International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 2, February 2015
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 46-54

Security Decision Making Framework for SOA

Amr Galal[1], Hesham Hassan[2] and Mohamed El-Ramly[3]

[1, 2, 3] Faculty of computers and information, Cairo University, Cairo, Egypt

[1]amrgalal@aucegypt.edu, [2]Drhesham2007@gmail.com, [3]m.elramly@fci-cu.edu.eg

ABSTRACT

In this paper we show that the security problem in banking environment is a severe one. Security problem becomes even more complicated within SOA projects. Criticality of the banking domain combined with the complexity that commonly exists is such huge working environment push the security problem in SOA projects to the edge. We believe that such kind of problems does not have a silver pullet. Therefore, we recommend a solution for that problem using a proposed security decision making framework. The recommended technique is proved to enhance the overall working environment security by (1) optimizing and standardizing security control selection process, (2) reducing the cost of securing new software projects and (3) speeding up securing new SOA projects. The study has been applied in banking domain but we believe it can be easily used in other domains. The architecture of the framework achieves flexibility and extendibility through using metadata and keeping the framework steps simple. The feasibility and effectiveness of the framework is proven by applying it to create a security decision making tool to automate the process of selecting the security controls needed for each new project. The results of applying the tool is compared to the traditional method and the result show clear superiority of the recommended technique.

Keywords: *SOA, Security, Software engineering, Web services, banking.*

1. INTRODUCTION

Although Service Oriented Architectures (SOA) existed many years ago, some authors like Nodehi et al [7] (2013) still consider it a new style to develop applications. While many authors like Jebna and Mahmuddin [1] believe that SOA architecture simplifies the structure of systems, others believe it is harder to develop, as our survey shows. (Discussed in section two) Although SOA is now more than a decade old, new applications of SOA still emerge, e.g. see Galal and Hassan [2]. One of the most severe problems in using SOA is the security problem. The basic framework of SOA does not have any security measures, rules or constructs. [4] This problem has many other factors that make it much more complicated in banking domain. Trying to find an ultimate solution for this problem is very difficult but relieving its severity seems much more realistic. The security problem in SOA projects, especially in banking domain and generally in huge working environments, has many reasons and effects. We will list some of these reasons and effects and will show how the proposed technique will handle each of them. The rest of this paper is organized as follows: Section two discusses SOA security problem. Section three presents the proposed SOA security decision making framework architecture and behavior. Section four describes a sample tool for SOA security decision making generated by propagating the framework with a specific set of metadata developed by a security expert. Section five shows the analysis and discussion of the value of the framework and its advantages. Finally, section six presents the conclusion and future work.

2. SOA SECURITY PROBLEM

Securing the infrastructure underlying the SOA is a major problem. [3] We surveyed 30 IT specialists working in banking domain in Egypt. More than 90% of them believe that the security problem in banking domain is a severe one. Moreover more than 90% of them also agreed that using SOA means extra security problems and more threat handling to be needed.

It is worth mentioning also that although SOA became very popular, securing it is not always done appropriately. Most of the time, securing SOA is not done by security experts, but rather by ordinary developers. [5][6]

Using SOA in banking domain helped in solving some severe problems in nearly optimum ways. For example SOA enabled money transfers between financial institutes and all its required queries without violating bank account secrecy act. Another problem is inquiring about the credit score of an individual or an entity across all banks all over the country. Some other problems like inquiring about stolen credit cards have been enhanced dramatically since applying SOA. Generally speaking the banking industry needs interaction between different databases which exist

in different places and institutes without merging them or revealing all their details and that is exactly what SOA do.
Adding to the above, we found from our experience which is more than 20 years in the banking domain, that exposing any of the bank IT components to any external parties outside the bank premises increase the risk tremendously. For example, exposing a banking component like a home banking application to the public via the Internet, multiplies the potential attacks on this component by a factor between 10 and 100. But on the other hand, a large portion of the real advantages of implementing SOA cannot be achieved unless it is exposed to the Internet. So SOA is not a silver bullet for implementing banking applications. SOA offer many advantages and disadvantages. And one of the SOA problems is the security problem which affects the entire working environment in banking domain in many ways.
We found that 22 of the surveyed IT specialists (73%) admitted that new software projects suffer delays, by at least 3 days, to get approved by the IT security department or specialist and to identify which security controls to be used to secure it. Twenty five (83%) of them agreed that the selected set of security controls may vary from one security expert to another within the same organization. They also said that the security level of similar projects may vary significantly within the same organization because they are developed by different teams and approved by different security specialists. This leads to discrepancies in the security level of different systems in the same organization. Increasing the number of times a security resources used (e.g., a firewall, a security expert, etc.) decreases the cost and increases the enterprise efficiency. [8] In our survey we found also that hiring highly skilled security specialists is very expensive and cannot be rewarding except in very large organizations that have so many projects that need security assessment. Another finding is that some security specialists use more security controls than needed. Amazingly we found during our experience that using more security controls sometimes lead to less security level. For example using two different firewalls, while doubling the cost, may not lead to doubling the security as it seems. In fact, under certain configurations, this may lead to probability of preaching that equals the absolute summation of the probability of preaching of each individual firewall.

3. FRAMEWORK ARCHITECTURE AND BEHAVIOR

To overcome the above security problems or at least to minimize their effect we propose a framework that could be easily customized to fit different working environments. Our target within that research is to solve the security problem of using SOA in banking domain. Although our study is focused on banking domain, we believe that the same framework can be used successfully in many different domains especially the ones that have common features with the banking domain, e.g. telecommunications. It is also worth mentioning that this framework when configured with new specialized metadata will generate a customized security decision making tool for the domain of the given metadata. Figure 1 shows the UML use case diagram of the behavior of the proposed framework. It is not a use case for the tool. So many use case diagrams could be produced for tools that may be created to aid applying the proposed framework. Although the proposed framework could be applied as it is without any aiding tools, it is preferred to use automated tools derived from it. First, this equips it with the necessary expert's knowledge represented in the metadata. Second, this makes applying the framework much easier and minimizes the probability of human mistakes. We followed this recommendation by preparing and using an automated tool to apply the framework easily. The actors of this use case diagram are:

- Automation developer: is responsible of creating the automation tool to hold the metadata and enabling the framework to be applied easily with minimum human errors.
- IT security officer: approves the new project from security point of view and recommends the proper set of security controls for the project.
- Security expert: creates the metadata according to the work environment and makes sure it enables the framework to perform accurately.

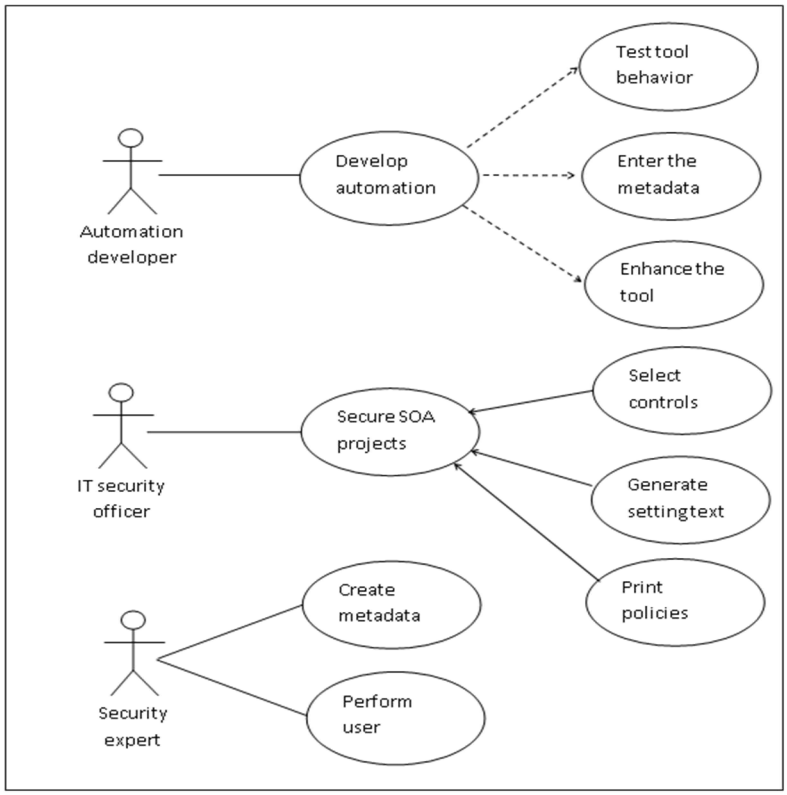

Fig. 1. Use case diagram for the proposed security framework for SOA decision making.

To illustrate the behavior and main idea of the proposed framework we need to provide the necessary definitions. The given terms are defined in the context of the proposed framework:

- **Metadata**
  The set of metadata is a set of data that describe data that describe a certain environment and users' preferred components in that environment. As we will describe in the following definitions, those components are security controls, control categories, conditions, control conditions, services, service controls and service condition answers.
- **Security Control**
  Security control means any hardware (e.g., security token) or software (e.g., implementing full access matrix) that can be used to secure a certain web service. Each security control belongs to a control category and has a priority in that category. Each control could be enough from its category or not. Priorities are set based on a number of factors, e.g., preference, availability, cost, etc.
- **Control category**
  Control category is a set of controls with a certain one or more common feature. Each category could be mandatory or not.
- **Condition:** A condition is a question with the answer of "Yes, No or NA (Not Applicable)" that affects the decision of the need to apply certain security control or not.
- **Security control conditions:** A set of conditions that describe for a specific control which situations need applying this control.
- **Service:** We mean by a service a web service that handles a specific task.
- **Service controls:** Set of controls needed to secure a specific web service.
- **Service condition answer:** It is the answer which we get if we checked this condition against that specific service.

Figure 2 shows class diagram to show the relation among entities.
The proposed framework obeys a set of rules that control and regulate its working steps and flow. Those rules could

be considered as the constitution that identifies the borders that the framework works within.

Our proposed framework applies the following rules:

1) The framework works in 2 modes namely: *normal mode* and *minimum mode*.
2) For both modes, if the control category is mandatory, then one control at least of that category should be selected for each service.
3) In minimum mode, if the control is enough from its category then no other controls from that category should be selected.
4) In minimum mode, if more than one control is enough from its category only the control with the highest priority is selected.
5) Each control condition may be a simple condition or a compound condition
6) A simple control condition can be considered as an "if statement" that means the control is needed if that condition is satisfied.
7) A compound condition may include multiple simple conditions with a weight for each one as well as a threshold for that compound condition. The compound condition can be considered as an "if statement" with the following format: "if sum (weights of simple conditions) >= the compound condition threshold then that control is needed".
8) The framework is intelligent enough to check only the minimum required number of conditions to identify the required controls for a certain list of services.

We believe that using Metadata in that framework will give high flexibility in describing the required conditions that are needed for each control. Another advantage of using metadata is enabling the framework user to build his/her own structure of controls, control categories, conditions and with his own flavor of priorities in applying security controls.

It is worth noticing that we included a set of metadata that we used to apply our framework within one of the banking domain sites. In fact, this set of metadata could be considered as a generic set of metadata to be used as a starting point for building any application specific Metadata using spiral methodology.

4. CASE STUDY: A TOOL FOR INTELLIGENT SECURITY CONTROL SELECTION

In this section we propagate the proposed framework with the suggested metadata proposed by security experts in the banking domain to secure SOA applications. This could be considered as an instantiation of a sample intelligent tool for SOA security decision making for the banking domain.

Set of metadata for a specific banking organization:

The following is the list of control categories. Each control category has list of security controls. Each list of security control is sorted according to the priority of each security control within the list. Each control has a property indicating if it is enough if selected from that category or not.

We have a list of service conditions. Each condition should be either simple or compound. If the condition is simple it will be represented by a simple if-then statement.

If the condition is compound it will be represented by a threshold and a compound question that consists of a group of simple if-then statements with a score for each of it.

For each service we should specify if this condition applies, or not, or N/A.

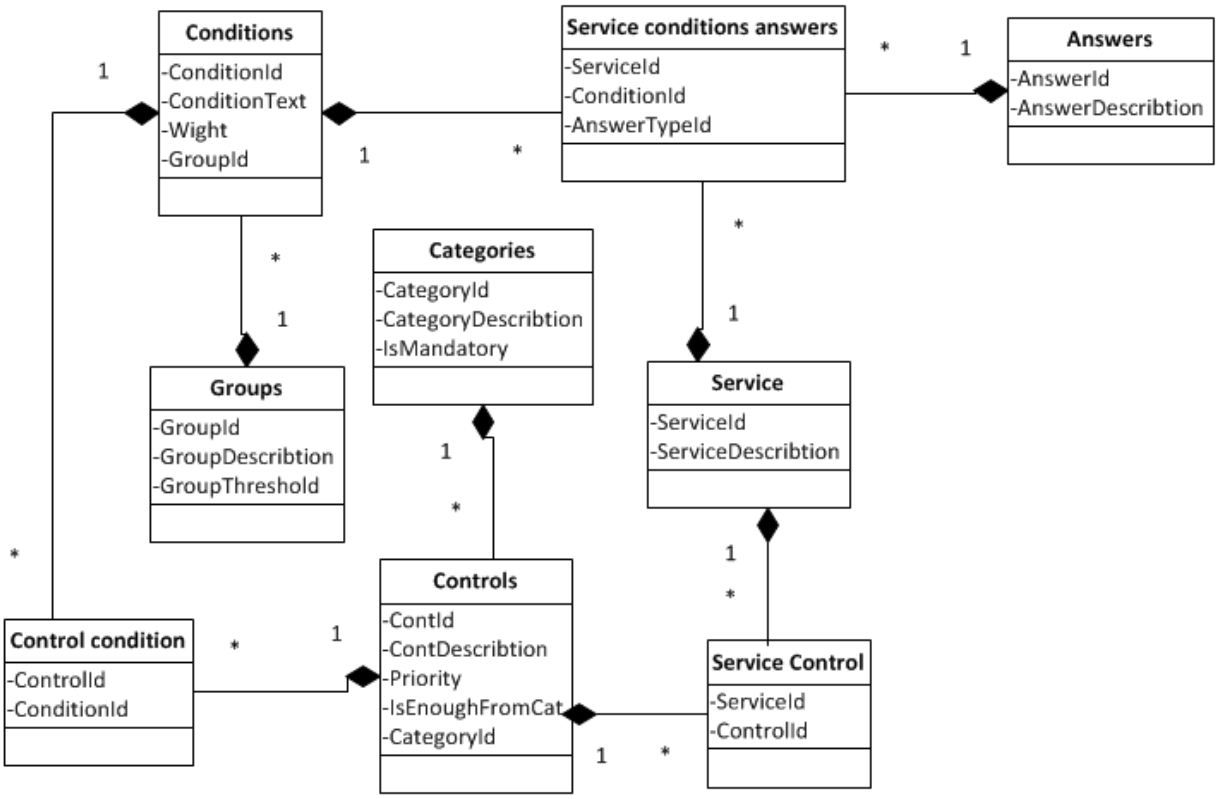

Fig. 2. Class diagram for security controls selection tool

Control categories:

1) Hardware
1) Stateful firewall
2) Stateless firewall
3) Proxy server with software firewall
4) Password (security) token
5) Intrusion prevention system (IPS)

2) Software
1) Implementing Detailed history logs for the service usage
2) Implementing full proper access matrix

3) Network
1) Site-to-site VPN
2) VPN
3) Access list with specific mac addresses

4) Policies and procedures (mandatory)
1) Clear policy to define the service usage

5) Tunneling
1) Digital certificate issued by external Certification Authority (CA) "if selected it is enough from its category"
2) Digital certificate issued by internal CA

6) Authenticating the clients
1) Binding the IP with the user
2) Password
3) One time password

Conditions:

1) Is it a service related to financial transactions? Wight = 10
2) This service could be used for internal users? Wight = 10
3) This service could be used for external users? Wight = 10
4) Is it for bank customers only? Wight = 10
5) Is it a chargeable service? Wight = 10

6) Is the service user a corporate?
Wight = 100
7) Is the service user known?
Wight = 10
8) Is the service user security level advanced?
Wight = 10
9) Is the service provider security level advanced?
Wight = 10
10) Is it critical service (cannot be stopped for more than 60 minutes)? Wight = 10
11) Always answer this question with "Yes" to be used as a stub to force all the mandatory controls for all services.
Wight = 10

Groups are just an architectural solution to represent the compound conditions. The following groups are a subset of the groups advised by the security expert to show how they can be used.

Groups:

Group1: condition1, threshold = 5

Group3: condition3, threshold = 5

Group4: condition4, threshold = 5

Group5: condition5, threshold = 5

Group11: condition11, threshold = 5

Group12: condition1 and condition5 and condition6, threshold = 101

Group13: condition6 and condition8, threshold= 101

The following is a subset of the required control conditions that show all the types of control conditions and the full list could be built easily by listing all the controls and build a list of control conditions that satisfy each of it.

Control conditions:

Use stateful firewall if group 3
Use clear policy to define the service usage if group 11
Use Password if group 1
Use Password if group 4
Use Password if group 5
Use password token if group 12
Use site to site VPN if group 13

How the framework works:

In case of minimum mode:

1) Create a set of metadata that describe the required environment.
2) Build a list of "service conditions' answers" by checking each condition against each service.
3) Select the set of mandatory "control categories".
4) For each control category in the selected set check if it has any control that is "enough from category" or not. If it has one, select it. If it has many, select the one which has the highest priority.
5) For the unique selected control list, select the ones which have a simple condition and a service condition answer which satisfy it plus the ones which have compound control conditions with service condition answers whose weight summation is more than its threshold.

In case of normal mode:

1) Create a set of metadata that describe the required environment.
2) Build a list of "service conditions' answers" by checking each condition against each service.
3) For each control select it if it has a simple condition and has a service condition answer which satisfies it or it has compound control conditions with service condition answers that have weight summation more than its threshold.

Example:
For a service called FMT which stands for (Fast Money Transfer) the answers for all conditions are as follow:

1) Is it a service related to financial transactions? (Wight = 10) ➔ the answer is Yes so the score = 10
2) Is it for internal use? (Wight = 10) ➔ the answer is No so the score = 0
3) Is it for external use? (Wight = 10) ➔ the answer is Yes so the score = 10
4) Is it for bank customers only? (Wight = 10) ➔ the answer is Yes so the score = 10
5) Is it a chargeable service? (Wight = 10) ➔ the answer is Yes so the score = 10
6) Is the service user a corporate? (Wight = 100) ➔ the answer is Yes so the score = 100
7) Is the service user known? (Wight = 10) ➔ the answer is Yes so the score = 10
8) Is the service user security level advanced? (Wight = 10) ➔ the answer is No so the score = 0

9) Is the service provider security level advanced? (Wight = 10) ➔ the answer is Yes so the score = 10
10) Can we afford to stop the service for more than 60 minutes? (Wight = 10) ➔ the answer is Yes so the score = 10
11) Use this question as a stub to get the Answer this question with yes for all services to force the controls needed for all services (Wight = 10) ➔ the answer is Yes so the score = 10

According to the above list of answers and the metadata that we mentioned we can select the following security controls:

- For group 1 ➔ Use Password if group 1 (Group1: condition1, threshold = 5) ➔ and since this service has a score of 10 in that condition it means that we have to use Password for that service.
- For group12 (Group12: condition 1 and condition 5 and condition 6, threshold = 101) ➔ since this service has a score = 120 (condition 1 = 10 + condition 5 =10 + condition 6 = 100) exceeds the threshold (101) it means that we have to "Use password token"

Implementation

We implemented this sample tool using visual basic and prorogated it with the full set of metadata mentioned above. We applied the tool to the FMT case study above and 4 other projects based on real scenarios. It is available upon request from the authors.

5. ANALYSIS AND DISCUSSION

We used the proposed framework to build an automated tool to cure many effects of the SOA security problem. The results showed that applying the proposed framework has many advantages. The following table shows some of the most important advantages of the proposed framework by comparing the status before and after applying the proposed framework. It is not a must that all the listed problems exist in one working site but Table 1 shows the effect on each problem if existing.
The framework is capable of selecting security controls as good as the security expert who prepared the metadata used with it. That means we may achieve advanced security decision making with low cost and low expertise if we used the proposed framework in a large scale. Achieving large enough scale could be done via convincing supreme organizations like central banks.
For all the previously mentioned reasons we created the proposed framework to standardize the process of selecting the security control all over the organization, reduce the time needed to launch new projects, reduce the IT security cost and achieve better IT security across large organizations. This framework can be considered as a cookbook that simplifies selecting the right combination of security controls without the need for strong security background. The framework could be used conceptually without any other aid tools. But it is recommended to use it via an automated tool to minimize mistakes and to guide its usage. Obviously using automated tools will simplify applying the framework and minimize users' mistakes. We created a simple friendly tool with GUI to prove the correctness of the framework behavior and to simplify its application. It is worth mentioning that according to our study that included 5 projects securing any new project consumed less than 3 minutes, assuming that the tool was supplied with the metadata. On the other hand, the minimum meeting time with security expert to consult him about securing new project consume at least 60 minutes for each project plus one working day delay on average to meet him. Plus other delays for administrative activities. With a total of 3 working days on average to get the new project approved. We concluded that the time and efforts spent in building the framework and the automation tool worth its benefits since the saved time and efforts is tremendous. Other benefits include minimizing the number of used security controls, standardizing the security level across projects and reducing the time to launch new projects.
Security decision making framework can be used to solve many of SOA security problems or at least to reduce its severity. Building automated tools could be a useful method to formalize this framework and proof its effectiveness. Another advantage of using automation tool is to ease the framework usage.
In this research we concentrated on applying the framework in securing SOA new projects but it can be used in so many other situations.

6. CONCLUSION AND FUTURE WORK

Depending on the above we conclude the following:

1. Decision making frameworks could be used to reduce the severity of the SOA security problem in banking domain.
2. Using decision making frameworks not only enhances the security problem in many ways, but also has many other benefits, most notably reducing overall organization

security cost and minimizing the time needed to launch new projects.

This work could be complimented in the future by the following:

1. Apply the proposed framework in other domains, for example in telecommunications domain.
2. Recommend techniques to customize metadata for specific working environment.
3. Extending the framework to address more SOA problems.

Table1: Advantages of Using the Proposed Security Decision Making Framework

No.	The problem	Status before applying the framework	Status after applying the framework	Notes
1	Weak security standards across the organization.	If the same project is analyzed by more than one security specialist each one may select different set of security controls according to his opinion.	If the same project is analyzed by more than one security specialist using the same set of metadata all of them will select the same set of security controls.	This affects problems No. 4 & 6
2	Security specialists are not aware of all work-around techniques done by other specialists in the same organization.	Security specialist should go through each and every new project case by case to be aware by each new work-around.	The metadata updated each time a new work-around applied and from that time on the new work-around is used by the framework whenever needed.	This affects problems No. 3, 4, 6 & 7
3	Delay in approving the new projects from IT security department.	New projects need at least 3 working days on average to get approved.	New projects are approved within the same day.	This affects also problem No. 7
4	Security level varies from project to project within the same organization.	Majority of the surveyed IT specialists agreed that the level varies in practice.	Majority of the surveyed IT specialists agreed that the level does not vary. Since the security level or all projects depends on the quality of the metadata.	This affects also problem No. 6
5	Some security controls may be used although they are not needed.	In our experience, by reviewing a set of projects we found indeed that some security controls are used although they are not needed.	By applying the framework redundant security controls are removed and the utilization of these controls is optimized. Obviously this depends on the quality of the metadata.	This affects problems No. 6 & 7
6	Weak security level for new SOA projects.	Majority of the surveyed IT specialists agreed that the level of security after applying the framework is better than before applying it. They were presented with the framework and its security controls' decision making process.		
7	High cost of securing new projects.	Majority of the surveyed IT specialists agreed that the cost of securing new SOA projects is high.	The cost is reduced because of reducing the number of working hours and removal of the redundant security controls.	

REFERENCES

[1] Abdulkarim Kanaan Jebna, Massudi Mahmuddin, "Decomposing of SOA on File Transfer of Web Service on Windows and Mobile Environments", Academic Research International, 2013.
[2] Amr Galal, Hesham Hassan, "Creating enhanced Wikileaks", International Journal of Information & Network Security (IJINS), Vol.2, No.2, April 2013.
[3] Anu Soosan Baby, Deepu Raveendran, Aswathy Josephine Joe, "A Study on Secure and Efficient Access Control Framework for SOA", International Journal of Computer Science and Telecommunications, Volume 3, Issue 6, June 2012.
[4] Deepti Sisodia, Lokesh Singh, Sheetal Sisodia, "WEB BASED SECURE SOA", International Journal of Computing Algorithm, Vol 01, Issue 02, December 2012.
[5] Muhammad Qaiser Saleem, Jafreezal B. Jaafar, Mohd Fadzil Hassan, "Domain-Specific Language for Modelling Security Objectives in a Business Process Models of SOA Applications", Advances in information Sciences and Service Sciences(AISS), volume 4, number 1, January 2012.
[6] Muhammad Qaiser Saleem, Jafreezal Jaafar, Mohd Fadzil Hassan, "secure business process modeling of SOA

applications using 'UML-SOA-SEC'", International Journal of Innovative Computing, Information and Control, volume 8, number 4, April 2012.
[7] Tahereh Nodehi, Sudeep Ghimire, Ricardo Jardim-Goncalves and Antonio Grilo," On MDA-SOA based Intercloud Interoperability framework", CMSS - Volume I, Issue 1, 2013.
[8] Ying-Hong Wang, Jingo Chenghorng Liao, "Design of SOA Integration for 3C Distribution Channel", Wseas Transactions on Information Science & Applications, Volume 8, 2011.

AUTHOR PROFILES:

Amr Galal is an Egyptian researcher, he hold M.Sc. and post-graduation diploma in computer sciences from Cairo University, B.Sc. of accounting from Cairo University. Amr has been published many research papers in scientific journals and conferences. He is also holding many professional certificates from many organizations like GSEC and GWAPT from SANS, OM from Oracle and NCC certified. Amr interests are Information and network security, software engineering, reusability and Service Oriented Architecture (SOA).

Hesham Hassan is an Egyptian researcher born in Cairo in 1953. Hesham's educational background is as follows: B.Sc in Agriculture, Cairo University, Egypt in 1975. Postgraduate diploma in computer science, from ISSR, Cairo University, Egypt in 1984. M.Sc in computer science, from ISSR, Cairo university, Egypt, in 1989. PhD in computer science from ISSR, Cairo University (dual supervision Sweden/Egypt) in 1995. He is now a PROFESSOR and HEAD of computer science department at the faculty of computers and Information, Cairo University. He is also IT Consultant at Central Laboratory of Agricultural Expert System, National Agricultural Research Center. He has published over than 51 research papers in international journals, and conference proceedings.
He has served member of steering committees and program committees of several national conferences. Hesham has supervised over 27 PhD and M. Sc theses. Prof. Hesham interests are Knowledge modeling, sharing and reuse, intelligent information retrieval, Intelligent Tutoring systems, Software Engineering. Cloud Computing and Service Oriented Architecture (SOA)

Mohamed El-Ramly is an assistant professor of computer sciences at the Faculty of Computers and Information, Cairo University. His main research area is software engineering, and specially software evolution, reverse engineering and reengineering. He hold PhD in computer sciences from University of Alberta, 2003, Canada, M.Sc. and Diploma of operations research, 1996 and 1992 from Cairo University and B.Sc. of computer engineering from Ain Shams University, 1990. Previously Mohammad worked as lecturer at University of Leicester, 2003-2007. Mohammad has over 20 publications in the area of software engineering and reverse engineering.