SOA BASED E-BUSINESS SYSTEMS DESIGN

Edzus Zeiris and Maris Ziema
Faculty of Computer Science and Information Technology, Riga Technical University, Meza 1/3, LV-1048, Riga, Latvia

Keywords: SOA, e-Services, e-Business, Web services, QoS, Design methods, Multi-criteria optimisation, System architecture, Algorithm graph.

Abstract: The development of e-business promotes the creation of new e-services; consequently, ways of fast and quality designing of e-services are required. When developing e-services in SOA, it is very important to build the architecture of e-business system that makes the e-service compliant with all and any quality criteria (QoS) specified for it, which would expand its usability; furthermore, it is necessary to implement any changes swiftly and in good quality to be able to adjust to the rapidly changing business environment. This means that effective design methods should be used in creating e-business systems and e-services, which would ensure the building of an acceptable e-business system architecture. A drawback of the existing methods is the subjective opinion of the system's architect, and that may not always lead to the best solution. Therefore, it is possible to apply the Quality Attributes Driven Design method for web services that is based on the use of formal optimisation methods. Initially the e-service is described as an algorithm graph, and by segmenting its vertices in all possible ways the web service graphs are obtained. The segmentation of the algorithm graph means that all the possible solutions that can affect the quality of the e-service system architecture are dealt with. Using multi-criteria optimisation, a Pareto optimality set is obtained from all the web service graphs. Web service graphs of the obtained Pareto optimality set can serve as the basis for selecting an acceptable e-business system architecture.

1 INTRODUCTION

The global dynamic advancement of information technologies has reached a level where there is high demand for simple and effective means for receiving, processing, storing and exchanging information. Information technologies have become widely available to the public, and it creates demand for e-services that facilitate the advancement of e-business. There is a particular need for e-services in the public sector (government and local government spheres), where there is a very wide range of customers and the number of services is high. More and more government and local government institutions offer their customers services that can be delivered electronically. Compared to government and local government institutions, the advancement of e-services is faster in the private sector. The reason for the advancement of the e-business environment in the private sector could be cost-effectiveness. In the long run, the use of e-services saves resources, which means profit for private commercial structures. Areas where commercial e-services are most advanced are internet banking, internet shops, information publishing, communications etc. In the public sector, the implementation of e-services is hindered by the bureaucracy and the existing legislation procedures that require a lot of work and time to make them electronic.

Undoubtedly, the availability, range and variety of e-services are required for the convenience of the customers (service users) and to save resources of the service providers. If the range of e-services becomes wider, the clients will benefit from saving their time and resources; therefore, it is important to expand the range of e-services both in the public and private sectors. Development costs are one of the factors that hinder the implementation of e-services. Another factor, which, in terms of making services electronic, affects more the public sector, is the requirements to change the legislation and procedures. Also, it is essential that simultaneous processing of a number of communication channels has to be considered when making services electronic in the public sector. Since services of the public sector must be available to all customers, it

Zeiris E. and Ziema M. (2010).
SOA BASED E-BUSINESS SYSTEMS DESIGN.
In *Proceedings of the International Conference on e-Business*, pages 265-272
DOI: 10.5220/0002921102650272

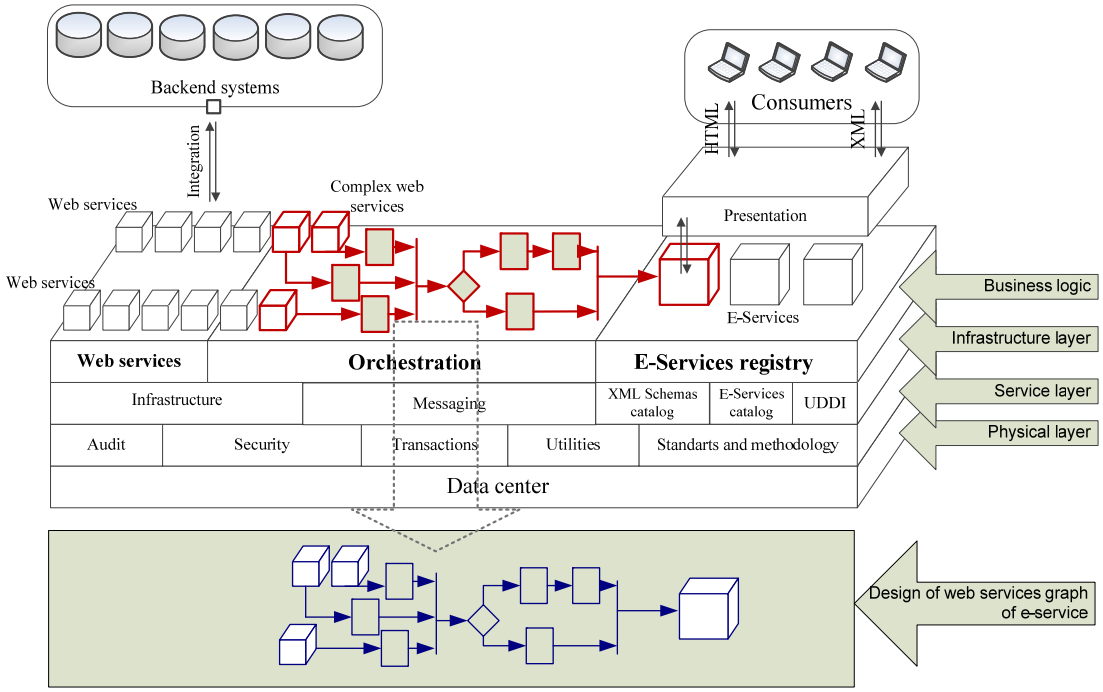

Figure 1: e-Business system architecture

has to be ensured that they can receive services on the Internet, in presence, by phone, mail, e-mail etc.

In the development of e-services, another problem is the rapidly changing business processes that require relevant modifications of IT systems, inter alia e-services. To develop e-service systems in such a dynamic environment, a fast and quality design method that makes it possible to adapt to the business processes and satisfies all the requirements specified for the e-service system is required. It is of great importance to meet the quality of service (QoS) requirements that is the basis for developing a successful e-business.

Currently, the provision of services in the form of web services is advancing at a rapid pace. This is due to the fact that the business-related functionality is offered as ready-made applications (services). Usually a number of web services are combined in e-services. To create an e-service, it is necessary to strictly define service interfaces and their functionality and to organise the performance of the e-service in certain succession. It means that e-business systems are created on the basis of SOA.

Several quality criteria can be specified for the e-business system, and its architecture must comply with them: development costs, maintenance costs, integrity, scalability, security etc. To evaluate the quality of system architecture, ISO 9126 Standard for Software Engineering-Product Quality Assessment and its quality metrics and guidelines for the use thereof can be used.

Quality criteria are usually mutually controversial; therefore, to be able to comply with all the quality criteria set for the system, it is necessary to design a system architecture that, to a larger or lesser extent, meets all the requirements. In such cases, it is not enough to use functionality-based or object-oriented design approach. It is necessary to apply methods that solve architecture design quality problem.

To ensure the creation of an acceptable e-business system architecture and to be able to adjust e-services to the constantly changing business environment and its requirements, it is possible to employ formal optimisation methods in the designing of the system architecture.

2 E-BUSINESS SYSTEM ARCHITECTURE

On the whole, an e-business architecture, within which e-services are designed, is rather complex and consists of a number of parts. It includes all the components, conditions and mutual links required to design e-services. Fig.1 shows how the systems and system components used in the service are combined into a single e-business system architecture.

Fig. 1. shows how the systems and system components used in the service are combined into unified e–service system architecture. For every data object that is required in the implementation of e–service, an XML Schemas Catalog has to be developed. Data call from the relevant functional system is done by means of the web services. When web service calls are done, also metadata that describe the request are sent. Also any information that is required for the audit trails is sent together with the metadata. Web services can be distributed into two groups: simple and complex. Complex web services in fact are logical combinations of several simple web services that result from the process integration requirements; a combination may comprise simple services of one or several independent functional systems. Complex web services can be executed by using a BPEL processor. A BPEL processor is used as the orchestration (integration) environment for the e–service's web services. Portals, one–stop agency applications etc ensure the delivery of e–services to the users. E–service entry forms, stop points, information on payments and execution results are transferred through HTML or HML pages that can be used in the portal to implement the service using the XSLT transformation. Web service and e–service holders, i.e. institution specialists and system administrators who are responsible for the maintenance and development of web services and e–services, must have a possibility to intercommunicate on various issues connected with the execution and advancement of web services and e–services. Also, asynchronous e–services have to be executed. Messaging systems are designated for this purpose. The messaging system enables working with text messages and work tasks. The application is inter-integrated with the e–services register, from which it receives data on XML schemas, web services and e–services. On the other hand, the messaging application is a client of the orchestrations as messages on the execution of web services and e–services are received from them. Data on all XML schemas, web services and e–services are entered in a single register called E–services Register. The register keeps all the versions of schemas, services and e–services so that this information is accessible to anyone who is engaged in the development and advancement of e–services.

The main problems in the e-business system architecture are related to the e-service's data standardization, audit, the creation of a web service catalog (UDDI), the execution of asynchronous e-services, security and the development of e-service orchestration that is connected with web service design. If e-business environment components that ensure quality e-services are once created, they can be later used repeatedly, whereas web services of particular e-services and their orchestration must be developed for every e-service individually. Furthermore, the development of web services and their orchestration significantly affects the e-service's performance metrics and, consequently, the overall quality of e-service (QoS). To design web services and their orchestration, the Quality Attributes Driven Web Services Design Method, which ensures swift and quality designing, can be used.

3 QUALITY ATRIBUTES DRIVEN WEB SERVICES DESIGN METHOD

The offered e – business system architecture design method consists of the following steps (Fig. 2.):

1. Initially it is necessary to create the e-service algorithm graph G that describes the e-service to be designed;
2. In the e-service algorithm graph, it is necessary to determine the possible restrictions for inter-segmentation of vertexes;
3. By recursively segmenting the e-service algorithm graph, the web service graphs $X = \{G'\}$ are obtained that are then used as the basis for the development of the e-service system architecture; The e-service algorithm graph is assumed as the initial web service graph;
4. The numerical vales of quality metrics of the obtained web service graphs are calculated;
5. From all the web service graphs, the Pareto optimality set $P = \{G'^*\}$ is obtained;
6. Web service graphs of the obtained set P can serve as the basis for selecting an acceptable e-business system architecture. The web service graphs that are contained in the obtained Pareto optimality set can be designed in detail and implemented in the e-business system architecture;
7. In most cases, the Pareto optimality set P contains more than one solution. If the obtained set is sufficiently small, the selection of web service graphs from the set can be left to the system designer. However, if there are many possible solutions, then in order to select a particular web service graph, the criteria can be decreased or combined with another optimisation or design method.

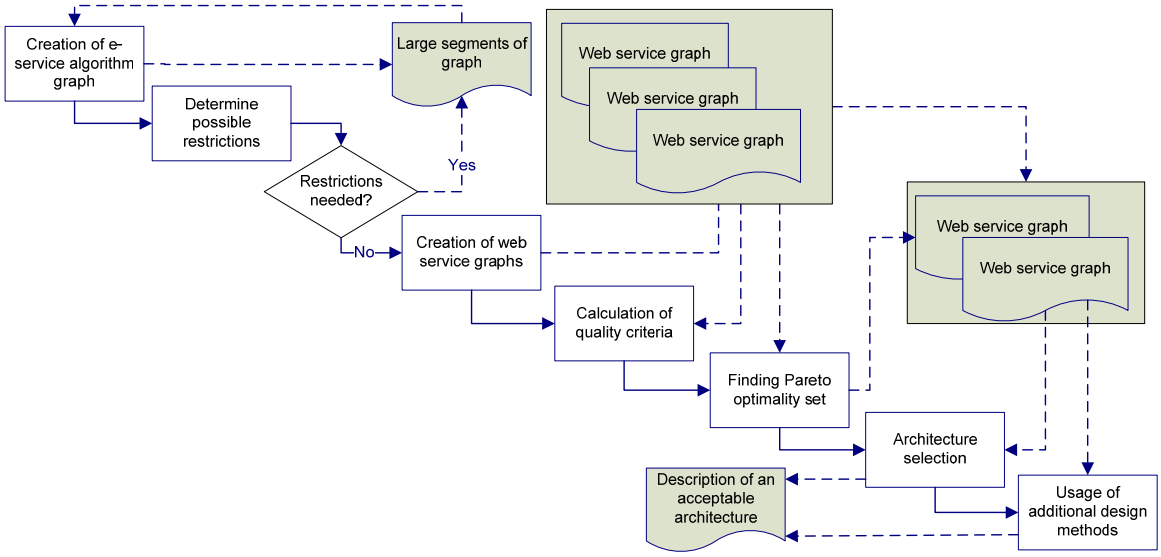

Figure 2: Quality attributes driven web services design method.

3.1 e-Service Algorithm Graph

The e-service algorithm graph is defined as an oriented graph $G = (S, L)$, where $S = \{s_1, s_2, s_3, \cdots s_n\}$ is a final set – graph vertexes, which, to their substance, are the executable operations of the e-service algorithm, and $L \subset S \times S$ are edges. The edge $l = (s_j, s_k)$ in the graph means that in the execution of the e-service algorithm the execution of the operation s_j is followed by the execution of the operation s_j. The edges in the e-service algorithm graph show the information flow (Fig. 3). When creating the e-service algorithm graph that can be used as the basis for developing the e-business system architecture, some of the restrictions related to SOA have to be considered:

1. Every algorithm graph vertex s_i must perform an independent executable activity. It means that the vertex operates in one transaction and is not connected with the algorithms executed in other vertexes. This condition is connected with high cohesion and minimal dependence.
2. Every vertex must contain at least one executable operation $M_{s_i} = \{m^0_{s_i}, m^1_{s_i}, m^2_{s_i}, \cdots m^g_{s_i}\}$. In practice, this is related to the algorithm implementation methods.
3. 3. The operations that repeat during the execution of the algorithm may not be implemented in the graph as various vertexes $\forall s_i, s_j,\ s_i \neq s_j,\ s_i \in S,\ s_j \in S,\ M_{s_i} \cap M_{s_j} = \emptyset$.

   It is required in order to ensure high cohesion and to exclude, from the very beginning, the processing of unnecessary variants.

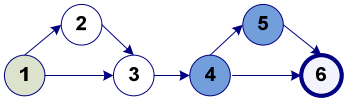

Figure 3: Example of e–service algorithm graph.

3.2 Web Services Graph

The basis for selecting the e-business system architecture is the selection of web services. It means that the e-service algorithm has to be implemented as web services in order to design the e-business system architecture. The algorithm graph can be realised as a web service graph in several ways: from realising every algorithm vertex as an individual web service to realising all the algorithm vertexes as one web service. It means that various e-service algorithm graph segments can be realised as web services, in this way changing the e-service architecture and affecting various e-service execution metrics (Fig. 4.).

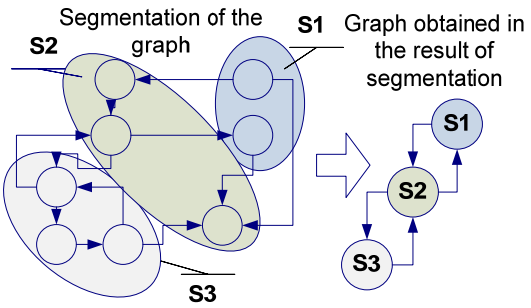

Figure 4: Segmentation of the graph.

To segment the graph and describe it formally, we will deal with the graph *G* as the depiction Γ in the vertexes set S, attributing to every vertex a sub-set ($s_i \subset S$), which actually consists of the vertexes that can be reached from the vertex s_i. Γs_i is the edges coming out from the vertex s_i, and $\Gamma^{-1}s_i$ is the edges going into s_i. For such e-service algorithm graphs, there is at least one vertex $s_0 (s_0 \in S)$, which none of the edges $\Gamma^{-1}s_0 = \emptyset$, which we will call the origin of the e-service algorithm, enters into; similarly, there exists at least one vertex $s_r (s_r \in S)$, from which none of the edges $\Gamma s_r = \emptyset$, which we will call the result of the e-service algorithm, is going out. To find all the possible web service graphs from which the one to be implemented in the e-business system architecture is selected, initially the given algorithm graph G is assumed as the first possible web service graph. All other web service graphs are obtained by segmenting the initial web service graph, which concurs with the algorithm graph. Since several segmentations are possible, the set $X = \{G'\}$ that contains all the possible graphs that are recursively derived from the initial web service graph *G* is defined. Graph transformations are done by merging the vertexes. The combination s' of the vertexes s_i and s_j is as the combination of the ingoing and outgoing edges of both vertexes. $\Gamma s' = \Gamma s_i \cup \Gamma s_j$ and $\Gamma^{-1}s' = \Gamma^{-1}s_i \cup \Gamma^{-1}s_j$. The set of methods in the new created vertex develops as follows: $M_{s'} = M_{s_i} \cup M_{s_j}$.

In practice, when creating the web service algorithm graph, there are a number of restrictions, e.g. it is necessary to integrate the functionality of various systems in one e-service, or it is required to use already existing web services. Therefore, restrictions (vertex feature $I(s_i)$) can be applied to the algorithm graph, which is assumed as the original web service graph. In the given example (Fig. 3.) vertexes 4 and 5 are created in another system, therefore they are marked in another colour. Consequently, $I(s_1) = I(s_2) = I(s_3) = I(s_6) = 0$ and $I(s_4) = I(s_5) = 1$. In this case, the combination of vertexes s' only is possible if $I(s_i) = I(s_j)$. Considering that the algorithm graph that initially is assumed as the web service graph has attributed features, the combination of vertexes s' only is possible if $I(s_i) = I(s_j)$.

Restrictions must be applied to the combining of algorithm graph vertexes and the development of web services also to be able to process large algorithm graphs and find all the possible graph vertexes segments. If there are no restrictions and the algorithm graph is complex, segments have to be extracted in the e-service graph, and then every segment is treated as an individual design task. The division of graphs is related to computation tasks, where it is necessary to divide the graph in two (or more) large parts, minimising the number of edges that cross the split.

Obtaining all the possible algorithm graph segments is an NP–complete problem, where all the possible web service graphs only may be obtained by executing a recursive algorithm.

3.3 Pareto Optimality Set

There are several criteria that have to be taken into account when selecting an acceptable e-business system architecture, and very often they are contradictive. It means that it is necessary to find a web service graph in the set *X* (which has been formed after segmenting the algorithm graph) that is the best (or at least not worse) according to the specified quality criteria and on the basis of which an acceptable e-business system architecture can be build. It is very essential to deal with all the criteria simultaneously, as it is impossible to determine which of them is more important, and also, they are not mutually comparable. The task of finding the web service graph can be reduced to a multi-criteria optimisation task. Multi-criteria optimisation means finding the Pareto optimality in the sets, which can be formally defined as follows:

$$Q(X) \to \min_{X \in \Omega} \to P \tag{1}$$

$Q(X) = \{q_1(X), q_2(X), \cdots q_N(X)\}$ are criteria to be minimised; Ω – X definition area. The solution $X_i \in \Omega$ will be called a Pareto optimality if and only if $X_j \in \Omega$ does not exist such that

$$q_i(X_j) \leq q_i(X_i) \tag{2}$$

for all $i = \{1,2, \cdots N\}$, where at least one is a strong inequality. In other words, for any value $X_i \in P$ in the set Ω no better than the selected value can be found according to every criterion to be optimised. Depicting the Pareto optimality set graphically by two criteria to be optimised, they are all solutions that are nearest to the origin (Fig. 5.)

To be able to mutually compare and view simultaneously various criteria to be minimised, they must be normalised.

In the selection of the web service graph, the application of the Pareto optimality set is done as follows. An e-service algorithm graph is given. Initially it is assumed as the web service graph *G*.

The task is to find the web service graph set

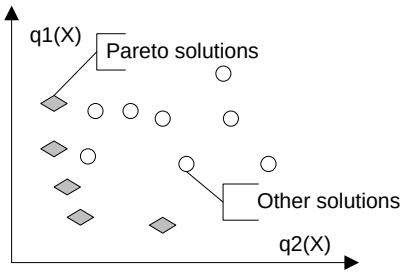

Figure 5: Pareto optimality set.

$P = \{G'^{*}\}$, which is defined in the area Ω (i.e. all possible algorithm graph segment combinations). The criteria Q, which are the quality criteria specified for the system, are minimised. To determine the system quality criteria, it is possible to use the following methods: scenario-based, simulation-based, mathematical model-based and experience-based.

4 METHOD ANALYSYS AND PRACTICAL USAGE

The application of the Quality Attributes Driven Web Services Design Method has been analysed by using a number of study examples; also, e-business system architecture designs obtained by using this design method and classical object-oriented or functionality-based design methods have been compared. E-services developed for Latvian government and local governments in the portals www.latvija.lv, www.riga.lv and www.epakalpojumi.lv have been used to analyse and approbate the method. The method analysis and the offered e-business system architecture are based on VISS, Latvian Government Information System Integrator, which is an environment for e-services and integration for government and local government services. There are similar e-business and integration systems also in other countries, for example, X-road in Estonia, PSB in Ireland, OIO in Denmark, GovTalk in the UK etc.

Analysing the results obtained from by using the method and comparing them with already implemented e-services, it can be seen that the design method can be used in practice and that it does not have any apparent drawbacks that exclude any essential solutions. The used design methods are different as to the resources that are required for building an acceptable system architecture. It was established that several e-services had been developed iteratively during several years, improving them constantly to achieve the desired quality, and that several service versions that include improvements in the system architecture had been released. To develop an e-service, a number of experienced system architects had been engaged, which means that resources had been considerably consumed. The offered method achieves the same result by engaging one system architect only and in a comparatively shorter period.

4.1 Design Methods Comparison

To design a complex e-business system, often it is not enough to use one design method, instead, several methods must be combined to meet the required quality of service (QoS). There are several methods that ensure that the required QoS level is achieved, like Attribute Driven Design (ADD), Software Architecture Analysis Method (SAAM), Jan Bosch architecture design method, Architecture Tradeoff Analysis Method (ATAM), Hazard Analysis of Software Architectural Designs (HASARD) etc.

Considering that in the result of using the Quality Attributes Driven Web Services Design Method a Pareto optimality set that contains several results is obtained, then in more complex cases when the system architect is unable to make a selection from the Pareto optimality set, any other design, decision-taking or optimisation method that helps select a solution from the Pareto optimality set can be applied. In this case, the possibility that a local minimum is selected is excluded.

For the system architect to be able to select an acceptable for them design method, a comparison of methods by the following assessment criteria is offered:

1. Multi-Criteria – in selecting the design method, it is important to see to that all the quality criteria specified for the system are dealt with simultaneously;
2. Is the subjective expert position excluded – in many design methods, the selection of solution depends on the system architect, whose subjective opinion influences the system to be designed;
3. Can be estimated in advance – since the development of a system is very time-consuming, it is essential that the quality characteristics can be assessed prior to commencing it;

Table 1: Comparison of architecture design methods.

Method / Comparison Nr.	1.	2.	3.	4.	5.	6.
Functional based	No	No	No	No	Yes	Expert; Methodology
Object-oriented	No	No	No	No	Yes	Expert; Methodology
ADD	Yes	No	Yes	No	Yes	Expert; Methodology
SAAM	Yes	No	Yes	No	Yes	Expert; Methodology
Jan Bosch	Yes	No	Yes	No	Yes	Formulas; Expert; Methodology
ATAM	Yes	Partly, there is more than one expert.	Yes	No	Yes	Experts; Methodology
HASARD	Partly, method is oriented to achieve security requirements.	No	Yes	No	Yes	Expert; Methodology
Quality Attributes Driven Web Services Design Method	Yes	Yes	Yes	Yes	No	Formulas
Combined Quality Attributes Driven Web Services Design Method	No	Partly. To select solution from Pareto compromise set there additional analysis is required.	Yes	Yes	Yes	Formulas; Experts; Methodology

4. Pareto Guaranteed – if the solution is selected from among all the solutions in the Pareto optimality set, it is sure that the selected solution is one of the best possible;
5. Exact Solution – it is important to make sure whether the design method provides one or more solutions;
6. Accuracy of the Solution: expert – the accuracy of the solution is determined by the system architecture's experience; methodology – the accuracy of the solution is determined by complying with the descriptive methodology; formula – the accuracy of the solution depends on the accuracy of the formulae used. As it can be seen from the given comparison (Table 1), there are methods that are multi-criteria and that make it possible to evaluate the quality characteristics of the system architecture prior to development; however, the accuracy of these methods relies only and solely on the expert's experience and the compliance with the methodology. It means that the possibility that a local minimum is selected is not excluded. Another drawback of these methods is the fact that the subjective position of the expert is not excluded, as in most cases only one system architect that takes the decision is engaged in system designing. Considering that successful advancement of computer systems relies upon their quality, any methods that are not multi-criteria often do not provide the desired result. The application of these methods can cause loss both to the client who ordered the computer system and the developer.

The existing multi-criteria design methods are acceptable in designing large and complex computer systems, which cannot be formally described and for which the mathematic multi-criteria optimisation methods cannot be applied. In such cases the ATAM method, which involves several system design experts, could be very efficient. Also other methods looked at herein could be useful in designing large computer systems if competent experts of the relevant sector and who can take decisions and evaluate the quality characteristics of the computer system to be designed are called in.

To design the e-service system architecture, a design method that provides fast and quality result regardless of the qualification of the involved experts is required. Since the e-service algorithm can be described formally in the form of an algorithm graph, it is possible to use the offered web service design method and the selection of an acceptable e-service system architecture that is based on multi-criteria optimisation. In such and similar cases, it is the multi-criteria optimisation method that should be used to prevent that a local minimum is selected.

Considering that in the result of multi-criteria optimisation a Pareto optimality set that contains several results is obtained, then in more complex cases, when the system architect is unable to make a

selection from the Pareto optimality set, any other design, decision-taking or optimisation method that helps select a solution that is contained in the Pareto optimality set can be applied. In this case, the possibility that a local minimum is selected is excluded.

5 CONCLUSIONS

To meet all the quality requirements set for the e-service, it is important to select an e-business architecture design method that is multi-criteria, as the quality criteria are mutually controversial; otherwise there is a possibility that, while improving one of the quality criteria, another potentially significant condition is not dealt with. If a multi-criteria design method is selected, a compromise among the quality criteria of the e-service is found. A drawback of other design methods is the subjective position of the designer that affects the quality metrics of the designed system and might lead to selecting a local minimum, thus leaving out the best possible solution.

The Quality Attributes Driven Web Services Design Method ensures swift and quality designing of web services, which is the basis for the e-business system architecture and affects the QoS of the e-service. By applying this method, it is possible to optimise the system design costs.

The offered Quality Attributes Driven Web Services Design Method can be used also in other systems that are not complex algorithmically (i.e. algorithm graph can be built) and are based on SOA and web services.

ACKNOWLEDGEMENTS

This work has been supported by the European Social Fund within the project „Support for the implementation of doctoral studies at Riga Technical University".

REFERENCES

Bass L., Clements P., Kazman R. *Software Architecture in Practice 2nd ed*. - Boston: Addison-Wesley , 2003.

Bass L., Klein M., Bachmann F. *Quality Attribute Design Primitives and the Attribute Driven Design Method*// Lecture Notes In Computer Science. Revised Papers from the 4th International Workshop on Software Product-Family Engineering. - London: Springerferlag, 2001. - Nr. 2290 – p.169.-186.

Bosch J. *Design and Use of Software Architectures*. - Harlow: Addison-Wesley , 2000.

Hofmeister C., Nord R., Dilip S. *Applied Software Architecture*. - Massachusetts: Addison-Wesley , 2000.

Kan S. *Metrics and Models in Software Quality* Engineering 2nd ed. - Boston: Addison-Wesley, 2003.

Kazman R., Bass L., Webb M., Abowd G. *SAAM: A Method for Analyzing the Properties of Software Architectures*, International Conference on Software Engineering. Proceedings of the 16th international conference on Software engineering. - Los Alamitos: IEEE Computer Society Press, 1994. –p. 81.-90.

Kazman R., Klein M., Clements P. *ATAM: Method for Architecture Evaluation*. - Pittsburgh: Carnegie Mellon Software Engineering Institute CMU/SEI-2000-TR-004 ESC-TR-2000-004 , 2000.

Losavio F. *Quality Models to Design Software Architecture*, Journal of Object Technology. - 2002. - Nr. Vol. 1, No. 4

Miettinen K. Nonlinear *Multiobjective Optimization*. - Boston: Kluwer Academic Publishers , 1998.

Triantaphyllou E. *Multi-Criteria Decision Making Methods: A Comparative Study*. - Dordrecht: Kluwer academic publishers , 2000.

Zeiris E., Ziema M. *Criterions for Architecture Estimation and Architecture Selection of E – Services System*// Scientific Proceedings of Riga Technical University. Computer Science. Part 5. Volume 27. - Riga: RTU, 2006. – p. 91.-98.

Zeiris E., Ziema M. E-Service *Architecture Selection Based on Multi-criteria Optimization*// Product-Focused Software Process Improvement 8th International Conference, PROFES 2007, Riga, Latvia, July 2007 Proceedings. - Berlin Heidelberg: Springer-Verlag, 2007. - 345.-35

Zeiris E., Ziema M. *Selection of E – Service Systems Architecture*, Scientific Proceedings of Riga Technical University. Computer Science. Part 5. Volume 32. - Riga: RTU, 2007. – p. 99.-107.

Zhu H. Software *Design Methodology From Principles to Architectural Styles*. - Amsterdam: Elsevier Butterworth Heinemann , 2005.