

The columns of the matrix (15) are vectors, which for each quality describe its evaluations from each perspective defined for viewpoint  $\omega_k$ . Using the relation (19), these qualities can be balanced and, using relations  $\tilde{r}^{(\psi_{i,j})}$  defined for quality by formula (20), the obtained values in each tree can be propagated in a backward manner to leaf nodes. Union of balanced trees produces the final trees for each quality observed from the viewpoint  $\omega_k$ . The value of  $\omega_k$  can be calculated considering the roots of these trees as child nodes of the node  $\omega_k$ . In this way, we finish the construction of a view  $v_k$ . Finally, using the relation  $\rho_{\text{eqlb}}^\Omega$ , views (i.e. values of elements of  $\Omega$ ) can be balanced. Resulting view is chosen using formula  $\max((\mu_\Omega(\omega_1)_{\text{prj}}, \dots, (\mu_\Omega(\omega_5)_{\text{prj}}))$ . The linguistic value describing quality defined by the resulting view can be propagated in a backward manner to the leaf nodes.

#### 5.4. Problem Statement and Solution Procedure

Let

- $X$  is a set of linguistic variables referred to as *EBS quality characteristics* defined by formula (16);
- $\rho_{\text{eqlb}}^X$  is a labelled equilibrium fuzzy relation on  $X$  defined by formula (17);
- $\Omega$  is a set of weighted linguistic variables referred to as *viewpoints* to EBS quality defined by formula (11);
- $\rho_{\text{eqlb}}^\Omega$  is a total equilibrium relation on  $\Omega$  defined by formula (18);
- $\Pi$  is a set of weighted linguistic variables describing different aspects of EBS quality (e.g. network aspect, data aspect, etc.) and referred to as *perspectives* defined by formula (12);
- $\Gamma$  is a set of given EBS goals (or *qualities*) defined by formula (10);
- $\rho^\Psi$  is a fuzzy relation defined by formula (13), which relates viewpoints, perspectives and qualities and  $\rho^{\Psi_k}$ ,  $1 \leq k \leq 6$  is a family of relations produced by projection of  $\rho^\Psi$  to  $\Pi \times \Gamma$ ;
- $\Phi_1 = \{\rho_{\text{eqlb}}^{\Psi_k} \mid 1 \leq k \leq 6\}$  is a family of total equilibrium relations (each on corresponding  $\rho^{\Psi_k}$ ) defined by formula (19);
- $\Phi_2 = \{\tilde{T}_{\text{and}}^{(\tilde{\psi}_{i,j}^k)} \mid 1 \leq k \leq 6, 1 \leq i \leq 8, 1 \leq k \leq 6, 1 \leq j \leq n, n \leq N_\Gamma\}$ , is a family of fuzzy AND trees of EBS quality characteristics defined by formula (20), where each tree describes a quality  $\gamma_i \in \Gamma$  evaluated from a perspective  $\pi_j \in \Pi$  and observed from a viewpoint  $\omega_k \in \Omega$ ;

then a tuple  $\langle X, \rho_{\text{eqlb}}^X, \Omega, \rho_{\text{eqlb}}^\Omega, \Pi, \Gamma, \rho^\Psi, \Phi_1, \Phi_2, \text{Input}, \text{Output} \rangle$ , where *Input* is a set of initial linguistic values of variables from  $X$ , describing for each view the bottom level EBS quality characteristics and *Output* is a final linguistic value of the balanced view  $v$  on EBS quality and quality characteristics of all levels describing this view, is a view balancing problem.

In our framework, this problem is solved by the following procedure:

- for each viewpoint to set linguistic values of input variables and, using fuzzy relation  $\rho_{\text{eqib}}^X$ , to resolve conflicts among these values;
- for each fuzzy tree from the family  $\Phi_2$  to propagate the defined values forward up to the root of this tree;
- for all viewpoints using the relations from the family  $\Phi_1$ , to resolve conflicts between values of linguistic variables describing qualities, which arise evaluating these qualities from different perspectives;
- for each fuzzy tree from the family  $\Phi_2$  to propagate the new values of linguistic variables backward to leaf nodes;
- for each viewpoint applying fuzzy graphs union operation to unite trees which describe qualities from different perspectives;
- to infer value of each linguistic variable  $\omega \in \Omega$ ;
- using relation  $\rho_{\text{eqib}}^\Omega$ , to resolve conflicts between values of  $\omega \in \Omega$ ;
- for all qualities of each viewpoint to propagate new values backward to leaf nodes;
- to choose from  $\Omega$  the variable  $\omega'$  with the greatest weight and to append to the tree associated with this variable missing sub-trees from the trees associated with the other viewpoints;
- to do linguistic approximation of fuzzy values of all output variables.

The description of algorithms, which are used in the above problem solution procedure, is out of scope of this paper. The paper presents only the model of the view-based EBS quality evaluation framework, in terms of which these algorithms are described.

## 6. Conclusions

Current SOA QoS and EBS quality models do not take into account all the quality aspects. The emphasis as a rule is on the technical and technological considerations, while ignoring the other attitudes required to define EBS quality and to consider quality conceptions of the all stakeholders of a service-oriented enterprise system. The paper presents a model which models a view-based framework intended to balance the different viewpoints on a service quality. This framework is based on the assumption that despite differences between the tangible products (e.g. software products) and intangible services, the definitions of a QoS can be grounded by the same attitudes to quality as the definitions of a quality for products. The main conclusions of the paper are as follows.

The view integration and reconciliation methodology, which is widely accepted in the software and enterprise systems requirement engineering, can also be successfully applied to solve a view balancing problem in service quality requirements formulation context. The philosophy of goal-oriented modelling, especially  $i^*$  methodology, proved also to be valuable in this context. However, quality is a vague concept and  $i^*$  techniques cannot be applied directly to evaluate the quality goals. For this reason, these techniques should be modified using fuzzy set theory and fuzzy logic. Another difference between  $i^*$  and our approaches lies in our approach focus on a balancing of quality goals, i.e.,

on some equilibrium between these goals. For this reason, the goal-oriented modelling methodology should be extended by adding an equilibrium relation.

**Acknowledgments.** This research has been supported by the project “Theoretical and Engineering Aspects of E-Service Technology Development and Application in High-Performance Computing Platforms” (No. VP1-3.1-ŠMM-08-K-01-010) funded by the European Social Fund.

## References

- Abramowicz, W., Hofman, R., Suryan, W., Zyskowski, D. (2008). SQuaRE based web services quality model. In: Esantis, S.I., Ao, O., Castillo, C., Douglas, Feng, D., Lee, A.-J. (Eds.), *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2008), 19–21 March 2008, Hong Kong, LNECS*, Vol. 2168. Newswood Limited, Hong Kong, pp. 827–835.
- Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E. (2010). Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 25(8), 841–877. doi:10.1002/int.v25:8.
- Bean, J. (2011). Data services: resolving the integration challenge with SOA, white paper. In: *Dataversity, Web-shire Conferences*, 1 April 2011. <http://www.dataversity.net/data-services-challenge-resolving-the-integration-challenge-with-soa/>.
- Benbernou, S., Brandic, I., Cappiello, C., Carro, M., Comuzzi, M., Kertesz, A., Pleban, P. (2010). Modeling and negotiating service quality. In: Papazoglou, M., Pohl, K., Parkin, M., Metzger, A. (Eds.), *Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems. LNCS*, Vol. 6500. Springer, Berlin, pp. 157–209.
- Bianchini, D., De Antonellis, V., Melchiori, M. (2004). QoS in ontology-based service classification and discovery. In: *Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04)*. IEEE Press, New York, pp. 145–150.
- Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R. (2005). *Service-Oriented Architecture (SOA) Compass – BusinessValue, Planning, and Enterprise Roadmap*. IBM Press.
- Chapell, D. (2007). *Introducing Windows Communication Foundation*. Microsoft Corporation.
- Chung, L., Nixon, B., Yu, E. (1995). Using non-functional requirements to systematically support change. In: *Proceedings of the 2nd IEEE International Symposium on Requirements Engineering (RE'95)*, York, England, March 1995. IEEE Computer Society, Washington, pp. 132–139.
- Cisco. (2008). *Enterprise QoS Solution Reference Network Design Guide, Version 4.0*. Cisco Systems, Inc.
- Deissenboeck, F., Wagner, S., Pizka, M., Teuchert, S., Girard, J.-F. (2007). An activity-based quality model for maintainability. In: *Esantis Proceeding of the IEEE International Conference on Software Maintenance, (ICSM '07), Paris, 2–5 October 2007*. IEEE Computer Society Press, Washington, pp. 184–193. doi:10.1109/ICSM.2007.4362631.
- Döweling, S., Schmidt, B., Göb, A. (2012). A model for the design of interactive systems based on activity theory. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12), 11–15 February 2012, Seattle, Washington, USA*. ACM, New York, pp. 530–548. doi:10.1145/2145204.2145287.
- Farroha, B.S., Farroha, D. (2007). Policy-based QoS requirements in a SOA enterprise framework-an investigative analysis. In: *The Proceedings of IEEE Military Communications Conference (MILCOM 2007)*. IEEE Press, New York, pp. 1–7.
- Franch, X. (2006). On the Quantitative Analysis of Agent-Oriented Models. In: Dubois, E., Pohl, K. (Eds.), *CAiSE'06*. Springer, Heidelberg, pp. 495–509.
- Garvin, D. (1984). What does “product quality” really mean? *Sloan Management Review*, 26(1), 25–43.
- Giorgini, P., Nicchiarelli, E., Mylopoulos, J., Sebastiani, R. (2002). Reasoning with goal models. In: Paccapietra, S., March, S., Kambashi, Y. (Eds.), *Conceptual Modeling – Proceedings of the 21st International Conference on Conceptual Modeling (ER'02)*, Tampere, Finland, 7–11 October 2002, LNCS, Vol. 2503. Springer, Berlin, Heidelberg, pp. 167–181.
- Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R. (2003). Formal reasoning techniques for goal models. *Journal on Data Semantics(I)*, 1–20.



- Giorgini, P., Mylopoulos, J., Sebastiani, R. (2004). Simple and minimum-cost satisfiability for goal models. In: Persson, A., Stirna, J. (Eds.), *CAiSE 2004, LNCS*, Vol. 3084. Springer, Heidelberg, pp. 20–35.
- Grönroos, C. (1982). *Strategic Management and Marketing in the Service Sector*. Research report, Swedish School of Economics and Business Administration, Helsingfors.
- Horkoff, J., Yu, E. (2009). A qualitative, interactive evaluation procedure for goal- and agent-oriented models. In: Yu, E., Eder, J., Rolland, C. (Eds.), *Proceedings of the CAiSE'09 Forum at the 21th International Conference on Advanced Information Systems Engineering, Amsterdam, The Netherlands, 8–12 June 2009*, pp. 19–24.
- Horkoff, J., Yu, E. (2010). A framework for iterative, interactive analysis of agent-goal models in early requirements engineering. In: Castro, J., Franch, X., Mylopoulos, J., Yu, E. (Eds.), *Proceedings of the 4th International i\* Workshop (iStar-2010), Hammamet, Tunisia, 7–8 June 2010*, pp. 52–56.
- Yu, E. (1995). *Modelling Strategic Relationships for Process Reengineering*. PhD thesis. University of Toronto, Department of Computer Science, Toronto.
- Kavakli, E., Loucopoulos, P. (2004). Goal modelling in requirements engineering: analysis and critique of current methods. In: Krogstie, J., Halpin, T., Siau, K. (Eds.), *Information Modeling Methods and Methodologies*. Idea Group Publishing, London, pp. 102–124.
- Kim, E., Lee, Y. (2005). *Quality Model for Web Services (WSQM-2.0)*. September 2005, OASIS.
- Kim, E., Lee, Y., Kim, Y., Park, H., Kim, J., Moon, B., Kang, G. (2012). *Web Services Quality Factors, Version 1.0, Candidate OASIS Standard*. 31 October 2012, OASIS.
- Kritikos, K., Plexousakis, D. (2008). Requirements for QoS-based web service description and discovery. *International Journal of Web Services Practices*, 3(2), 72–82.
- Leite, J. (1988). *Viewpoint resolution in requirements elicitation*. PhD thesis, Department Computer Science, University California, Irvine.
- Letier, E., van Lamsweerde, A. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. In: *Proceedings of the 12th ACM International Symposium on the Foundations of Software Engineering (FSE 2004)*. ACM, New York, pp. 53–62.
- Liu, L., Yu, E. (2004). Designing information systems in social context: a goal and scenario modelling approach. *Information Systems*, 29(2), 187–203.
- Lukasiewicz, J. (1967). Philosophical remarks on many-valued systems of propositional logic. In: McCall, S. (Ed.), *Polish Logic 1920–1939*. Clarendon, Oxford, pp. 40–65.
- Mamdani, E. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26(12), 1182–1191.
- Papaoannou, I., Tsesmetzis, D., Roussaki, I., Anagnostou, M. (2006). A QoS ontology language for web-services. In: *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, Vienna, Austria, 18–20 April 2006. IEEE Computer Society, Washington, pp. 101–106.
- Parasuraman, A., Zeithaml, V.A., Berry, L. (1985). A conceptual model of service quality and its implications for future research. *Journal of Marketing*, 49(1985), 41–50.
- Pathak, J., Koul, N., Caragea, D., Honavar, V. (1985). Discovering web services over the semantic web. In: *Proceedings of the 7th ACM International Workshop on Web Information and Data Management*, Bremen, Germany, pp. 1–10.
- Rolland, C., Prakash, N. (2000). Bridging the gap between organisational needs and ERP functionality. *Requirements Engineering*, 5(3), 180–193.
- Ross, D.S. (1977). Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, 1(3), 363–386.
- Sebastiani, R., Giorgini, P., Mylopoulos, J. (2004). Simple and minimum-cost satisfiability for goal models. In: Persson, A., Stirna, J. (Eds.), *Advanced Information Systems Engineering, LNCS*, Vol. 3084. Springer, Heidelberg, pp. 20–35.
- Sommerville, I., Sawyer, P. (1997). Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering* 3, 101–130.
- Stirna, J., Persson, A. (2007). Ten years plus with EKD: reflections from using an enterprise modeling method in practice. In: Pernici, B., Gulla, J. (Eds.), *Proceedings of the 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2007)*, Vol. 365, pp. 99–108.
- Usmani, S., Azeem, N., Samreen, A. (2011). Dynamic service composition in SOA and QoS related issues. *International Journal of Computer Technology and Applications*, 2(5), pp. 1315–1321.
- van Lamsweerde, A. (2001). Goal-oriented requirements engineering: a guided tour (invited paper). In: *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE 2001)*. IEEE Press, Washington, pp. 249–263.

- Wang, G., Chen, A., Wang, C., Fung, C., Uczekaj, S. (2004). Integrated quality of service (QoS) management in service-oriented enterprise architectures. In: *The Proceedings of 8 IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004)*, pp. 21–32.
- Webster, J., Watson, R.T. (2002). Analyzing the past to prepare for the future: writing a literature review. *MIS Quarterly*, 26(2), xiii–xxiii.
- Wu, Q., Li, P. (2010). Study and implement of dynamic routing based on QoS in enterprise service bus. *Journal of Computational Information Systems*, 6(7), 2093–2098.
- Zade, L. (1975). Fuzzy logic and approximate reasoning. *Synthese*, 30(3–4), 407–428.

**A. Lupeikiene** is a docent, senior researcher at the Institute of Informatics and Mathematics of Vilnius University. Her main research interests include information systems engineering, service-oriented enterprise systems engineering, software engineering, and decision making.

**J. Miliauskaite** is a PhD student at the Institute of Informatics and Mathematics of Vilnius University. Her areas of research include quality of service, service-oriented enterprise systems.

**A. Caplinskis** is a professor, principal researcher and the head of the Software Engineering Department at the Institute of Informatics and Mathematics of Vilnius University. His main research interests include software engineering, service-oriented enterprise systems engineering, information system engineering, legislative engineering, and knowledge-based systems.

## **Modelis integruoti skirtingais požiūriais grindžiamus įmonės sistemos teikiamos verslo paslaugos kokybės reikalavimus**

Audronė LUPEIKIENĖ, Jolanta MILIAUSKAITĖ, Albertas ČAPLINSKAS

Paslaugų stiliaus architektūros įmonių sistemos susiformavo kaip dviejų sričių – įmonės sistemų architektūros ir paslaugų stiliaus architektūros – simbiozės rezultatas. Tokių sistemų atsiradimas sąlygojo daug naujų tyrimų problemų paslaugų stiliaus architektūros įmonių sistemų inžinerijoje. Viena iš svarbiausių yra paslaugų kokybės nusakymas, jos specifikavimas ir vertinimas. Straipsnyje pateikiamas modelis, leidžiantis apimti ir subalansuoti skirtingais požiūriais grindžiamus įmonės sistemos teikiamų verslo paslaugų kokybės reikalavimus.