IMPROVEMENT OF APPLICATIONS DEVELOPMENT USING SERVICE ORIENTED ARCHITECTURE

Pavel Semenchuk

Riga Technical University, Applied Computer Science Department,
Meza str. 1/3, Riga, Latvia, Pavels.Semencuks@rtu.lv

Abstract. Business applications play important role in understanding and usage of newest technologies. That's why any tendencies in their evolution attract high attention. Distributed applications are popular today. Service Oriented Architecture (SOA) is one of modern and progressive ways to develop such systems. But it is known fact: starting projects of Service Oriented Architecture implementation a lot of organizations discover that this is much more difficult task than it seemed in the beginning. There are many causes which make SOA implementation more difficult process as it could be. There is no worked out common methodology for projects' transition to SOA. It is important to implement Service Oriented Architecture gradually. This paper gives advices of transition to SOA and promotes the idea to correct some popular errors in system's development. In future these ways can be used as a part of unified SOA methodology. Those advices are worked out based on author's experience in SOA implementation.

Keywords: BPEL, methodology, SOA, SOAP, WSDL.

1 Introduction

At the modern stage of the market evolution a lot of organizations already understand what SOA is and why it is necessary. But there is no common opinion for such SOA implementation, which would give benefit for the business and ensure successful functioning of SOA for many years forward. Software developers have their own (private) methodologies [1, 2, 28], which were specially worked out for the projects with SOA. These methodologies are very common and not standardized. Such situation needs deeper research of already existing methodologies for SOA and creation of unified SOA methodology.

Some companies discover that there are a certain number of organizational, technological and architectural problems which leads to the stagnation of projects in which would be possible to apply SOA. In many other companies projects of SOA implementation in real information systems do not go out of the pilot stage.

This paper is organized as follows. Section 2 describes the basics of SOA. Section 3 describes advantages of SOA, which were defined analyzing SOA architecture. Section 4 describes advices for SOA implementation in real projects. Section 5 describes the main obstacles for SOA implementation. The advices and obstacles are author's private opinion. The common criteria for SOA based system development are described in section 6. Section 7 describes software development models. Section 8 describes current situation with SOA methodologies. Section 9 promotes the idea of new SOA methodology. Conclusions state further directions of research.

2 SOA basics

Service Oriented Architecture – architecture of information systems which is developed and based on week-binding of system parts principle. The smallest parts of such kind of systems are services. SOA is fully independent from programming languages, protocol specifications and platforms. The main reason of SOA appearance on the market is the replacement of coding of programs (from the beginning till the end) to assembly of applications from the standard parts.

SOA includes the following standards which gives opportunity to offer service oriented solutions:

- *WSDL (Web Service Description Language)* – formal standard for the description of Web services which is based on XML (*eXtended Markup Language*) language. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating details involved in applications communication [27].
- *UDDI (Universal Description Discovery and Integration)* – an XML-based registry for businesses worldwide to list themselves on the Internet. Its ultimate goal is to streamline online transactions by enabling companies to find one another on the Web and make their systems interoperable for e-commerce [26].
- *SOAP (Simple Object Access Protocol)* – protocol of exchange of structural information in the distributed systems. The exchange is based on XML technology. SOAP is a way for program running in one kind of operating system (such as Windows 2000) to communicate with a program

in the same or another kind of an operating system (such as Linux) by using the World Wide Web's Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange [20].

- *BPEL (Business Process Execution Language)* – language of description of business processes. The description is based on XML. BPEL for Web services is an XML-based language designed to enable task-sharing for a distributed computing or grid computing environment – even across multiple organizations – using a combination of Web services [5].

Many SOA definitions appeared during the last couple years, but all of them describe two basic moments [13]:

- *SOA* – week-binding architecture of the systems.
- *SOA* – set of the standards which give opportunity to realize SOA principles.

From the business point of view the Web service can be expressed like a realization of business process or part of this process. SOA is the relating element of all business processes for the execution of main function.

3 Advantages of SOA integration

When using SOA, organization gets a wide area of opportunities. These opportunities are defined by the characteristics of Service Oriented Architecture. The main opportunity is the business respond speed on the changes of the environment and the evolution of business processes with minimal expenses (for example, in comparison with CORBA).

The analysis of SOA advantages and characteristics is based on author's experience [17, 18] and analyzing literature [13]. The main characteristics of Service Oriented Architecture are:

- *Week-binding.* Week-binding provides simple and quick adaptation of system to the changes in the structure and realization of the services. It gives opportunity on-the-fly reorganize business processes accordingly to the changes of the market (due to the week bindings between the services). Week-binding increases competitiveness because of the full confrontation of service and business process. It decreases price of implementation due to the high level of reusing of the services.
- *Modularity.* Module principle of building gives opportunity to organize graft on development, implementation and maintenance. This principle also gives opportunity for the gradual replacement of applications and hardware.
- *"SOA is not aggressive".* SOA gives opportunity to use all investments, which were earlier invested in the information technologies, allows avoiding the reorganizing and retesting of existing systems when including them in SOA.
- *Standardized.* Platform-independence of SOA gives opportunity to choose any software and hardware. Standardization allows avoiding software lock-in, decreases complicity and fragmentation of the result from the using of closed products.
- *Technological**.*** One and the same technology can be applied for the wide spectrum of business problems.

Oriented Architecture – architecture of information systems which is developed and based on week-binding of system parts principle. The smallest parts of such kind of systems are services. SOA is fully independent from programming languages, protocol specifications and platforms. The main reason of SOA appearance on the market is the replacement of coding of programs (from the beginning till the end) to assembly of applications from the standard parts.

4 Transition to SOA and possible results

Analyzing the ideology, which is included in SOA, author can pick out two stages which are necessary for the successful transition to SOA [17, 18]:

- Creation of Web services.
- Building SOA infrastructure.

Creation of Web services can be achieved:

- Web service can be written from the beginning.
- Transformation of existing applications to the service oriented applications.
- Supplementation of existing applications with shell, which gives opportunity to work with this application like with a Web service.

- Web service can be found between already created Web services (e.g. UDDI).
- Web service can be ordered (Web services providers).

Infrastructure of SOA is an instrument of management and security both for the system at all and for each separate service which is attached to the infrastructure. Interaction of systems and applications without an infrastructure is shown on Figure 1; with infrastructure – on Figure 2.

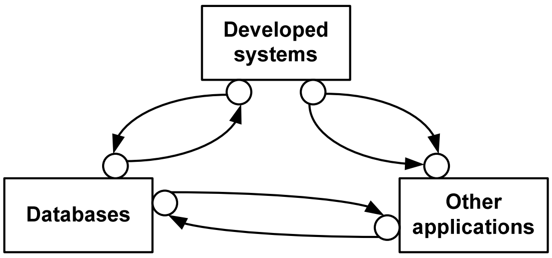

Figure 1. Interaction of the systems and the applications without the infrastructure

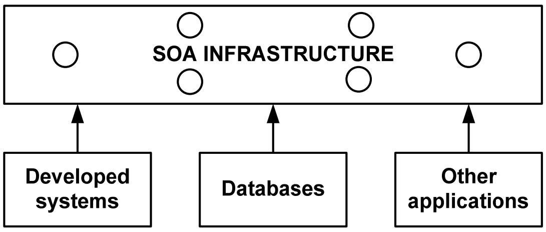

Figure 2. Interaction of the systems and the applications with the infrastructure

In the Figures 1 and 2 cycles means the integration points of applications. Through these points interaction between systems is supported.

In the Figure 1 the interaction of the systems without SOA infrastructure is shown. Such approach needs the usage of specific relations for each interaction. To relate N systems each with each other N*(N-1) specific one-sided relations are needed. Any change in such systems is very difficult to make, in many cases it is not possible at all.

In the Figure 2 the interaction of systems with SOA infrastructure is shown. All interaction points are in the one interaction scope. Because these points don't need any relations between the systems; they work independently. Using such approach in the worst case we need to organize not more relations than in infrastructure without SOA. Even in this case these relations qualitatively differ from relations without SOA. Relations with SOA describe interaction with abstract system. This gives an opportunity to reorganize interaction of systems without any problems. Infrastructure allows creating systems which can be complicated, scalable and ready for continuous changes. It also represents the common point of management and maintenance of all system's parts.

5 SOA problems

Nothing is ideal. SOA is not ideal too. There are some big obstacles, which trouble successful evolution of projects with SOA [14]:

- *Full functional compatibility is not achieved.* SOA is based on standards of functional compatibility. Functional compatibility of different software products and platforms forms meaning of Web services. There are a lot of organizations which work on standards for interaction of Web services. But still, even with basic standards such as SOAP and WSDL, there are problems. In different cases these basic standards can be used a little bit different ways. To solve such kind of problems organizations should force their providers to make products which are compatible at least with basic standards of Web services.
- *There are not enough SOA specialists.* There are many kinds of architectures for information systems development. Most of the system architects usually work with one or two architectures. To be a real specialist in SOA this is not enough. To work with SOA, architect should understand corporative architecture and have good knowledge of technical, informational, business processes and data architectures. Such qualified architects are rarity.

- *Standards are not full, limited or badly worked out.* The real problem is that standards are limited or badly developed. For example, WSDL is limited standard; WS-BPEL is badly developed standard. It is not able to provide requirements to service or contract limitations for user of service. WS-BPEL includes WSDL and UDDI standards, which are badly combined one with another.
- *Software products are limited.* SOA creation can't be achieved only with one software product. During SOA implementation organizations need to solve different tasks and problems with different software products. As the result organizations need to use different software products from different providers. But these providers still try to achieve functional compatibility which was described above.

6 Criteria for SOA development

For SOA development is necessary to define criteria. In author's opinion some already existing methodologies (RUP, XP) can be taken as a criterion for SOA methodology. The criteria for SOA development are presented below [28]:

- *Delivery strategy.* There exist three common strategies in delivering a SOA, depending on the amount of front-end analysis of the business domain and the treatment of existing legacy systems [25]. The top-down strategy is closely tied to an organizations' existing business logic from which required services are derived. The bottom-up strategy is the opposite in that it focuses on legacy systems, and Web services are built on an as-needed basis. The meet-in-the middle (agile) strategy finds a balance between incorporating service-oriented design principles into business analysis environments without having to wait before integrating Web services technologies into technical environments [25].
- *Lifecycle coverage.* Some proposed approaches aim to support the full SOA lifecycle, including planning, analysis and design, construction, testing, deployment, and governance activities while others limit their scope to a subset of these phases, such as analysis and design.
- *Degree of prescription.* SOA methodologies range from the most prescriptive ones that specify phases, disciplines, tasks, and deliverables for each of them, while others provide less detail, by purpose or not, leaving room for more flexibility and tailoring of the approach depending on the project's context.
- *Availability.* A number of methodologies proposed by industry players such as IBM, Sun, Microsoft, and others, are proprietary and the detailed specifications are not openly available. In contrast to open methodologies for which documentation is available to the interested public, for the proprietary methodologies it is difficult to fully analyze their capabilities and to make comparisons.
- *Process agility.* A number of methodologies suggest an agile approach to Service Oriented development in order to address risks and add flexibility to change. Yet, some others follow a more rigid approach in the process lifecycle, or do not address the issue of agility at all.
- *Adoption of existing processes/techniques/notation.* A large number of SOA methodologies propose reusing proven existing processes like XP and RUP, A Survey of Service Oriented Development Methodologies 3 and techniques like OOAD, CBD, and BPM, seeing service-oriented development as an evolutionary rather than revolutionary step in software engineering. Also standardized notations, such as UML and BPMN, are being adopted to visually model various artifacts.
- *Industrial application.* It is important that a methodology be validated in proof-of-concept case studies to show that it has practical applicability and to refine it based on feedback from case studies. Unfortunately, most of the existing SOA methodologies are at an early stage and have not been applied in industrial projects yet.
- *Supported role(s).* A service-oriented methodology may support the provider view, the consumer view, or both the provider and consumer views in an integrated framework. In the consumer's view, development is declarative and business process oriented through service composition, while in the provider's view it is programmatic and component oriented.

7 Existing SOA methodologies

Methodology – the science that studies the methods of problem solving [9]. This is the set of methods, principles, facilities for the achievement of the goal. The methodology can be divided into two parts: theoretical and practical.

Modern SOA methodology includes two main processes [21]:

- *Service-Oriented Analysis Process.* A separate analysis is dedicated to each business process definition associated with a given service inventory. For a full definition of a service inventory blueprint, a complete top-down delivery process is carried out, comprised of numerous iterations through service-oriented analysis process steps.
- *Service-Oriented Design Process.* All of the effort put into the analysis and service modeling processes results in a collection of service candidates that establishes the starting point for service design. Every candidate definition can be used as input for a service-oriented design process. A different process exists for each of the four primary service models, but all are shaped and structured around the application of service-orientation design principles.

There is one central aspect in a SOA that drives agile development: The service model – the model of services, their dependencies, choreography, and flows [19]. There are three common service classifications that represent the primary service models [21]:

- *Entity service.* The entity service model represents a business-centric service that bases its functional boundary and context on one or more related business entities. It is considered a highly reusable service because it is agnostic to most parent business processes.
- *Task service.* A business service with a functional boundary directly associated with a specific parent business task or process.
- *Utility service.* It is dedicated to providing reusable, cross-cutting utility functionality, such as event logging, notification, and exception handling.

Some of the existing SOA methodologies are shown below:

- IBM Service-Oriented Analysis and Design (SOAD) [12];
- IBM Service Oriented Modeling and Architecture (SOMA) [3];
- SOA Repeatable Quality (RQ) [23];
- CBDI-SAE Process [15];
- Service Oriented Architecture Framework (SOAF) [4];
- Service Oriented Unified Process (SOUP) [8];
- Methodology by [11];
- Thomas Erl's [25];
- BPMN to BPEL [6];
- Steve Jones' Service Architectures [24].

All these methodologies doesn't support all criteria which were defined (section 6) for successful SOA implementation. Such situation needs more common methodology for SOA implementation in IT systems.

8 Software development models

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. The general software development model consists of 4 phases:

- *Requirements.* Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.
- *Design.* The software system design is produced from the results of the requirements phase. This is where the details on how the system will work are produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.
- *Implementation.* Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

- *Testing.* During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

The general software development model is shown on Figure 3.

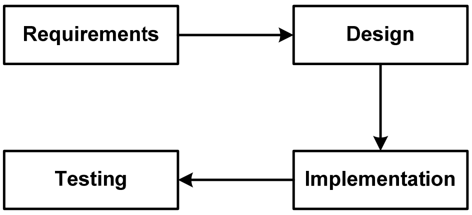

Figure 3. Software development general model

The traditional and widely used software development models are:

- Waterfall life cycle model;
- V-shaped life cycle model;
- Incremental life cycle model;
- Spiral life cycle model.

9 SOA methodology

At this moment technological aspects are worked over much better than methodological questions of projects realization with integration of SOA. Software developers have their own (private) methodologies (mainly based on RUP [16] and XP [7]), which were specially worked out for the projects with SOA. But still there is no unified SOA methodology. This disturbs the realization of unified integration system and stops many architects from transition to SOA. Methodology described above is very common. It is based on agile methods, but there isn't any advises about exact methods for exact tasks. It is very difficult to decide which method to use in exact situation.

There are already some ideas for SOA life cycle (on Figure 4) and Web service life cycle (on Figure 5) [29]. Author of this paper consider that these parts can be taken into account for the unified SOA methodology creation.

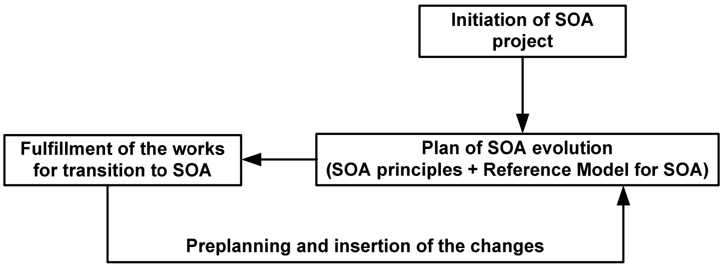
Initiation of SOA project
Fulfillment of the works for transition to SOA
Plan of SOA evolution (SOA principles + Reference Model for SOA)
Preplanning and insertion of the changes

Figure 4. SOA life cycle

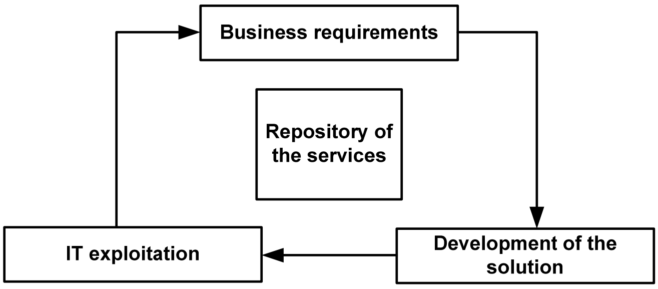

Figure 5. Web service life cycle

As shown in Figure 3 SOA integration is iterative process. It means that the spiral model of development process can be applied as a base for SOA model of development process. SOA model of development process can be applied as a base for SOA methodology (Figure 6). All these need a deep research. This research is planned as the future work on doctoral work.

The idea of SOA methodology is based on SOA model of development process and other new methods, which are not defined yet and will be included after a deeper research of already existing standards of SOA. SOA model of development process will be based on best practices of already existing models of development process (etc. spiral, waterfall and others) and combined with new original ideas, which are not worked out yet and is a part of the future research. One of the new methods, which is planned to work out for SOA methodology, will be SOA test method (the starting point of this idea can be found in corresponding works [22]). Testing is an important part of any kind of projects. Errors in the big projects cost too much (SOA projects included).

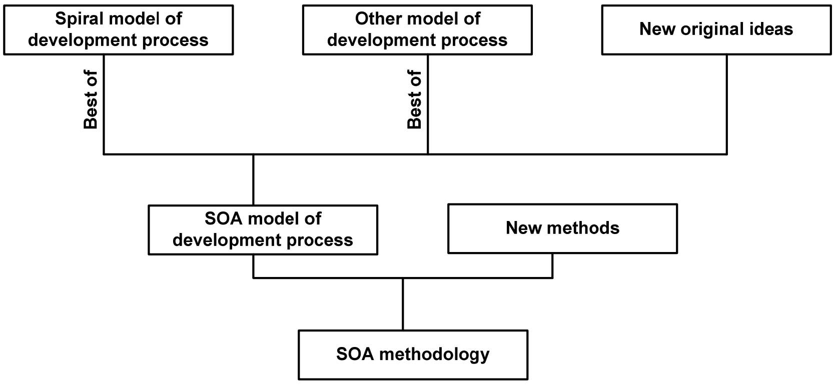
Spiral model of development process
Other model of development process
New original ideas
Best of
Best of
SOA model of development process
New methods
SOA methodology

Figure 6. Idea of the SOA methodology

There are already some positive examples of new model creation which are based on previously developed models. For example, Microsoft Solutions Framework model [10] which includes best practices of spiral and waterfall models. In author's opinion SOA needs unified methodology to make easier transition to SOA.

One of the main criteria for SOA development is *life cycle coverage* (Section 6). There is a lot of existing software development models (some of them are mentioned in section 8) which can be applied implementing SOA. Another one of the main criteria for SOA development is the *degree of prescription* (Section 6). That's why such models as waterfall, incremental or spiral model could be applied for SOA implementation; they are enough detailed and have enough phases. As the SOA implementation process should be iterative and flexible for changes the waterfall and incremental software developments models can't be fully applied for SOA projects. Waterfall model is not iterative and flexible. In the incremental model each phase of an iteration is rigid and do not overlap each other. In such direction spiral software development model is a good solution as it fully iterative and flexible.

10 Conclusions

There are a lot of obstacles (some of them are discussed above in this paper) for the successful SOA implementation. All of them are overcome if to act in series. The positive result of SOA implementation can give the following advantages:

- Quicker and moreflexible changes of business processes.
- Decrease of expenses for the IT operations.
- Quick implementation of updates and additional opportunities of software products.

The integration is a successful relation between data, applications, processes, people and organizations. From this point of view integration is a process, not technology. That is why while implementing SOA, it is important to think about the investment which the architecture will invest in improvement of business process from which depend existence and evolution of the organization.

From the methodological point of view SOA needs an iterative approach. This is because the requirements are constantly changing. At this moment technological aspects are worked over much better than methodological questions of project realization with integration of SOA. Software developers have their own (private) methodologies, which were specially worked out for the projects with SOA. In author's opinion, there are not enough good specialists in SOA and there is no unified methodology for SOA implementation. These disturb the realization of unified integration system. In author's doctoral work it is planned to offer (work out) the new methodology for the development of distributed information systems which will be based on SOA principles. One of the main aspects of this methodology will be SOA testing.

References

[1] A Methodology for Service Architectures. Access via the Internet: http://www.oasis-open.org/committees/download.php/15071/A%20methodology%20for%20Service%20Architectures%201%202%204%20-%20OASIS%20Contribution.pdf

[2] A Survey of Service Oriented Development Methodologies. Access via the Internet: http://www.dcs.shef.ac.uk/~ajhs/research/papers/soasurvey.pdf

[3] **A. Arsanjani**. Service-oriented modeling and architecture. IBM Corporation, November 2004. Access via the Internet: http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/

[4] **A. Erradi et al.** SOAF: An architectural framework for service definition and realization. *In Proceedings of the IEEE International Conference on Services Computing*, pp. 151-158, Chicago, USA, September 2006.

[5] BPEL. Access via the Internet: http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci845110,00.html

[6] **C. Emig et al.** Development of SOA-based software systems - and evolutionary programming approach *In Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services,* pp. 182, Guadeloupe, French Caribbean, February 2006.

[7] Extreme programming. Access via the Internet: http://www.xprogramming.com/xpmag/whatisxp.htm

[8] **K. Mittal.** Service Oriented Unified Process (SOUP). Access via the Internet: http://www.kunalmittal.com/html/soup.shtml, 2006.

[9] Methodology. Access via the Internet: http://www.knowledgerush.com/kr/encyclopedia/Methodology/

[10] Microsoft Solutions Framework. Access via the Internet: http://www.echoes.com/msf/

[11] **M. P. Papazoglou and W. J. van den Heuvel.** Service-oriented design and development methodology. *International Journal of Web Engineering and Technology (IJWET)*, 2006.

[12] **O. Zimmermann et al.** Elements of Service-Oriented Analysis and Design. *IBM Corporation*, June 2004. Access via the Internet: http://www-128.ibm.com/developerworks/library/wssoad1/

[13] Perehod predprijatij na servis-orientirovannuju arhitekturu (in Russian: Переход предприятий на сервис-ориентированную архитектуру) Access via the Internet: http://tm.ifmo.ru/tm2006/src/143as.pdf

[14] Pochemu vnedrenie servis-orientirovannoj arhitekturi trebuet mnogo vremeni (in Russian: Почему внедрение сервис-ориентированной архитектуры требует много времени). Access via the Internet: http://citcity.ru/11420/

[15] **P. Allen.** The service oriented process. *In CBDi Journal*, February 2007. Access via the Internet:
http://www.cbdiforum.com/report summary.php3?page=/secure/interact/2007-02/service oriented process.php&area=silver

[16] Rational Unified Process. Access via the Internet:
http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

[17] **Semenchuk P., Kornienko Y., Narnicka V.** Modern Government E-service Infrastructure: Latvian Approach. *Databases and information systems*, Eighth International Baltic Conference on Databases and Information Systems, pp. 267-278, Estonia, Tallinn, 2.-5. June, 2008.

[18] **Semenchuk P., Kornienko Y., Narnicka V.** Service Oriented Architecture Application for Latvian Public Sector E-service Development. *Databases and information systems*, pp. 267.-278, IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, Netherlands (2009).

[19] Service-oriented agility: Methods for successful Service-Oriented Architecture (SOA) development. Access via the Internet: http://www.ibm.com/developerworks/webservices/library/ws-agile1/#ibm-pcon

[20] SOAP. Access via the Internet: http://www.bitpipe.com/tlist/SOAP.html

[21] SOA methodology. Access via the Internet: http://www.soamethodology.com/

[22] SOA Test Methodology. Access via the Internet: http://www.thbs.com/pdfs/SOA_Test_Methodology.pdf

[23] SUN Microsystems. “SOA RQ methodology - A pragmatic approach”. Access via the Internet: http://www.sun.com/products/soa/soa methodology.pdf

[24] **S. Jones.** A Methodology for Service Architectures. Capgemini UK plc. Access via the Internet: http://www.oasis-open.org/committees/download.php/15071/A%20methodology%20for%20Service%20Architectures%201%202%204%20-%20OASIS%20Contribution.pdf, August 2005.

[25] **T. Erl.** Service-Oriented Architecture: Concepts, Technology, and Design, Upper Saddle River: Prentice Hall PTR, 2005.

[26] UDDI. Access via the Internet: http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci508228,00.html

[27] Web Services Description Language (WSDL) 1.1. Access via the Internet: http://www.w3.org/TR/wsdl

[28] Web Services Implementation Methodology for SOA Application. Access via the Internet: http://www.fwsi.org/resources/INDIN06-WS%20Methodology-Final.pdf

[29] Zhiznennij cikl servisov (in Russian: Жизненный цикл сервисов). Access via the Internet: http://www.servicemodels.ru/