A Model for Implementing Security at Application Level in Service Oriented Architecture

Said Nabi, Saif Ur Rehman
Department of Computer Science
Shaheed Zulifikar Ali Bhutto Institute of Science and Technology (SZABIST)
Saidnabi115,saifi.ur.rehman{@gmail.com}

Simon Fong
Department of Computer and Information Science
University of Macau Taipa, Macau SAR
ccfong@umac.mo

Kamran Aziz
Abasyn University, Islamabad Campus
Islamabad, Pakistan
kamrandik@gmail.com

***Abstract*— Securing the communication channels only, cannot guaranty end-to-end security in SOA based systems. To provide complete security, there is need to provide security at application level for SOA based systems. But it is a great challenge for the developer of the web services to implement the security during development of the web services. In this paper we have proposed a model, which will automate the generation of security policies for web services. This system will facilitate and enable the developers of the web services, to generate and implement security policies during the development of the web services, without having intensive knowledge of the security domain and the underlying system. The proposed system will also make the application level experts independent of the security experts for the generation and implementation of the security policy for the development of the web services.**

***Index Terms*—Web Services Architecture (WSA), Service Oriented Software (SOS), Service Oriented Computing (SOC), Service Oriented Architecture (SOA), Application Level Security in SOA, SOA security policy.**

I. INTRODUCTION

Requirement engineering is considered as one of the most critical phases in software engineering, specifically, in software design and development. If errors are introduced at the requirement stage, then they remain undetected till the later stages of software development process and [1]. Requirement engineering addresses the issues of requirement collection to design and develop the desired software. Requirement engineering has a direct impact on all the stages of software development including software design, architecture, implementation, testing and deployment. Software architecture deals with design and development of the abstract level structure of the software. It consists of a number of architectural elements like components and connectors, which are assembled in such a way to satisfy the functional and performance requirements [2]. There are a number of architectural styles or patterns used by the software architects including layered systems, event-based, object-oriented, data-abstraction and implicit invocation, etc. Although these styles provide sufficient space of architectural choices to the architects, but alongside pose challenge for the architects to realize the tradeoffs while selecting the best suitable style in a particular situation and environment [3].

There is a remarkable difference among the development of traditional software and SOA based application, especially at design and analysis phase. SOA application requires undergoing analysis and design phases. Analysis phase produces candidate services from the business requirements. Business analysts and service architects emphasize on the usage of standards to refine the candidate services. Therefore, the formal definition of business processes is very important in SOA. The testing and development phase of SOA, however, are similar to the traditional development processes [4].

SOA is a set of services which are the collections of software component and carry out business process independently. Services must have the properties like loose-coupling, self-containment and should have well defined independent interfaces. Self-containment of the services mean it would be able perform their functionality independent of the other services. Loose-coupling mean services communicate with each other through sending messages and are not aware of technical details of other collaborator services [5]. The flexible and modular designs of software applications have a negative impact on the security of software applications [6]. SOA design

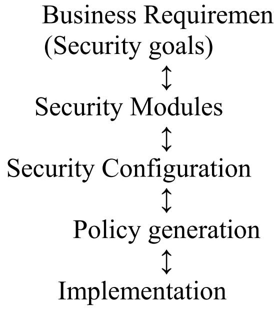
doi:10.4304/jetwi.6.1.157-163

principles provide guidelines for developing interoperable and agile service logic, which includes composition, discoverability, statelessness, independence, loose-coupling, abstraction, and reusability and service contract [7].

Although the information security standards like identification, authentication, authorization, confidentiality, integrity and availability have same connotation in SOA based applications, but the agile implementation of SOA makes it difficult to ensure secure SOA implementation within the organizations. There is a lesser possibility for successful implementation of secure SOA without specific information security guidelines [8]. There is a need for web service architecture that can control all the levels of web service stack which may include: describing, registering, managing, monitoring, deploying, wrapping, and discovering fundamental software components to self-repeatedly composing software [11]. The QoS attributes like performance, scalability, accounting and security plays a vital role for web services of different business domains [14]. The reputation and trust of data intensive services in cloud computing have generated important issues e.g., trust evaluation in collaboration services becomes a challenge. To represent service collaboration for data intensive services, the concept of service collaboration graph has been used in [15].

Securing the communication channels cannot guaranty end to end security in SOA based systems. Application level security plays a vital or key role to enable end to end security implementation in SOA. To deal with SOA security at application level, there is a need for expertise of two domains like security domain expertise and application domain expertise. There is a very rare possibility that a single person is equipped with both expertise in security domain and application domain. It is a great challenge for the developer of the web services to implement the security modules during development of the web services. To handle this issue, different researchers has proposed various techniques to facilitate the application domain experts to implement security standard for web services during its development.

Next section provides an abstract view that business and security requirements in the form of security goals are mapped into security policies, which will be implemented by the web services developers.



The above figure provides some basic idea to convert security relevant business requirements into security polices, which can be easily implemented by the web services developers. First the security goals for the business are identified, then these requirements are categorized into different security modules like identity management, confidentiality, Integrity, authentication and authorization. Next step is the configuration of above modules by applying different approaches like pattern oriented etc. further these patterns will be used by the application to generate the security policies in an automated way. Finally generated polices will be implemented by application developer for web services.

In this paper we propose a model, which will automate the generation of security policies for web services. This system will facilitate and enable the developer of the web services, to generate and implement security policies during the development of the web services, without having intensive knowledge of the security domain and the underlying system. The proposed system will make the application level experts independent of the security experts for the generation and implementation of the security policy for the development of the web services.

Previously some of the researchers has worked on the automatic generation of security polices and provided some basic information to enable the developer of the web services, to develop and implement security policies without having intensive knowledge of the security domain and the underlying system. These approaches have a number of limitations and are not mature enough to be implemented. [12] has proposed a model driven technique to generate the security policies for web services which can be easily implemented by developers of web services. But their proposed system still need for a security expert to provide some basic information about the security domain and help in the enforcement of a particular security policy to achieve the security goals for the web services. Similarly architecture for the security advisor has been proposed by [13]. It consists of an algorithm, which selects a pattern and generates an enforceable policy for given security requirements. In other word it facilitates and enables the developers of the web services to choose the security goals, generate and implement security policies with having complete knowledge of the underlying system. But proposed architecture is still depended on the security experts to capture the basic security knowledge into security pattern and application needs to be preconfigured by the security expert. To generate the security policies, it uses the Apache axis2 as application server for web services which has some limitations like complexity from user point of view and not full support for JAX-WS. But comparatively we will use the Apache CXF for enforcing standards for policy generation. Usage of Apache CXF has the following advantages over the Apache axis2.

The rest of this paper is organized as follows: the literature survey on application level security in SOA is provided in section II. A proposed system is discussed in section III. Evaluation of the proposed system is provided in section IV. Finally, we conclude in section VI along with potential future directions to this research.

II. Literature Review

According to [9] Security requirements models are used by Model driven security (MDS) approach at abstract level to generate security policies automatically. These policies are used for security services configuration. The MDS solutions focus on a single security pattern for each security requirement. This is not enough because the current Cloud and SOA services are distributed among different heterogeneous security domains and continuously changing infrastructure. It need multiple security pattern support for single security requirement. To present such security services whose configuration support for different security patterns is still a challenging task. To solve this problem author has presented a framework that adds and integrates a new layer "security pattern refinement layer" to existing MDS layers. The security pattern refinement layer helps to configure a single security service with various patterns. This framework has been properly validated by applying on healthcare system. But the proposed system is still dependable on the security domain experts and not clearly mentioned how to automate the generation of security artifacts/policies.

Menzel et al. [10] has found that although to represent security requirements as Security policies can insure the flawless usage of services. But as specification of the web services are complex, so it is a tricky and error prone task to practically implement the security policies. The proposed system will facilitate the mapping of architectural models representing simple security goals into security policies by Model driven technique. Security configuration patterns of the web services play major role in mapping process. The SOA meta model of the security configuration patterns provide a foundation to describe the objects and their association at the modelling layer. This system will also provide a formal pattern structure and to specify these pattern(s), a domain specific language was used.

Imamura et al. [8] has stated that the current tools used for the configuration of security assets of the web services presents a technology aspect, where user must fill the gap between configuration and security needs manually. This leads to misconfiguration problem and extra configuration costs. Author introduces the SOA and MDA and presented a framework for polishing the user's security requirements by using the idea of SOA and MDA. In this framework the users describe their requirements using the given vocabulary list and then transform these requirement step wise into details requirements. After getting sufficient details of the security requirements, the users select the countermeasures and transform them into a required level of details. Using a policy language the users illustrate these details and uses best practice patterns for linking the gap between these levels. The future work includes extending the tool and framework and also developing an easy to use way to select an appropriate pattern from those currently available.

Some of researchers have presented techniques used to automate the security policies generation e.g., [12] has proposed an idea of meta model driven security mechanism to ease the task of web services developers to implement security in the web services. Security meta model give a base for exchange of information and model interactions. These also illustrate the fundamental objects, associations and related roles in service oriented architecture. But it is still dependent on the security domain experts to generate and configure all the security patterns and there is no way to dynamically add and configure new security patterns.

Architecture for the security advisor has been proposed in [13]. It consists of an idea and workflow for generating security policy automatically for web services, which selects a pattern and generates an enforceable policy for given security requirements. In other word it facilitates and enables the developers of the web services to choose the security goals and configure the security module without having complete knowledge of the underlying system. But proposed architecture is still depended on the security experts to capture the basic security knowledge into security pattern and application needs to be preconfigured by the security expert. To generate the security policies, it uses the Apache axis2 as application server for web services which has some limitations like complexity from user point of view and not full support for JAX-WS. Usage of Apache CXF has the following advantages over the Apache axis2.

i) CXF uses standard API's while axis2 in general uses proprietary things.
ii) CXF is more responsive to the users issues and ensure the quick availability (release fixpacks every month or after 2 months) of the "fixpacks" to the user, while axi2 has least compatibility for older version and slow (user have to wait from 9 or 10 months to get complete patches) response to the user issues .
iii) Apache CXF is well advised for those who use Spring framework because CXF have a better integration for spring framework. CXF is considered as more embeddable into other applications.
iv) Performance wise both apache axis2 and apache CXF are comparable but in case, when you use JAXB API's then CXF is faster than axis2.

III. Proposed System

In this section we propose a model, which will help to automate the generation of security policies for web services. This system will facilitate and enable the developer of the web services, to implement security policies during the development of the web services, without having intensive knowledge of the security domain and the underlying system. The proposed system will make the application level experts independent of the security experts for the generation and implementation of the security policy for the development of the web services.

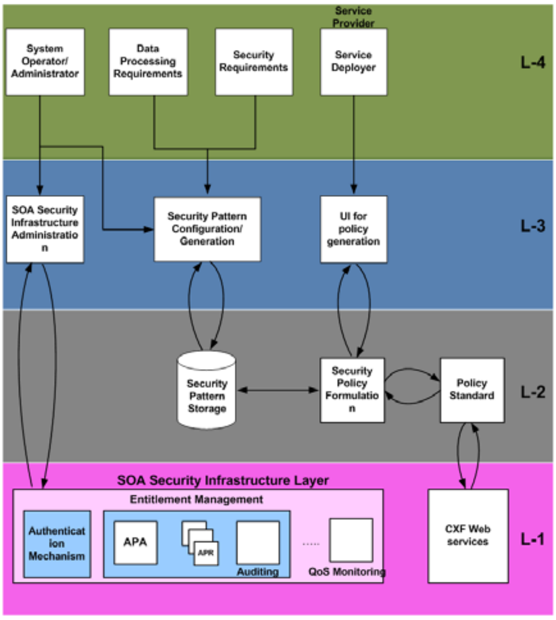


Figure 1: Architecture of the proposed system

The above figure shows the basic architecture of our proposed system, which is partitioned into four different layers including, security infrastructure layer, Business Logic Layer, Processing Layer and Interaction or application layer.

A. Infrastructure Layer

SOA infrastructure security has vital role in providing security to the web services. Infrastructure layer includes authentication, access-policy administration, access policy resolution and auditing etc. Enforcement of access policy is unluckily access policy resolution, auditing and administration are owned by different people other than developers of the business logic, changes at different times and different rates. The main benefit of separating infrastructure layer from the business logic layer is that, in case of any compliance or change in the security requirement, to adopt modification in access policy without need in the business logic of the web services. For example, consider the following scenario of Order Management of the Component Service, in which access policy for the above scenario (Component Order Management Component Service) are described as:

i) Order can only be entered by those people, who have a “broker” role,

ii) Order can only be updated by the owner or a manager of the owner of the order and

iii) Order can only be read by the owner, manager of the owner or a subject of the owner with the role of “reconcile”.

In the given scenario the task of the order management development, the developers are required to highly focus on how to implement the order management services most efficiently. The resolution of access policy requires accessing the proper contextual information. The policy resolution shows wither to deny or permit the given request, which is enforced by the policy enforcement according to the decision on the request. To provide effective and practical SOA security infrastructure should allow distributed access-policy resolution by using various distributed objects of the resolution service, because the centralized access policy resolution has some scalability, availability, and performance related issues.

The auditing of access policy is needed for composite service and component service as well or the business process which invokes the component services and auditing is especially valuable for SOA based environment. There are very controlled and limited use of the application functions in non SOA based environment. On the other hand SOA based environments, the services will may called from very unpredictable and diverse ways. Auditing is a critical tool to predict issues, before they occur and locate the basic reason of problems when they do occur. Infrastructure layer also consists of application server for web services which is named as apache CFX. Apache CXF provide and enforces standards for policy generation.

B. Business Logic Layer

Business logic layer is the second layer of proposed system and consists of: security pattern repository, security policy formulation module and security policy standard provider and enforcer:

i) Security pattern repository is used to store different security patterns generated at the processing/operational layer. It communicates with security pattern generation and configuration module and on the other side it will communicate with security policy formulation module by providing the required security pattern(s).

ii) Security formulation module: this module plays a key role in the security policy creation and is the major focus of this research. After data processing requirements and security requirement has been captured in the security patterns and refined, the security formulation module is used to generate enforceable policy based and knowledge captured in the security patterns.

iii) Policy standard component of business logic layer provide and help in enforcement of security policy standards.

The detailed work flow of the security policy formulation module is shown in Figure 2.

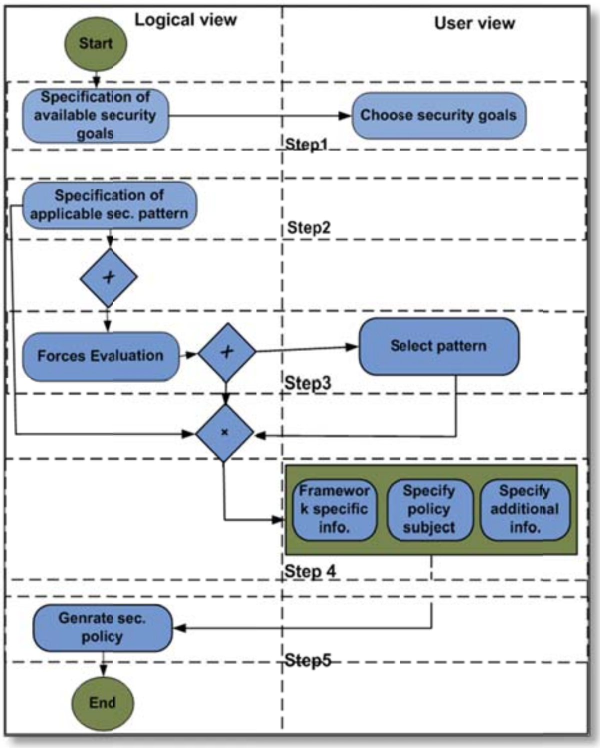


Figure2: Working details of the Security policy Formulation module

The above figure shows a model of the working flow of application for security policy formulation and is categorized into six major steps. In the first step, application will perform full search to identify the available security goals/challenges. In this step, the users of the application will select the required security goals like confidentiality, integrity, authentication etc, based on the requirements captured from business experts and requirement engineers. User can select more than one goals based on their need for the corresponding web services development. In the second step, based on the user selection of the required goals, the application will identify all those security pattern(s), which can achieve or helpful in achieving the required goal(s). If there were more than one relevant security patterns applicable to a particular security goal, then this process will enter to the forth step, otherwise it will be forwarded for policy generation. Third step is performed only when there are several approaches (in terms of security patterns) are available, then the security policy formulation module will be use the forces attributes (constraints) to select the most suitable security pattern for the security policy generation. After selecting the most relevant and highly applicable security patterns, next step of the security policy formulation module mostly concern with users of the module or application and before going to allow the security policy formulation application to generate the security policy for the selected security goals, the users are required to enter some additional information including, specification of the corresponding framework, provide information about policy subject, and some other information to further refine the policy generation process. The availability of the specific information about the underlying framework, where the generated policy will be implemented, is very important. The reason is that, it is not necessary and even not possible that each framework will be able to support the same security patterns i.e to provide confidentiality for communicating messages, using encryption techniques and different security pattern define separate encryption methods like a one security patterns uses AES-128 encryption standard and other uses security pattern uses simple DES technique for encrypting messages. Similarly it is not possible for all frameworks to support AES-128 encryption techniques. In the last step the user will generate security policy.

C. Processing/Operational Layer

Processing/operational layer is the third layer of our proposed system. This layer acts as an interface between interaction layer and Business logic layer. It has three major Components including, SOA security infrastructure administration view, security pattern generation and configuration and User Interface for security policy generation. The SOA security infrastructure administration view is used by the local domain administrator to provide relevant security information of the underlying infrastructure and to recognize and specify all relevant information to develop the enforceable security policies. Security pattern generation and configuration part of processing/operational layer is used by the system operator and application domain expert to generate and configure security patterns based on the knowledge captured from business process experts and requirement engineers. Security pattern generation and configuration module also interact with security pattern storage to confirm the existence of a particular security pattern. Before generation and configuration of a new security pattern, it will first interact and scan completely the security pattern storage to insure that wither the required pattern is already exist or any pattern can reused to generate and configure new pattern(s). After proper configuration of security patterns and underlying security infrastructure for achieving the required security goal the developer of the web services will use security policy formulation User Interface to generate the required security policy.

D. Interaction Layer

Interaction layer is the top most layer of our proposed system, where developers or system operators and service deployer interact with application. In this layer, data processing requirements are collected from business analyst and business process experts and security requirements provided and analysed by the requirement engineers for further use.

IV. Comparison-based Evaluation of Proposed System

To generate the security policies for web services, here we present different approaches initiated by other researchers. These research initiatives mostly concentrate on meta model, model driven and pattern oriented techniques to generate security policies. Few of these researcher has focus on idea, to automate the policy generation process and has provided some basic information and future plans.

Menzel and Meinel, has proposed a meta-model based technique to generate the security policies for web services which can be easily implemented by developers of web services. But their proposed system still need for a security expert to provide some basic information about the security domain and help in the enforcement of a particular security policy to achieve the security goals for the web services. The proposed system does not provide complete information about how to generate and implement web services security polices with least required effort and lack of discussion of the automatic generation of security policies.

The system proposed by Menzel et al. [10] will facilitate the mapping of architectural models representing simple security goals into security policies by Model driven technique. Security configuration patterns of the web services and their Meta model provide a foundation to describe the objects and their association at the modelling layer. This system will also provide a formal pattern structure and to specify these pattern(s), a domain specific language was used. On the other hand for the proposed system, to secure web services, possibly there is a need for knowledge of the domain experts to devise a proper plan. Pattern engine is at initial stage, providing limited functionality and need further enhancement. The proposed system is unable to provide full information about how to generate and implement web services security polices and lack of discussion of the automatic generation of security policies.

Similarly architecture for the security advisor has been proposed by Schnjakin et al.[13]. The security advisor is an idea of an application which will enable automated generation of security policies. It selects a pattern(s) pre-configured by the security domain experts and generates an enforceable policy for given security requirements in an automated way. But on the other side the proposed architecture is still depended on the security experts to capture the basic security knowledge into security patterns and application needs to be preconfigured by the security experts. There is a possibility to have a new pattern(s) which is required to achieve a certain security goal or handle a security challenge(s), to generate and configure this new pattern, there is need to consult with security experts. In the proposed system a workflow for the automation of security policy has been presented without providing any proper algorithm or interface. To generate the security policies, it uses the Apache axis2 as application server for web services which has some limitations like complexity from user point of view and not full support for all plate forms like JAX-WS.

Our proposed system will not only combine most of the functionality provided by the system presented by Menzel et al.[10], Menzel and Meinel [12], and Schnjakin et al.[13], but also proposed some additional changes to enhance the SOA security policy generation and to automate this process. The proposed system will not be depended on the security domain experts for capturing security related information in to the security patterns and their configuration. New pattern can be generated and configured as per requirement easily and dynamically. Further comparatively we will use the Apache CXF for enforcing standards for policy generation. Apache CXF will solve the issues faced while using apache axis2 like multi-plateform support and will reduce the complexity from usability point of view. This system will generate security policy in both machine readable format like XML and normal text form.

V. Conclusion and Future Work

Due to the agile nature of SOA, it is difficult to ensure the secure implementation of SOA within the organizations. Most of the techniques proposed in the literature focus on the use of security patterns, which provide some basic concept and understanding about the security domain needed for the web services developers. But, it is still a challenge for the developers of the web services to implement security modules for the web services.

In this paper we have proposed a model, which will enable the developers of the web services to automatically generate the security policies for web services. This system will also facilitate the developer of the web services, to implement security policies during the development of the web services, without having thorough knowledge of the security domain and the underlying system. The proposed system will make the application level experts independent of the security experts for the generation and implementation of the security policy for the development of the web services. In future, we intend to look into the possibility of further refinement/enhancement of the automatic generation of security policy so that developers can implement security policies on the fly.

References

[1] A. Chakraborty, M. K. Baowaly, A. Arefin, A. N. Bahar. "The Role of Requirement Engineering in Software Development Life Cycle", Journal of Emerging Trends in Computing and Information Sciences, ISSN : 2079-8407 , Vol. 3, No. 5, pp: 723-729, 2012.

[2] P. Kruchten "The 4+ 1 view model of architecture. Software", IEEE, 12(6), 42-50, 1995

[3] D. Garlan, & M. Shaw, "An introduction to software architecture," 1994.

[4] N. A. Delessy, "A Pattern-Driven Process for Secure Service-Oriented", In Workshop on Security in Object-oriented Systems, Florida Atlantic University. Vol. 70, p. 79, 2008.

[5] D. T. Sanders, J. A. Hamilton Jr., & R. A. MacDonald, "Supporting a service-oriented architecture, "Proceedings of the 2008 Spring simulation multiconference. Society for Computer Simulation International, 2008

[6] N. Delessy and E. B. Fernandez. "A pattern-driven security process for SOA applications," Proceedings of the 3rd Int. Conf. on Availability, Reliability, and Security (ARES 2008).Barcelona, Spain, 2008

[7] J. Chetty, M. Coetzee, "Towards An Information Security Framework For Service-oriented Architecture," IEEE 2010.

[8] T. Imamura M. Tatsubori Y. Nakamura, C. Giblin,“ Web Services Security Configuration in a Service-Oriented Architecture,” WWW, pp. 1120-1121.ACM, 2005

[9] G. D. M. H., Menghwar, Depar, A. A. Jalbani, W. M. Mashwani, (2012). Security modeling for service-oriented systems using security pattern refinement approach. Software & Systems Modeling, 1-24.

[10] M. Menzel, R. Warschofsky, C. Meinel, (2010, July). A pattern-driven generation of security policies for service-oriented architectures. In Web Services (ICWS), 2010 IEEE International Conference on (pp. 243-250). IEEE.

[11] Y. Baghdadi, “A metadata for Web services architecture: A framework for service- oriented software development,” GCC Conference & Exhibition, 2009 5th IEEE Issue, On page(s): 1 – 6. March 2009.

[12] M. Menzel and C. Meinel. “A security meta-model for service-oriented architectures,” In Proc. SCC, 2009.

[13] M. Schnjakin, M. Menzel, and C. Meinel. “A pattern-driven security advisor for service- oriented architectures,” Pro 6th Workshop SWS (in conjunction with 16th ACM CCS),ACM Press, Chicago, USA, pages 13–20, 2009.

[14] A. Wahl, B. Hollunder, V. Sud and A. Al-Moayed. Quality Attributes for Web Services: A Model-based Approach for Policy Creation. International Journal On Advances in Software, 5(3 and 4), 166-178, 2012.

[15] L. Huang, S. Deng, Y. Li, J. Wu, J. Yin and G. Li. “A Trust Evaluation Mechanism for Collaboration of Data-Intensive Services in Cloud”, Appl. Math, 7(1L), 121-129, 2013.

TABLE 1

COMPARISON OF OUR SECURITY MODEL FOR POLICY GENERATION WITH EXISTING SECURITY POLICY MODELS

Security Meta-Model for SOA[12]	**Pattern Driven Security advisor for Policy generation[13]**	**Pattern driven generation of security policy in SOA[10]**	**Our Security Model for policy generation**
Meta-Model Based	Pattern Driven	Model -cum-Pattern driven approach	Pattern Based
Define basic -Entities-Relationships -Roles Associated	Define security advisor(application)	Define -Transformation of architectural model into security policies	Define -security policy formulation module
Process Requirements includes Business requirements -Security requirement	-	Process Requirements includes -Security requirements in the form of simple security intensions.	Process Requirements includes -Business requirements -Security requirement
	Security domain knowledge provided by the security experts	Possibly there is a need for knowledge of the domain expert to make a proper strategy to secure web services and other resources	Security requirements and Business requirements are directly taken from -Business process experts -Requirement engineers
Capture process requirements into: Meta Models	Capture security domain knowledge into: -security patterns	Capture security domain knowledge into: -Security policy model using security modelling language like secureSOA	Capture business and security requirements into: -Security patterns
Dependent on:- Security domain Expert	Dependent on:- Security domain Expert	-	Do not dependent on: -security expert
Application pre-configuration by the security experts is needed	Application pre-configuration by the security experts is needed	Security configuration pattern provide expert knowledge on web services	-
	-	-	Directly generate/configure new security pattern(s) based on requirements
	Apache Axis2 web application server is used	-	Apache CXF web application server is used
Provide mapping to WS Policy and WS Security Policy	-	Transform simple security intentions into security policies.	-
Informal stated security patterns	Informal stated security patterns	Introduced a formalised pattern structure of security patterns.	Using pattern engine [10]
	Generate policy in XML form	-	Generate Security policy in both XML and normal Text form

Call for Papers and Special Issues

Aims and Scope

Journal of Emerging Technologies in Web Intelligence (JETWI, ISSN 1798-0461) is a peer reviewed and indexed international journal, aims at gathering the latest advances of various topics in web intelligence and reporting how organizations can gain competitive advantages by applying the different emergent techniques in the real-world scenarios. Papers and studies which couple the intelligence techniques and theories with specific web technology problems are mainly targeted. Survey and tutorial articles that emphasize the research and application of web intelligence in a particular domain are also welcomed. These areas include, but are not limited to, the following:

- Web 3.0
- Enterprise Mashup
- Ambient Intelligence (AmI)
- Situational Applications
- Emerging Web-based Systems
- Ambient Awareness
- Ambient and Ubiquitous Learning
- Ambient Assisted Living
- Telepresence
- Lifelong Integrated Learning
- Smart Environments
- Web 2.0 and Social intelligence
- Context Aware Ubiquitous Computing
- Intelligent Brokers and Mediators
- Web Mining and Farming
- Wisdom Web
- Web Security
- Web Information Filtering and Access Control Models
- Web Services and Semantic Web
- Human-Web Interaction
- Web Technologies and Protocols
- Web Agents and Agent-based Systems
- Agent Self-organization, Learning, and Adaptation
- Agent-based Knowledge Discovery
- Agent-mediated Markets
- Knowledge Grid and Grid intelligence
- Knowledge Management, Networks, and Communities
- Agent Infrastructure and Architecture
- Agent-mediated Markets
- Cooperative Problem Solving
- Distributed Intelligence and Emergent Behavior
- Information Ecology
- Mediators and Middlewares
- Granular Computing for the Web
- Ontology Engineering
- Personalization Techniques
- Semantic Web
- Web based Support Systems
- Web based Information Retrieval Support Systems
- Web Services, Services Discovery & Composition
- Ubiquitous Imaging and Multimedia
- Wearable, Wireless and Mobile e-interfacing
- E-Applications
- Cloud Computing
- Web-Oriented Architectrues

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the "Call for Papers" to be included on the Journal's Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. "Special Issue: Selected Best Papers of XYZ Conference".
- Sending us a formal "Letter of Intent" for the Special Issue.
- Creating a "Call for Papers" for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jetwi/.

(Contents Continued from Back Cover)

SURVEY PAPERS

REGULAR PAPERS