

TOWARDS QUALITY DRIVEN WEB SERVICE COMPOSITION

Georgiana STEGARU¹, Aurelian Mihai STĂNESCU²

Service composition allows organizations to automate complex cross-organizational business processes. In order to assure the success of a service-oriented business process not only the services must be of a great quality but also the processes of creating the services. While state of the art approaches focuses on service selection algorithms with desired QoS, this work proposes a methodological modelling framework for quality driven Web Service Composition that tackles quality from the user, final product and process views. A case study within E4H domain for Indoor Air Quality Management is presented to demonstrate the application of the proposed modelling framework.

Keywords: QoS, Service Composition, E4H (Environment for Health)

1. Introduction

The dynamic of the Information Communication and Technology (ICT) field is bringing radical changes and new opportunities in different aspects of our lives: economic, social and political. The European Commission envisions for 2020 a "Digital Society" for the competitive and sustainable innovation based European economy, where Future Internet represents a catalyst for the formation of new business ecosystems beyond the key enabling technologies [1].

One area of research for FInES is Internet of Services where everything (e.g. information, software, platform and infrastructure) is available as a service. SaaS has become increasingly significant in the past few years due to the popularity growth of Service Oriented Architecture (SOA). With concepts inherited from software component development, SOA provides a framework for development of services based on principles like flexibility, interoperability, adaptability, reusability, granularity and governability [2].

Along with SOA development the rapid extend of services to the www, Web services, led to a major interest in the possibility of composing them to provide new value-added services. Service composition typically involves the

¹ PhD Student, Depart. Automatics and Industrial Informatics, University POLITEHNICA of Bucharest, Romania, e-mail: Georgiana_stegaru@yahoo.com

² Prof., Depart. Automatics and Industrial Informatics, University POLITEHNICA of Bucharest, Romania, e-mail: amstanescu@yahoo.com

composition of atomic services or composite services into a new service that fulfils some requirements. Nowadays service composition automates a business process allowing organizations to develop new services and products or to customize existing ones to better suit customer's needs (fig. 1).

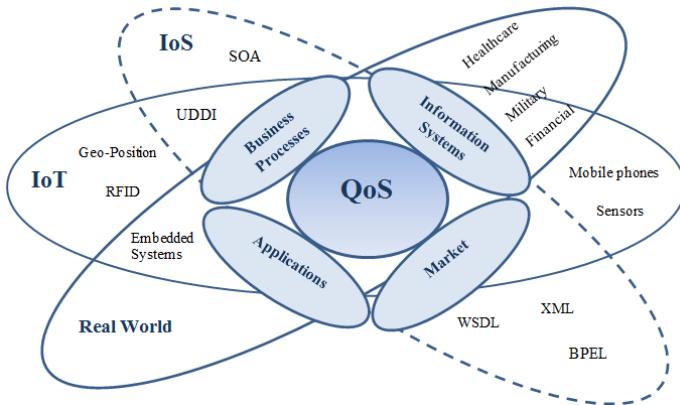


Fig. 1. Service Composition space

Service discovery and selection is an important phase in the service composition process. A large number of QoS driven service composition approaches focus on different selection methods for the best quality services. Service selection is formulated as an optimization problem and a linear method is used to compute optimal service execution plans for composite services [3]. Another solution to the service selection is suggested by using global optimizations techniques and local selection methods to find the best combination of services so that their aggregated resulting into a QoS-aware service composition for web services [4]. Using a popular algorithm from Artificial Intelligence literature "beam stack search" the QoS based selection of web services for a composition is computed earlier than when using traditional algorithms [5]. In [6] the authors present a service query framework that enables users to easily access services with desired QoS using a cost model based on QoS to select best execution service composition plan.

A number of works focus on QoS management using different methods to monitor and control QoS properties of services involved in the composition: policies [7], [8], contracts [9], agents [10], [11]. Inspired by nature's and artificial immune system (AIS), the authors of [12] have used intelligent algorithm of particle swarm optimization (PSO) and applied it to service composition problem. In [13] the authors propose a relaxable QoS-based service selection algorithm (RQSS), the term of "relaxable" meaning that the degree of constraints will be heuristically released or relaxed. The main idea is to find a solution with smaller

amount of constraints violation if there is no a feasible solution to fulfil the QoS constraints.

Current approaches reviewed from the literature concerning quality of service composition translate the quality concept in a selection and optimization problem that must be solved. Most of the related work focuses on the selection of the non-functional properties of different services in order to obtain the desired QoS of the composite service. The quality is concentrated in the final product (e.g. composite service) and less or not at all on the service composition process. There is the need for a more general approach to service composition quality and not only from the point of view of the output but also of the process from which is developed. A reliable service composition also requires service interoperability due to the fact that (i) there are many services developed on different platforms (e.g. .NET or Java) by different vendors which do not use the same data types, formats or schemas; (ii) standards are implemented differently, some developers extend the service with new functionalities which are not covered by the standard. Adaptability must be also addressed in service compositions [14] due to (i) the gap between non-interoperable services must be overcome or mediated when possible, (ii) optimization of QoS and business process operations executed by services is a continuous process, (iii) especially in distributed systems services invoked in the composition might fail due to network problems, thus recovery or replacement of faulty service is essential.

The objective of our work is to provide a methodological modelling framework for Web Service composition focusing on quality of each Web Service involved in the composition and of the service composition process.

2. Quality Management Modelling Framework

Quality of service composition is a result of product, process and people characteristics and inputs. The quality of a service composition depends on the quality of each service involved in the composition, but also on the service composition process.

The proposed modeling framework addresses quality from both product and process view. The perception of quality is different for each person. Therefore User Quality Profiles are used to capture user requirements, both functional and non-functional. This way the gap between expected quality by the user and the quality offered by the Service Provider is considerably reduced.

The proposed conceptual modeling framework shown in figure 2 comprises of:

- Quality for Service Composition (QoSC) Model, which models the quality of the composed service from business, operational and system views

- Quality of Service Composition Process Reference Model, which describes a general service composition process enhanced with verification and validation at each phase.

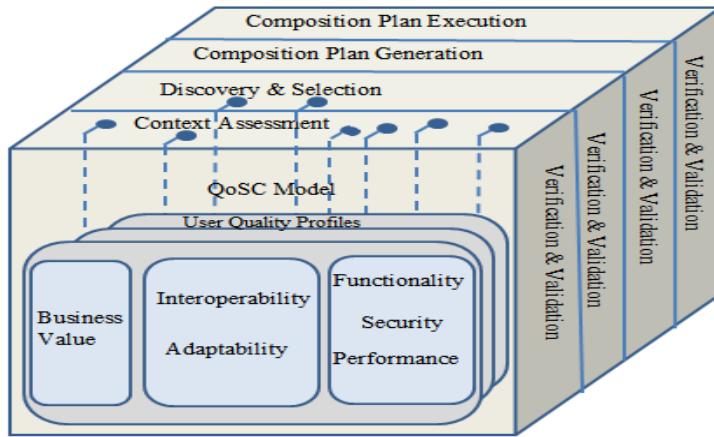


Fig.2. Quality Modeling Framework for Service Composition

2.1. Quality of Service Composition Model

The QoS model extended from [15] has three quality dimensions which assess the service composition from a certain point of view: business, operational or systemic. Each quality dimension is composed of one or more quality aspects. For each aspect we consider one or more quality items, which are tightly coupled with service desired characteristics and can be measured based on quality criteria. The measurement can be either quantitative or qualitative.

Business quality dimension models the business value aspect of a service composition. Business value concept captures aspects that are beyond economic value, usually viewed as monetary worth (profit, return on investment – ROI), including other forms of value such as customer value, managerial value, societal value. Business value refers to intangible assets that cannot be measured directly in monetary terms. Examples include intellectual property or organization's business model. Business quality dimension contains the following quality attributes: reputation (service provider reputation and service reputation), affordability, discoverability, penalty and incentive, usability and governance. These attributes can be qualitatively evaluated based on a history of the service usage experience and user feedback.

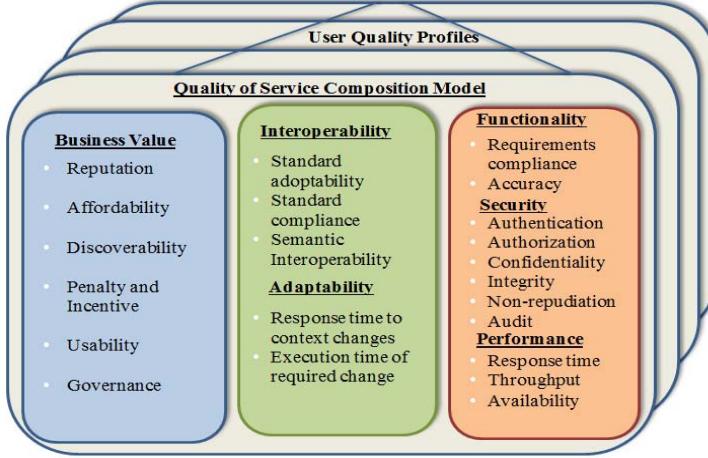


Fig. 3. Quality of Service Composition Model

Operational quality dimension assesses interactions between services involved in the service composition. Interactions between services are usually defined through operations that can be executed or invoked by a service. Operations execution involves message exchange between services. Thus compatibility at message level is needed to ensure that services are able to receive and interpret correctly messages that are passed from one another in a service composition. Operational quality dimension includes interoperability and adaptability quality aspects. Interoperability quality aspect is concerned with the evaluation of the compatibility level between services involved in the composition. Although services technologies have been standardized, services are still developed on different platforms and according to different specifications. Adaptability quality aspect evaluates the ability of a service composition to respond in real time to external stimuli such as changes in requirements, service failure or malfunctioning.

System quality dimension evaluates the service compliance with user defined requirements both functional and non-functional. At this level we assess the functionality, security and performance of the composed service. Although service's functionality is specified in the WSDL description it is possible that the service behaves improperly in certain situations. Atomic or composed services should be tested to check if they provide the functionalities requested by the service consumer. Besides functional requirements, security and performance are also important factors for the quality of composed services. Security in the context of service composition has the following quality items: authentication, authorization, confidentiality, non-repudiation, audit and integrity. In the literature there are many performance indicators [16], but we include in our model only

response time, throughput and availability which can be calculated from services monitoring during run-time.

2.2. Quality of service Composition Process Model

Based on the analysis on different service composition methods presented in [17], we identified the main phases of a service composition process. In figure 3 we present the model of a generic service composition process, enhanced with verification and validation that needs to be performed at each stage of the composition process to achieve a reliable service composition.

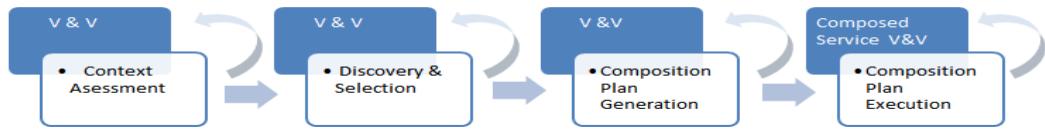


Fig. 4. Quality driven Service Composition Process (QoSCP)

Step 1: Context Assessment is triggered by a service request realized by a user. First requirements are collected from the service request initiated by a user. Context assessment goal is to obtain quality criteria for service discovery and selection by analyzing user requirements, user context, service context and business rules. Firstly, the service composition stakeholders (e.g. organization, end-user) are identified to generate the User Quality Profiles. User requirements can be explicit, stated by the user, and implicit, derived from the context. At this phase we must also consider user context (e.g. location, request scope and purpose of usage), service context (e.g. provider, availability, location, description, SLA) and business rules specific to the automated business process. Then the user's requirements and preferences are mapped to QoS properties specified in the proposed QoSC model. User Quality Profiles will contain quality aspects and quality items at the business, operational and system dimensions based on the QoSC model. User Quality Profiles are merged into the Service Selection Specification which includes the quality criteria for service selection. The quality criteria is specified through QoS metrics, both qualitative and quantitative.

Step 2: Context assessment verification and validation objectives are to check that user and service context are correctly assessed and requirements both functional and non-functional are correctly mapped to the QoSC model. Verification and validation of context assessment is very important because there might be conflicts between the User Quality Profiles, i.e. quality requirements that conflict. This can be identified when merging Organizations Quality Profile and Employees Quality Profile.

Step 3: Service discovery and selection processes are based on the QoS criteria identified at the previous phase. Service registries are queried to identify

services with required functionalities. To select the best matching service accordingly to QoS specification, each candidate service should be verified to ensure that it matches both functional and non-functional requirements (e.g. performance, security). The candidate services should be kept into a sorted list according to the probability for their selection for backup purposes.

Step 4: Service discovery and selection Verification and Validation is required to prevent possible issues at the execution phase. Interoperability and adaptability of services involved in the composition is assessed at this step. The proposed QoSC model captures these requirements in the operational quality dimension. It is essential that selected services are verified and validated before continuing to the next phase, since information found in services registries might not be up to date or accurate. In this case the service discovery and selection processes are reinitiated.

Step 5: Generation of service composition plan is done after having identified the services that match the requirements, both explicit and implicit, as captured in the first phase. The service composition plan represents the specification for the execution engine. The service composition plan depends on the composition method, service orchestration or service choreography [18].

Step 6: Service composition plan Verification and Validation is the next step. WS-BPEL service composition is workflow based language that can be verified with model checking techniques from the current literature as in software engineering.

Step 7: Execution of the service composition specification. Finally the builder executes the service composition specification and produces an implementation corresponding to the required composite service. Depending on the service composition language, the WS-CDL specification or WS-BPEL is executed. The Service Level Agreement (SLA) of the composite service is also produced during this step to specify the agreed levels of quality for the composite service. In case an error occurs during the execution phase, an alert is generated with the specified error which must be analyzed by the system administrator or by a Decision Making Module with reasoning capabilities.

Step 8 Composed service Verification and Validation is done prevent an unexpected behaviour. Similar as in software testing automated smoke tests can be defined and executed to verify and validate the composed service. To address possible Web Service dependencies BPEL Unit Testing framework can be used.

3. Case Study: Indoor Air Quality Monitoring System within Environmental Health domain

Environment for health is a branch of public health closely related to environment science that is concerned with all aspects of the natural and built environment that may affect human health.

Indoor air quality is an important aspect in the management of any building, being a factory, school or an office. Studies have shown that poor IAQ has a major impact on an individual's health and it also influences its performance.

Emerging technologies from the Internet of Things and the Internet of Services can help in achieving a better management of IAQ, by coupling "smart" objects capabilities with service oriented software. In this thesis "smart" objects are understood as devices that make use of technologies starting with radio-frequency identification (RFID) tags and smartcards to embedded chips that carry information with them. In the literature there are significant contributions to IAQ monitoring that make use of sensing technologies to provide cost-effective solutions to the matter in discussion [19], [20].

3.1. Use Case description

Let's consider the case of a company located in France, which contracts services for IAQ management of the building. The company mounted the necessary hardware equipment ("smart" sensors, HVAC systems, surveillance cameras etc) for implementing a MIAQ system.

An employee fills in a complaint about the temperature in the office room 12b. The Building Administrator (BA) receives a notification that a complaint has been received and logs in to IAQ Monitoring System (MIAQS), views the complaint and makes a request to view status on temperature from a list with several parameters (CO₂, temperature, humidity ventilation etc). The MIAQS presents the building's health score (poor) based on the selected parameter and some options the BA can take: modify policy for HVAC functioning, schedule HVAC revision, send notification to building maintenance personnel. The BA immediately notifies the maintenance personnel to check HVAC systems which discover that some HVAC systems at the 12th floor are malfunctioning. Then the BA chooses to schedule a HVAC revision with a specialized company by entering: company's address, his contact information, urgency, maximum cost for revision service. The BA also needs to adjust HVAC management policy to have HVAC systems running only during work hours (9:00 - 19:00), to reduce costs and increase system's lifetime. Also BA responds to complaint submitted by the employee to assure him/her that the problem will be fixed.

3.2. Implementation and Results

The MIAQS functionalities are implemented by Web Services. We assume that the Web Services are provided by a Service Broker. The Service Broker manages a service registry (e.g. UDDI) where service providers publish their Web Services, but it also offers composed Web Services. The QoSC framework is implemented by the Service Broker. The company using the MIAQ system pays a

monthly subscription for the WS used, which guarantees a certain level of quality negotiated between the company and the SB. The level of quality agreed is specified in a Service Level Agreement and explicitly states the quality criteria for WS used to automate the MIAQ system's functionalities. Within the subscription fee the company expects services with high reputation, usability, high availability and throughput. The company's domain area of activity (banking) and security policies require also the utilization of WS that offer integrity and confidentiality of company's sensitive data.

For implementation of the WS composition we used Netbeans IDE 6.5 which allows through the use of SOA plugin and WS-BPEL engine to implement a service orchestration.

WSs implementing MIAQ system's functionalities need to comply with company's quality requirements but also fulfill the end user's preferences. Quality criteria is extracted from context assessment and mapped to Company Quality Profile and User Quality Profile. Company Quality Profile (CQP) is extracted from non-functional requirements specified the SLA established between the company and the SB. The BA's requirements, both functional and non-functional, are mapped to the QoSC model, creating the User Quality Profile. These quality profiles are merged and the Service Selection Specification is obtained.

Service discovery and selection processes are based on the QoS criteria identified at the previous phase. Service registries are queried to identify services with required functionalities and QoSC. To select the best matching services accordingly to QoSC specification, each candidate service is tested. At this step, selection of services is not only based on QoS non-functional requirements as realized in most of state of the art approaches, but also interoperability and adaptability of services that need to interact is verified. After having identified the services that match the requirements, both explicit and implicit, the specification of the service composition is developed. The service composition plan represents the specification for the execution engine. In our use case service orchestration is realized using the WS-BPEL engine implemented in Netbeans IDE.

After the service composition plan is generated it must be verified and validated first according to some criteria specific to the execution engine and composition method, to avoid inconsistencies and mismatches between the plan and the execution engine. In this case the WS-BPEL service composition plan is verified using the Test Case Generator module that allows definition of unit tests for each WS to ensure a proper invocation during the composition.

Finally the builder executes the service composition specification and produces an implementation corresponding to the required composite service. The SLA of the composite service should be also produced during this step, but this is out of scope for our case study. In case an error occurs during the execution

phase, an alert is generated with the specified error which must be analyzed by the system administrator.

During the composition plan execution, the WS-BPEL specification is compiled, the result being the composite WS. Unit level verification of the composite Service is realized using WS-BPEL's test case generator. For validation of the proposed QoESC framework we used variation of WS parameters involved in the composition to simulate scenarios where unreliable, non-interoperable or insecure services would have been used in the service composition.

To evaluate the service composition reliability we used the estimation method for reliability rate and repair rate provided by Xie, Li and Wang [21]. We implemented algorithm I (i.e. reliability prediction based on failure behaviour) for the first case (a) and algorithm III (i.e. reliability prediction based on failure and repair rate) for reliability estimation in the second case (b) where faults caused by lack of interoperability can be detected in time and repaired. For case a) we set the reliability of WSs to $r = [0.8, 0.5, 0.9, 0.6, 0.3, 0.5, 0.7, 0.5]$, $n = 10$, the number of total runs = 10 to 100, step by 10. For case b) we use the same reliability r and repair rate $r_r = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$ suggesting a repair probability of 50%.

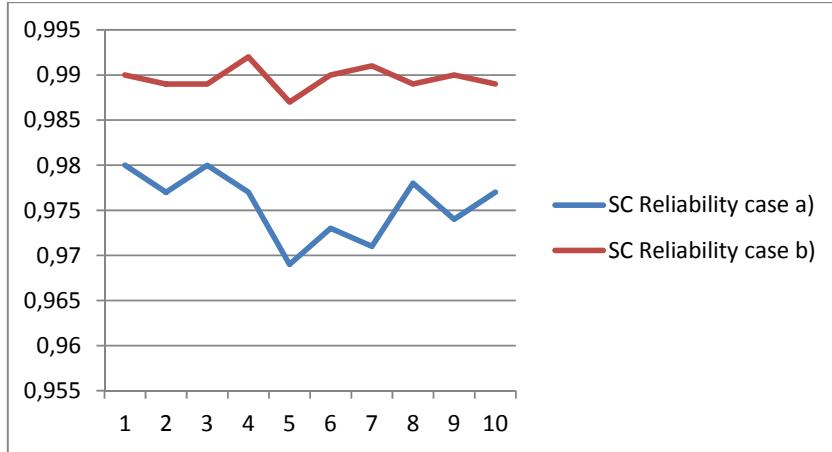


Fig. 4. Reliability for WS-BPEL in cases (a) and (b)

We can observe from figure 4 that the reliability of service composition, when faults are detected and repaired in due time because quality attributes such as interoperability are evaluated, is greater than reliability of service composition when these quality items are not taken into consideration. Increased service composition reliability means an improved quality of service composition and composed service, thus user satisfaction.

4. Conclusions

In this work we provide a quality modelling framework for service composition, independent of the composition language or platform. The framework addresses issues that might have an impact on the quality of the composed service such as interoperability, adaptability or security. Verification and validation is realized during the entire process, from user requirements gathering and service selection to execution of the service composition. A prototype of the proposed framework was implemented based on a case study on Indoor Air Quality Management within Environmental Health domain. The case study considers an MIAQ system used by a company in its building management program. To evaluate the advantages brought by the quality driven framework for service composition reliability has been calculate in a scenario where only QoS attributes used in stated of the art approaches and in a scenario where Quality of Interoperability was also verified. Results show that evaluation of WS interoperability improves reliability of the service composition.

The main objective of the proposed framework is to provide a methodology for development and assessment of Web Service compositions. The QoSC model layered on three quality dimensions of the service composition should serve as a guideline for organizations that use Web Services in their Business Process. The model can be customized to suit their specific quality requirements. The QoSCP model addresses service composition process in general and can be applied to different compositions methods: orchestration, choreography or coordination. Although certain technologies to implement a prototype for the MIAQ use case were used, these were selected based on the ease of use and are not dependent of the proposed framework.

R E F E R E N C E S

- [1] *European Comission, Future Internet Enterprise Systems (FInES)*. (2011) Position Paper on Orientations for FP8, A European Innovation Partnership for Catalyzing the Competitiveness of European Enterprises. [Online] <http://cordis.europa.eu/fp7/ict/enet/documents/fines-position-paper-fp8-orientations-final.pdf>
- [2] *T. Erl*, SOA Principles of Service Design.: Prentice Hall, 2008
- [3] *L. Zeng, B. Benatallah, and M. Dumas*, "Quality Driven Web Services Composition," in Proceedings of the 12th international conference on World Wide Web, ACM, New York, 2003.
- [4] *T. Alrifai and M. Risse*, "Efficient QoS-aware Service Composition," in In Proceedings of the 18th international conference World Wide Web, New York, USA, 2009
- [5] *Hyunyoung Kil Nam and Wonhong*, "Anytime algorithm for QoS web service composition," in In WWW'11 Proceedings of the 20th international conference companion on WWW, New York, USA, 2011
- [6] *Q. Yu, M. Rege, and Bouguettaya*, "A two-phased framework for quality-aware Web Service

selection," in IEEE International Conference on Service Oriented Computing & Applications – SOCA, 2010.

[7] *Farroha B. S. L. and Farroha D.*, "Policy-based QoS Requirements in a SOA Enterprise Framework – An Investigative Analysis," in 3rd Annual IEEE Systems Conference, Vancouver, Canada, 2009

[8] *Badidi E.*, "A Scalable framework for Policy-based QoS management in SOA Environments," Journal of Software, vol. 6, no. 4, 2011

[9] *Yeom Gwyduk, Wei-Tek Tsai, Xiaoying Bai, and Dugki Min*, "Design of a Contract-based Web Services QoS Management System," in In ICDCSW '09 Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshop, Washington, USA, 2009

[10] *R. Nyguen and X. T. Kowalczyk*, "Agent-Based QoS Conflict Mediation for Web Service Compositions," in In IAT '06 Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, Washington, USA, 2006

[11] *J.S. Malak, M. Mohsenzadeh, and M.A. Seyyedi*, "Web Service QoS Prediction Based on Multi Agents," in In Proceedings of ICCTD '09 International Conference on Computer Technology and Development, Kota Kinabalu, Malaysia, 2009

[12] *X. Zhao et al.*, "An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition," Applied Soft Computing Journal, Elsevier, 2012

[13] *C. Lin, R. Sheu, Y. Chang, and S. Yuan*, "A relaxable service selection algorithm for QoS-based web service composition," Information and Software Technology Journal, pp. 1370-1381, 2011

[14] *C. Zeginis and D. Plexousakis*, "Web Service Adaptation: State of the art and Research Challenges," Technical 410, ICS-FORTH, 2010

[15] *G. Stegaru, C. Danila, I.S. Sacala, M.A. Moisescu and A. M. Stănescu*, "Quality driven Web Service Modeling Framework," In 13th IFIP Working Conference PRO-VE, pp. 83-90, 2012

[16] *Shuping Ran*, "A model for Web Service Discovery with QoS," ACM SIGecom Exchange, vol. 4, no. 1, 2003

[17] *G. Stegaru and A.M., Sacala I.S. and Moisescu, M. A. Stănescu*, "Dynamic Interoperability Model for Web Service Choreographies," 6th International Conference on Interoperability for Enterprise Systems and Applications, vol. 5, no. 2, pp. 81-91, 2012

[18] *G., Kateros, D.A., Foukarakis, I.E., Prezerakos, G.N., Karlamani, D.I. and Venieris, I.S. Kapitsaki.* (2007) Service Composition: State of the art and future challenges. [Online]. <http://www.cin.ufpe.br/~redis/intranet/bibliography/middleware/kapitsaki-composition07.pdf>

[19] *O., Girao, P., Pereira, D., Ferraria, G., Barroso, N. and Postolache, G. Postolache*, "Indoor monitoring distress triggering factors using a wireless sensing network and a smart phone," in Instrumentation and Measurement Technology Conference, pp. 451-466, 2009

[20] *Y., Li, K., Tian, L., Piedrahita, R., Yun, X., Manasata, O., Lu, Q., Hannigan, M. and Shang, L. Jiang*, "MAQS: A personalized mobile Sensing System for Indoor Air Quality Monitoring," in 13th international conference on Ubiquitous computing, pp. 271-280, 2011

[21] *C. Xie, B. Li and X.A. Wang*, "Web Service Reliability Model based on Birth-Death Process", 23rd International Conference on Software Engineering and Knowledge Engineering – SEKE, pp. 625-628, 2011