# Audit Techniques for Service Oriented Architecture Applications

Liviu COTFAS, Dragoş PALAGHIŢĂ, Bogdan VINTILĂ
Academy of Economic Studies, Bucharest, Romania
liviu.cotfas@ase.ro, mail@dragospalaghita.ro, vb@vintilabogdan.ro

*The Service Oriented Architecture (SOA) approach enables the development of flexible distributed applications. Auditing such applications implies several specific challenges related to interoperability, performance and security. The service oriented architecture model is described and the advantages of this approach are analyzed. We also highlight several quality attributes and potential risks in SOA applications that an architect should be aware when designing a distributed system. Key risk factors are identified and a model for risk evaluation is introduced. The top reasons for auditing SOA applications are presented as well as the most important standards. The steps for a successful audit process are given and discussed.*

***Keywords:*** *Service Oriented Architecture, Audit, Quality Attributes, Interoperability, Performance, Security*

## 1 Introduction

With the increase of the informatics transactions between global organizations more efficient, flexible and reliable software products are needed. In order to provide end-users with these kinds of applications a rigorous audit process must be performed.

The audit process must evaluate the quality of the service oriented architecture. According to [1] a quality of service measure must be established in order to quantify SOA quality with regard to certain aspects of the service oriented architecture like deliverables, deadlines, software quality and costs.

In [2] the most important aspects to consider when estimating Quality of Service (QoS) for web services are:

- the execution cost and time;
- availability;
- successful execution rate;
- reputation;
- usage frequency.

The audit process must establish if the development processes are in compliance with the rules and regulations imposed by external entities and to which the organization adhered to. This is done by:

- identifying external requirements;
- review existing laws and regulations that the organization must be compliant with;
- establish whether the company considered these laws and regulations when the policies and procedures were developed;
- determine if employees adhere to existing policies and procedures, and if omissions exist.

Security audit has the goal to identify possible vulnerabilities and threats. The results of the audit process can be used in Risk management which typically requires following steps:

- identify assets;
- identify threats;
- perform risk analysis;
- perform risk mitigation;
- monitor software product behavior.

The outcome of the risk management process is the mitigation of critical threats and the development of fixes to resolve discovered vulnerabilities.

## 2 Service Oriented Architecture

The era of standalone applications and isolated computers is long gone and the Internet plays a major role in the development of companies and individuals. Standalone applications assume the existence of the binaries of the application on the user's computer and the user can use only the functionality the application has at a certain moment without the possibility to enlarge the scope. Updates are difficult and sometimes these bring in more unwanted stuff than desired content and functionality. The appearance of Internet brought new models of application design. The client-server model is widely used and it assumes the existence of a server and a client. The server hosts the application and the client access it whenever needs to. No processing is usually done on the client side making. This allows low-performance clients to access applications with

large consumption of resources. This model also has the advantage of easy updates and low resource consumption of resources reported to the number of users. The server executes the client's request and sends back the results. A connection is maintained between the two until the client disconnects or for a period in which the client makes no requests to the server.

In order to improve the quality of the applications, the service oriented architecture provides a loosely tied collection of services that can be used in many domains. The services implement single actions and are not dependent of each other. The interoperability is ensured through the use of the XML file format [3]. This is independent of platform and thus the services are independent too. There is no problem if some of the services are running on an Unix server, others on a Windows server and the application that accesses them is running on either a Unix of Windows server and is accessed by a mobile client. The XML format ensures a smooth transition of data between services and so that every service gets the data it needs and fulfills the user's request [4]. The service oriented approach is also useful for easily creating ad-hoc applications that implement only the functionality one wants.

Let $S=\{s_1, s_2, s_3, \ldots, s_n\}$ be a set of services that a company has installed and ready to run on their server. A client *C* wants to have access only to the orders it makes and to the financial summary at the end of the month. In the classic approach of applications the users would have been forced to access a standard application and then select the desired options. The service oriented approach allows the custom creation of an application for a specific client. From the *S* set is selected a subset *SS* formed only by the services the client *C* wants to access and, on the basis of a template, an application is built exposing only the *SS* subset. The client has in the application only the options that wants and needs no more to select many options and navigate through unwanted features. This improves the company-client relationship as the client feels the company treats him right and cares for his needs.

The service oriented architecture has the advantage of updating the application in small steps with no need to stop the application or have a dedicated server for the testing of the new services. The testing of the services can be done on the same server as the users can't access them unless these are exposed through the application. The simplicity of the update or upgrade process is a key factor for the maintenance process. The costs for this type of update are also much smaller than for a classic update in the case of a client-server application.

The development, usage and maintenance of the service oriented applications are ruled by the following principles:

- Reuse, granularity, modularity, composability, componentization, interoperability [5]; reuse is a major must in the service oriented architecture as a service, if used only by one client or only by one application has not much value; services must be designed so that many applications use them and their utility for the users is as great as possible; granularity is imposed by the need of reuse because a service that makes many operations loses its generality and can't be reused; assuming we have a service that takes the data of a customer and realizes its history on the bases of his unique identifier, the service can't be used for anything else; if, for instance we have a service that gets us only the history of the customer for a specified time span, we can use this one to show the user the last month or last year history; designing the services as simple as possible makes the task of putting the application together tougher but improves the reuse of the services; composability assumes the possibility of solving a client's request by using more than one existing services; if the services don't follow a logical path and can't be composed they are useless; grouping a set of services that treat the same issue under the hood of a component strengthens the logic of the application that is based on these services; interoperability of services assumes the possibility to work with other services regardless the hardware platform they run on; this is very important as companies use different hardware according to needs, preferences or existing resources; all these are very important for the service oriented applications as the lack of any of these sets the application and services on the path to perdition; Old applications can also be converted to web services [6].
- Standards-compliance [5]; in every industry there are standards that must be complied with in order to have a good product; quality is a relative term when there are no standards; one can appreciate a product as being qualitative by personal criteria unrelated to the real characteristics of that specific product

while another might appreciate the same product as being of lousy quality by using another set of personal criteria; when standards state which are the characteristics of a product and what is the degree these must fulfill in order to be considered a quality product we have an objective process deciding the quality level; the compliance with the standards ensures that the designed services and applications are accepted on the market as qualitative products; the user satisfaction is determined by the performance and features of the application, but without standards compliance the application would not even be taken into account by a possible client;

- Services identification and categorization, provisioning and delivery, monitoring and tracking [5]; the identification of services by a unique identifier is a must be as the server would not know what service to access at a user's request; there is also good to have service identifiers state the processing made within the service; the categorization of services is helping application designers chose more quickly from the wide range of services; all services must deliver something; because of the interoperability need, the services must return results in the form of XML files; a service that does not return such a file, returns nothing and thus has no use; monitoring and tracking services is useful for knowing which of them are most used, which cause problems, which need optimizations; assuming we have a service that all customers access with each access of the application; this service must be very quick and also return the users the desired results; having a service that is frequently used and that takes ages to return something leads the application to quick death.

Service oriented blocks can have the role of service provider or service consumer. The service provider ensures functionality for users as these consume the services made available. The service consumers send requests to the service providers and then they receive the results.

Service oriented architecture operates independent of specific technologies. Developers can implement SOA using a wide range of technologies [7]: Simple Object Access Protocol (SOAP), Remote Procedure Call (RPC), Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), Web Services, Windows Communication Foundation (WCF). Web services are a quick way of having services running in no time and ready for the service consumers.

One of the disadvantages of the service oriented architecture is the slower processing caused by the use of XML parsing for solving the interoperability issue. Even though the processing is slower than in the care of tight tied applications, the advantages of the service oriented architecture are of greater value than the disadvantages.

## 3 Quality Attributes of SOA Applications

Important quality attributes of Service Oriented Applications include: interoperability, performance, security, reliability, availability, modifiability, testability, usability, scalability. All these quality attributes have to be analyzed in order to determine if SOA is a valid choice for a specific project. We discuss bellow the first three quality attributes, that apply to most projects.

### 3.1 Interoperability

Interoperability of services assumes the possibility to work with other services regardless the software or hardware platform they run on, an aspect particularly important in the current changing economic conditions. As, fully integrated traditional enterprises are being replaced by business networks in which every company is specialized in certain services or products [8], [9], [10] interoperability becomes an ever important issue. By deploying interoperable systems, supply chains can be more rapidly established allowing the involved companies to benefit from reduced costs and increased flexibility.

Web service interoperability can be analyzed both at semantic and syntactic levels. At a semantic level, web service interoperability relies on shared ontologies and is particularly important in areas such as automatic and semi-automatic web service composition [11]. At a syntactic level, standards such as Web Service Definition Language (WSDL) and Simple Object Access Protocol (SOA) are used [12], allowing web service implemented using different programming languages, running on different hardware and under different operating systems can communicate based on pre agreed interface format and communication protocol. In order to achieve cross-platform and cross-vendor Web-Services Interoperability Organization (WS-I) was established in order to address issues

regarding web service interoperability. Large, legacy applications can also be integrated in SOA systems and can take advantage of web service interoperability as shown in [7].

### 3.2 Performance

Performance is key issue for any technology that directly affects its applicability in real world applications.

SOA facilitates the development of distributed systems, spread across multiple machines connected over the internet. Although distributing a processing task is usually seen as a method to increase performance, slow connections and reduce parallelism in SOA usually leads to higher response times as information is sent over the network. Additional performance overhead is introduced by the need to use data serialization and deserialization techniques. Higher response time can make SOA unsuitable for certain real-time applications [12].

In order to assure a high degree of interoperability several compromises have been made that negatively affect SOA applications performance. Using standard XML for web services enables interoperability between different platforms and operating systems, but also increases application response time. XML text-based messages can be up to twenty times larger than equivalent binary messages and also require operations such as validation and parsing before the contained data can be used. A possible approach to this problem is using binary XML as described in [13].

Identifying performance bottle necks in SOA applications can prove difficult when services from multiple providers are combined in web-service chains. Moreover, performance issues in any service will affect the entire chain. Performance evaluation is of particular importance in case of automatic and semi-automatic web service composition as the user building the chain might only have basic computer knowledge [14]. Performance also directly influences other quality attributes such as Scalability.

### 3.3 Security

A key challenge in service oriented architecture applications is assuring security across multiple computers, usually communicating using unsecured networks. Many features that make SOA an attractive paradigm such as web service composition, messages sent as clear text and relative autonomy of base web services conflict with traditional security models [15]. Web service security includes aspects such as: integrity, confidentiality, availability. Security audit of service oriented architecture applications is particularly challenging as web services can be added or removed at any time. Taking into consideration that the distributed nodes can be controlled by different providers, it becomes mandatory that all providers should comply with the same security rules.

If messages between web services are sent without using an encryption mechanism, all information together with the describing meta-data can be intercepted by an attacker. No confidential data should be thus sent without taking the necessarily security measures like using Secure Sockets Layer (SSL) or Kerberos. Also, all providers should comply with the same security principals regarding storing and processing confidential data.

Security of service oriented architecture applications is still an issue and multiple standards are currently being developed: WS-Authorization, WS-Privacy, WS-Trust, WS-Federation, WS-Policy, and WS-SecureConversation.

## 4 Potential risks in SOA Applications

The CSI/FBI 2008 report on Cyber Security states that virus incidents occur most often in 49% of cases, whereas insider abuse was found to occur 42% of breaches. The average annual loss in 2008 was estimated at just under 300,000 $. The 2009 CSI/FBI report on Cyber Security estimates the annual loss due to security incidents to 234,244 $. The 2009 report found that the insider abuse rose to be responsible for 60% of breaches. The reports state that the most important security breaches were caused by viruses, insider abuse and laptop theft.

The online environments in which most of the services exiting today operate are exposed to considerable larger number of threats than those that run on isolated networks. The intensive development of software architectures and the increase in complexity of customer needs will most probably create new vulnerabilities of which attackers can take advantage in order to cause new and more damaging types of security breaches. Figure 1 lists the main factors that must be managed after a security incident occurs. System costs are immediate to short term and organizational costs to be short to medium term costs.

All of the costs listed in figure 1 can be avoided

or at least minimized by identifying the risks the information system can encounter. The risk assessment is an important step in preventing informatics security attacks by eliminating the most damaging means an attacker has to successfully create a security breach in the software product. In [16] is described a five step risk model.

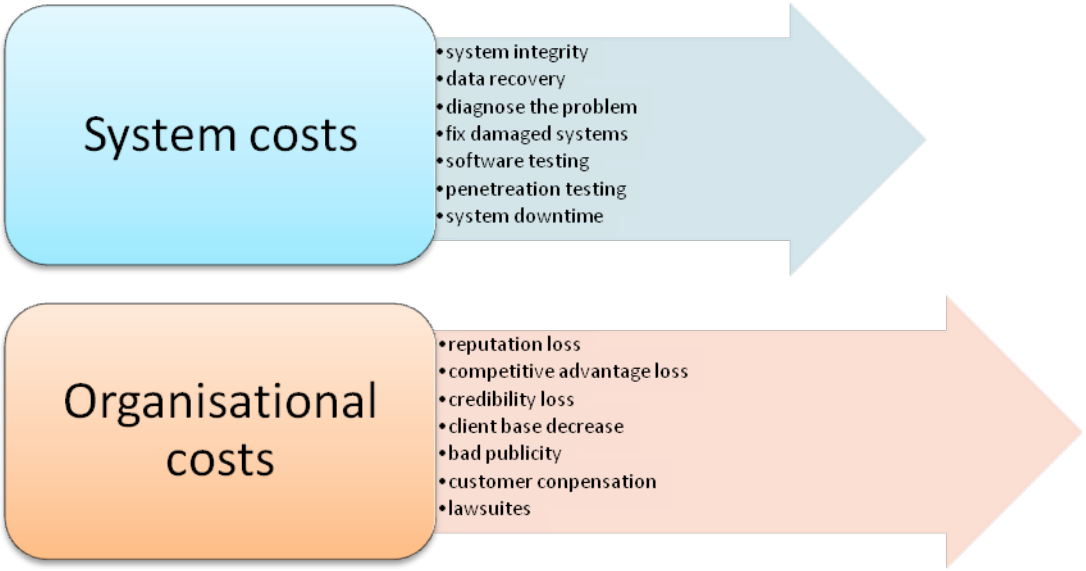


**Fig. 1.** System breach costs

In **step 1** there are two models:
A. The vulnerability model, formed from the set of vulnerabilities $SV = \{V_1, V_2, \ldots, V_n\}$ which has associated the set of probabilities $PSV = \{PV_1, PV_2, \ldots, PV_n\}$. Where $V_i$ represents the vulnerability *i*, and $PV_i$ is probability that the vulnerability was being exploited. The result of the vulnerabilities model is the vulnerability matrix, presented in table 1.

**Table 1.** Vulnerability matrix

| Vulnerability | Exploit probability |
|---|---|
| $V_1$ | $PV_1$ |
| $V_2$ | $PV_2$ |
| … | … |
| $V_n$ | $PV_n$ |

B. The threat model is composed of a set of threats $SA = \{A_1, A_2, ..., A_m\}$, with has associated a set of probabilities= $\{PA_1, PA_2, \ldots, PA_m\}$. $A_i$ is the threat *i*, and $PA_i$ represents the probability of the threat to quantify. The result of the threat model is the threat matrix presented in Table 2.

**Table 2.** Threat matrix

| Threat | Appearance probability |
|---|---|
| $A_1$ | $PA_1$ |
| $A_2$ | $PA_2$ |
| … | … |
| $A_m$ | $PA_m$ |

**Step 2** has been materialized in the development of the goods model which is represented by all the goods used by the application $SB = \{B_1, B_2, \ldots, B_k\}$.
**Step 3** represents the analysis of risks occurrence probability. It is considered that a threat exploits a set of vulnerabilities, each vulnerability of the set has an exploit probability $PV_i$, thus a matrix of perceived risk is developed by adapting the model described by [17] adding the consideration of threats at the vulnerability collectivity level, the matrix is presented in table 3.

**Table 3.** Exposure to risk model through the sets of vulnerabilities

| Threat | Appearance probability | | Vulnerability set | $B_1$ | $B_2$ | $B_3$ | … |
|---|---|---|---|---|---|---|---|
| $A_1$ | $PA_1$ | → | $SV_1$ | $PSV_1EB_1$ | $PSV_1EB_2$ | $PSV_1EB_3$ | … |
| $A_2$ | $PA_2$ | → | $SV_2$ | $PSV_2EB_1$ | $PSV_2EB_2$ | $PSV_2EB_3$ | … |
| $A_3$ | $PA_3$ | → | $SV_3$ | $PSV_3EB_1$ | $PSV_3EB_2$ | $PSV_3EB_3$ | … |
| … | … | → | … | … | | | … |

In table 3 $SV_i$ is the $i^{th}$ vulnerability set and $PSV_iEB_j$ is the exploit probability of one or more vulnerabilities from set *i* in order to get access to asset *j*. $PSV_iEB_j$ is computed as:

$$PSV_iEB_j = \prod_{k=1}^{v} PV_{ik} * EB_j$$

$PV_{ik}$ is the exploit probability of vulnerability *k* from vulnerability set *i* by a threat $A_l$. **Step 4,** mitigating the risk, consists in the use of countermeasures to minimize the threat. Thus each threat $A_i$ from the set *SA*, is associated with a set of reduction factors $SFD = \{FD_1, FD_2, \ldots, FD_k\}$.

The probability of risk quantification through a set of vulnerabilities is determined using the formula below:

$$R = \sum_{i=1}^{m} \left[ PA_i * \frac{\left( \sum_{j=1}^{k} \mathrm{PSV_iEB_j} \right)}{k} * FD_i \right] * \frac{1}{m}$$

Using the above formula a clearer picture is given over the quantitative aspects of risk. In figure 1 the risk model analysis model is presented.

**Step 5** is needed for future risk analysis as this will speed up the process in the future. Keeping clear and well documented records regarding past threats and vulnerabilities will provide the development team valuable statistical information regarding the evolution of the vulnerabilities and threats models effects on the asset model thus helping to develop better countermeasures to handle the risk.

The presented risk model will improve the quantitative assessment of the risk environment the software system operates in. The security risk should be decreased to a level that is considered as affordable by the organizations' management thus if a security breach occurs the initial risk assessment will enable the damage control procedure to be more effective.

## 5 Audit in SOA applications

Taking into consideration the large number of applications that aroused in the last years, the need of having someone certificating their quality was imperious. The audit appeared to fulfill this need and guarantee the quality of applications that were audited. The auditor, through the audit process, takes on himself the responsibility of the audited application. If there are any errors that are undiscovered by the audit process and the application encounters them after the audit process was successfully ended, the auditor is fully responsible for it. The audit has many issues but the ones of interest for us regarding the service oriented architecture are software quality assurance and software audit review. The top ten reasons for starting up IT auditing are [18]:

- Auditing around the computer was becoming unsatisfactory for the purpose of data reliance;
- Reliance on controls was becoming highly questionable;
- Financial institutions were losing money due to creative programming;
- Payroll databases could not be relied on for accuracy due to sophisticated programmers;
- The security of data could no longer be enforced effectively;
- Advancements occurred in technology;
- Internal networks were being accessed by employees' desktop computers;
- Personal computers became accessible for office and home use;
- Large amounts of data required advanced software programs to audit them, known as CAATs (Computer Assisted Audit Technique);
- The tremendous growth of corporate hackers, either internal or external, warranted the need for IT auditors.

Software quality assurance is the process of monitoring the developing tools and the used techniques for ensuring the quality of the product during the development cycle and the compliance with the software standards and procedures. The types of standards the application must comply with include [19]:

- Documentation standards specify the form and content of the planning, control and product documentation; the documentation ensures consistency throughout the project ongoing;
- Design standards specify the form and content of the product; these provide methods and rules for transforming the specifications of the project into software design;
- Code standards specify the coding language in which the application is to be coded and any restrictions regarding the use of features of the language; they define language structures, stile conventions, rules for data structures and interfaces and internal code documentation.

The activities of the software quality

assurance process are product evaluation and process monitoring. Product evaluation ensures that standards are being followed. Process monitoring compares the steps made for the fulfillment of the goal with those described in the documented procedures. The software quality assurance includes monitoring baseline control, configuration identification, configuration control, configuration status accounting and configuration authentication.

Software audit review is a type of software review in which one or more auditors who are not members of the company that realized the product conduct an independent examination of the software product to assess compliance with specifications, standards, contractual criteria or other criteria. The purpose of the audit process is to provide an external assurance of compliance with standards, guidelines, plans and procedures.

In the case of the service oriented architecture based applications, the audit is very important as the producing company ensures it is no longer responsible for the application's errors. Considering the fact that the applications are made of a collection of services executed in a logical order, the audit process must focus mostly on the services. The audit of all services used by an application leads to a correct execution of the application if the application's logic is correct. Auditing the application is also important in order to have the guarantee that the application's logic is correct.

The audit of the services has the following advantages:

- Considering the service is unmodified, the audit guarantee is available indefinitely;
- Only new and modified services are audited again leading to smaller costs;
- An audited service can be used by many applications;
- The quality of the service is guaranteed.

For the audit process the following must be done:

- identifying external requirements; this is done by examining the product's specifications; in the specifications the characteristics of the final product are clearly stated; the external requirements can also be obtained through discussions with the entity that required the audit process; this entity is interested in some special characteristics and the audit process must evaluate the product regarding them; this type of audit is made by entities that need a product but are not sure if it complies with all the requirements; the audit process clearly establishes the compliance of the product to the needs of the company;
- review existing laws and regulations that the organization must be compliant with; some of the product purchases are done in order to comply with the new legislation issues or with the ones already existent but not satisfied by the company; all laws and regulations that the company is ruled by must be reviewed and evaluate if the company complies with them or not; if the company does not comply with all the laws and regulations, measures must be adopted in order to fulfill all requirements;
- establish whether the company considered these laws and regulations when the policies and procedures were developed; the policies and procedures within the company are compared against the laws and regulations to establish if they still comply; if they don't comply they must be modified or changed;
- determine if employees adhere to existing policies and procedures, and if omissions exist; even if policies and procedures exist within the company this does not mean that the employees use them; the audit process must verify this thing; this is done by analyzing the work documents, activities, reports, outputs; if an employee does not respect the policies and procedures within the company the product of his work is very likely to be incompliant with the standards.

Audit process of the service oriented architecture applications is a must be as there are many users that access these types of applications and errors are inadmissible. The audit process is meant to detect the incompliance of the product with standards, procedures or external requirements. Through the audit process the responsibility is passed from the entity requesting the audit process to the entity executing it. The auditor is responsible if the things he states are proven wrong.

## 6 Conclusions and Future Work

The service oriented architecture brings many advantages over the traditional approach of applications. The use of services enables developers to create applications very quick and with very high quality, assuming that the used services have been audited. If the principles for developing, using and maintaining the service oriented architecture applications are respected, these ensure long life cycles. The audit process is caused by serious and various reasons. The

applications must comply with standards in order to pass the audit process. The audit of the services and applications has many advantages for the owner. The audit is successful if external requirements are identified, laws and regulations are reviewed, policies and procedures were created complying the laws and if the employees apply the policies and procedures.

Risk assessment is important in SOA audit due to the fact of making unforeseen threats and vulnerabilities visible to the quality manager in charge of the project. By identifying the risk factors the audit process makes it possible to deliver a high quality product with a high security level.

Our paper presents both the importance and challenges of auditing SOA applications as well as a model for risk assessment. Further development includes the development of an audit model for semantic web service composition.

**Acknowledgements**

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Program for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

**References**

[1] J. Cardoso, A. Sheth, J. Miller, J. Arnold and K. Kochut, "Quality of service for workflows and web service processes," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 1, No. 3, April 2004, pp. 281-308.

[2] J. Myoung Ko, C. Ouk Kim and I. H. Kwon, "Quality-of-service oriented web service composition algorithm and planning architecture," *Journal of Systems and Software*, Vol. 81, No. 11, November 2008, pp. 2079-2090.

[3] IBM, *IBM Service Oriented Architecture*. [Online]. 2010, Available at: http://www-01.ibm.com/software/solutions/soa/.

[4] M. Rosen, B. Lublinsky, K. T. Smith and M. J. Balcer, *Service-oriented architecture and design strategies*, Wiley Publishing, 2008.

[5] M. Bell, *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*, 1st ed., Wiley, 2008.

[6] Microsoft Corporation, *Microsoft Patterns and Practices - Web Service Facade for Legacy Applications*. [Online]. 2003, Available at: http://msdn.microsoft.com/en-us/library/ms979218.aspx

[7] Microsoft, *Understanding service-oriented architecture*. [Online]. 2010, Available at: http://msdn.microsoft. com/en-us/library/aa480021.aspx

[8] S. Gosain, A. Malhotra and O. A. ElSawy, "Coordinating for Flexibility in e-Business Supply Chains," *Journal of Management Information Systems*, Vol. 21, No. 3, 2005, pp. 7-46.

[9] I. Smeureanu and A. Diosteanu, "A Collaborative System Software Solution for Modeling Business Flows Based on Automated Semantic Web Service Composition," *Informatica Economica*, Vol. 13, No. 2, 2009, pp. 32-40.

[10] K. Kuldeep, "Technology for supporting supply chain management: introduction," *Communications of the ACM*, Vol. 44, No. 5, June 2001, pp. 58-61.

[11] G. A. Lewis, E. Morris, S. Simanta and L. Wrage, "Why Standards Are Not Enough to Guarantee End-to-End Interoperability," in *Seventh International Conference on Composition-Based Software Systems*, 2008, pp. 164-173.

[12] L. O'Brien, P. Merso and L. Bass, "Quality Attributes for Service-Oriented Architectures," in *International Workshop on Systems Development in SOA Environments (SDSOA'07: ICSE Workshops 2007)*, 2007, pg. 3.

[13] W3C, *XML Binary Characterization Use Cases*. [Online]. 2005, Available at: http://www.w3.org/TR/xbc-use-cases/

[14] A. Diosteanu and L. A. Cotfas, "Agent Based Knowledge Management Solution using Ontology, Semantic Web Services and GIS," *Informatica Economică*, Vol. 13, pp. 90-98, December 2009.

[15] E. Bertino, L. Martino, F. Paci and A. Squicciarini, *Security for Web Services and Service-Oriented Architectures, 1st ed.*, Springer, 2009.

[16] D. Palaghita and B. Vintila, "Security Risk Analysis and the Security Need in Citizen Oriented Applications," *Economy Informatics*, Vol. 9, No. 1, pp. 79-86, September 2009.

[17] H. P. In, Y. G. Kim, M. Chang-Joo, Y. Jung and I. Kim, "A Security Risk Analysis Model for Information Systems," in *Systems Modeling and Simulation: Theory and*

*Applications*. Berlin, Germany: Springer Berlin / Heidelberg, pp. 505-513.

[18] S. Senft and F. Gallegos, *Information Technology Control and Audit, 3rd ed.*, CRC Press, 2009.

[19] NASA, *Software Quality Assurance*. [Online]. 2008, Available at: http://sw-assurance.gsfc.nasa.gov/.

**Liviu COTFAS** is a Ph.D. student and a graduate of the Faculty of Cybernetics, Statistics and Economic Informatics. He is currently conducting research in Economic Informatics at Bucharest Academy of Economic Studies and he is also a Pre-Assistant Lecturer within the Department of Economic Informatics. Amongst his fields of interest are geographic information systems, genetic algorithms and web technologies.

**Dragos PALAGHIȚĂ** graduated from the Academy of Economic Studies of Bucharest, Cybernetics Statistics and Economic Informatics faculty, Economic Informatics section in 2008. He is programming in C++ and C# and his main areas of interest are Informatics Security, Software Quality Management, large data set analysis and graphical representation enhancements. Currently he is undergoing PhD studies at the Academy of Economic Studies of Bucharest, Cybernetics Statistics and Economic Informatics. He published 14 articles in JAQM, Informatica Journal, Economie Teoretica si Aplicata journal, Revista Romana de Automatica si Informatica.

**Bogdan VINTILĂ** graduated the Bucharest University of Economics, the Faculty of Cybernetics, Statistics and Economic Informatics. He is currently a PhD candidate in the field of Economic Informatics at University of Economics. He is interested in citizen oriented informatics applications, developing applications with large number of users and large data volumes, e-government, e-business, project management, applications' security and applications' quality characteristics.