International Journal of Innovative Computing, Information and Control
Volume **8**, Number **4**, April **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 2729–2746

SECURE BUSINESS PROCESS MODELLING OF SOA APPLICATIONS USING "UML-SOA-SEC"

Muhammad Qaiser Saleem, Jafreezal Jaafar and Mohd Fadzil Hassan

Department of Computer and Information Sciences
Universiti Teknologi Petronas
Tronoh 31750, Perak Darul Ridzuan, Malaysia
qaiser_saleem73@hotmail.com; {jafreez; mfadzil_hassan}@petronas.com.my

Received October 2010; revised April 2011

Abstract. *Nowadays enterprises are implementing their WIS through SOA using Web services. They are using MDA principles for design and development of WIS and using UML as a modelling language for business process modelling. Along with the increased connectivity in SOA applications, security risks rise exponentially. Security is not defined during the early phases of system development and left onto the developer. Properly configuring security requirements in SOA applications is quite difficult for developers because they are not security experts. Furthermore, SOA security is cross-domain and all required information is not available at downstream phases. Moreover, focus of the currently available security standards and protocols is technology; they do not provide high level of abstraction. Furthermore, a business process expert, who is the actual stakeholder of the business process model is unable to specify security objectives due to lake of security modelling elements in general purpose modelling languages like UML. As a result, he/she either ignores the security intents in his/her model or indicates them in textual way. We are fostering the specification of security intents at high level of abstraction by presenting a security intents DSL containing the essential SOA security objective. It is a UML profile where security intents can be modeled as stereotypes on UML modelling elements during the business process modelling. Aim is to facilitate the business process expert in modelling the security requirements along with the business process modelling. This security annotated business process model will facilitate the security expert in specifying the concrete security implementation. As a proof of work we apply our approach to a typical business process of "on-line flight booking system".*

Keywords: Service oriented architecture, Model driven architecture, Business process modelling, Security goals, Domain specific language, Unified modelling language

1. **Introduction.** Today's Information Technology (IT) environment is network/Internet centric such as Service Oriented Architecture (SOA), Cloud and SaaS (Software as a Service) which offer the IT (Information Technology) agility demanded by the business [1,2]. In SOA environment software applications are deployed over the Internet as a service. To support a business venture, these services are integrated within and across organizations to form Internet-based Web Information System (WIS) and perform cross application transactions [3]. However, it is full of daily virus alerts, malicious crackers and the threats of cyber terrorism [1,2]. With the increase in number of attacks on the system, it is probable that an intrusion can be successful [4]. The security violation defiantly causes losses, therefore it is necessary to secure the whole system. Regarding SOA security, it is not sufficient to just protect a single point, and a comprehensive security policy is required [5]. SOA environment required achievements of security at both levels, i.e., the overall security objectives of the entire systems as well as security compatibility between interacting services. Security measures implemented in SOA systems are viewed from

two different levels; first at high level security objectives, which are basically abstract representation of the security goals and the second at detailed security policies [3].

Security must be unified with the software engineering process; however, in practice it is considered afterthought and implemented in an ad-hoc manner [4]. Furthermore, it is left to the developer and added when the functional requirements are met or at the time of integration of distributed applications, which is not a realistic approach [6]. SOA applications are cross-domain and coupled over various network technologies and protocols; just adding security code to software applications is not a realistic approach because all required security information is not available at the downstream phases [6,7]. This approach degrade implementing and maintaining security of the system [8].

During the past few years, several security protocols, access control models and security implementations have been emerged to enforce the security goals [6,9]; however, focus of the SOA security standards and protocols is towards the technological level, which do not provide high level of abstraction; furthermore, mastering them is also a daunting task [5,10]. This approach leads to security vulnerabilities, which justify increasing efforts in defining security in pre-development phases, where finding and removing a bug is cheaper [11].

Individuals are performing various roles contributed to the modeling, development, deployment and management of the security aspects of a business application in SOA systems [12]. Business process modelling is the most appropriate layer to describe security requirements and evaluate risks [5]. Empirical studies show that those, who model the business process, i.e., business domain experts are able to specify security requirements at high level of abstraction, i.e., while designing the system [4]. The actual stakeholder of the business process is the business domain expert [4,9]. It is evident that business domain experts must define the security requirements at a business process model [13]. However, it is important to note that business domain experts define the broad goals for security objectives and possible range of offerings. It cannot be expected from him to specify security requirements in term of security technology, e.g., message security, encryption and SSL (secure Socket Layer); thus allow flexible implementations that meet these security objectives [12]. Both experts, i.e., business domain experts and security experts, work side-by-side while designing a business process model and defining security requirements [9]. Business domain experts, while modeling the business process; concentrate towards modeling the business process in a way that functional correctness is modeled and notion of security is often neglected. It may happen due to many reasons, e.g., the business domain expert is not a security expert [4,14]. Moreover, no currently available process modelling notation has ability to capture security goals [13]. Furthermore, system models and security models are disjoint and expressed in different ways, i.e., a system model is represented in a graphical way in a modelling language like Unified Modelling Language (UML) while a security model is represented in a structured text [4].

Business process modelling is normally performed in general purpose modelling language such as Unified Modelling Language (UML) or Business Process Modelling Notation (BPMN), and these modelling languages do not support specification of security requirements [15]. A general purpose modeling language like UML has a broad scope and there may be a situation where it is not appropriate for modeling some specific domains, e.g., security. There also may be situation when the syntax and semantics of UML elements are not able to express the specific concepts of particular systems or there may be a situation when a UML element may be customized or restricted which is normally too general and too abundant [16]. Some security extensions are proposed in general purpose modelling languages to annotate them with security goals [17,18] and work is still in progress.

To summarise, a business domain expert is unable to specify security requirements in SOA applications due to the following reasons:

- There is not a clear identification of security requirements to be modeled for SOA applications.
- Absence of notations to express these security requirements in a general purpose modeling language like UML.

Model Driven Security (MDS) and automatically developed software having security configuration has been a topic of interest among the research community and different research groups across the globe are trying to solve the security problems for SOA applications by presenting MDS Frameworks [6,9,15,17-20].

Our aim is to facilitate business domain expert to add security goals while modelling a business process for SOA applications by providing a Domain Specific Language (DSL) for security modeling. The security annotative business process model will facilitate the security expert while defining concrete security implementation.

In our work:

- We have provided detailed analysis of essential security intents for SOA applications. In our previous work [21], we have discussed only three basic security intents. Now, we have enhanced our DSL by incorporating more security intents.
- We have presented a DSL to express the security requirements and used UML-profiling mechanism to incorporate security intents in UML. UML is an industry standard for business modeling [4].
- As a proof of concept, we have projected our work to a real world business process model.

A business domain expert is facilitated to use a modeling language which is equipped with the vocabulary for specifying security objectives at PIM level of abstraction. Hence he/she only needs to understand the security concepts in the UML-based security design language and does not have to expertise in target security technologies [8]. Specifying security requirements at abstract level helps the architectural team in choosing different, and potentially better, security mechanisms, e.g., biometric devices such as retina scanners and fingerprint readers to meet the real underlying security requirements [2]. Being able to express security requirements in a widely used design notation likes UML or BPMN for SOA systems helps to save time and effort during the implementation and verification of security in system [22].

2. **Related Work.** To model the security objectives related to different aspects of the systems; different extensions are proposed in modelling languages and security DSLs are proposed. Mostly, authors represent the abstract syntax of their DSL by a meta-model using MOF framework and concrete syntax by UML profile [4,11,19,23]. Related work exists along different models of software development and following is its descriptions:

System Models: Static structure of the system is represented by UML class diagram and UML state diagram [24]. D. Basin et al. [8] have presented SecureUML to model the security objectives for modeling static structure of the system. Basically, it is a separate language based on Role Based Access Control (RBAC) protocol. Afterwards, SecureUML can be integrated with any system modeling language like UML to model the security in the system design. They have presented a meta-model for abstract syntax and used UML profiling mechanism for concrete syntax and security constraints are added through OCL.

Interaction Diagram: UML sequence diagram is used to represent the flow of control between the object of the system [24]. J. Jürjens [25] defined UMLSec and developed a UML profile to incorporate security to represent the secure interaction.

Deployment Diagram: UML component diagram is used for the representation of deployment of a system [24]. UMLSec presented by J. Jürjens [25] also support the secure modelling of UML component diagram.

Work Flow Model: UML activity diagram and BPMN are used to represent the business process work flow. This is the most important aspect of a system and most of the security extensions are proposed related to this aspect.

A. Rodriguez et al. created a meta-model for their security extensions and defined security stereotypes and developed a DSL. They also assign different symbols to these security stereotypes. They used the same DSL for extending BPMN [4] as well as UML [11]. C. Wolter et al. [13], incorporated security stereotypes in BPMN. R. Brue et al. [19] also presented security stereotypes in UML activity diagram.

We are also working along the way and focusing security modeling along with work flow modeling for SOA environment. We are trying to enrich our DSL to cover all essential security intents for SOA environment.

3. Foundation Concepts.

3.1. Service oriented architecture (SOA).

SOA paradigm makes the software application development easy by coupling services over intranet and via the Internet [6]. SOA paradigm has changed the Internet from being repository of data to repository of services [26]. SOA is an architectural style in which software applications are comprised of loosely coupled and reusable services by integrating these services through their standard interface. Services are independent of language, platform and location and may be locally developed or requested from the provider. A business process can be realized as a runtime orchestration of set of services. Software applications are often comprised of numerous distributed components such as databases, web servers, computing nodes, storage nodes etc and these components are distributed across different independent administrative domains. Services are used but not owned by the user and they reside on provider side. The reusability, agility, cost effectiveness and many other attributes of SOA paradigm has attracted the organizations to adopt it for software development [27-29].

The basic building block of a SOA paradigm is a service. "*A service is an implementation of a well-defined piece of business functionality, with a published interface that is discoverable and can be used by service consumers when building different applications and business processes*" [30]. SOA paradigm can be implemented with different technologies like CORBA, Web Services, JINI, etc.; however, Web services technology is a widespread accepted instantiation of SOA [29,31].

3.2. Web services technology.

Web Services are defined as "*self-contained, modular units of application logic which provide business functionality to other applications via an Internet connection*" [31]. Software applications are developed by integrating different web services either newly built or legacy applications by avoiding difficulties due to heterogeneous platforms and programming languages by exploiting the XML (Extensible Markup Language) and the Internet technologies [31,32]. Web service enable the dynamic connections and automation of business processes within and across enterprises for EAI (Enterprise Application Integration) and B2B (Business-to-Business) integration using the web infrastructure with relative standards like HTTP (Hyper Text Transfer Protocol), XML SOAP (Simple Object Access Protocol) WSDL (Web Services Description Language) and UDDI (Universal Description Discovery and Integration).

3.3. **Business process modelling.** Business process modelling is gaining more and more attention in an organization because it is the foundation to describe the organizational workflow [5]. An effective business process model will facilitate the stakeholders of the business to understand the different aspects of the business system and provide a platform to discuss and agree on key fundamentals for achieving the business goals [4]. A business process is defined as "*a set of procedures or activities which collectively pursue a business objective or policy or goal*" [4]. It can also be defined as "*a set of activities and execution constraints between these activities*" [5]. Different techniques are used for business process representation; N. Damij [33] grouped them in two categories: diagrammatic and tabular. C. Wolter et al. [13] described different popular diagrammatic business process modelling notations like BPMN, UML, XPDL and Jpdl; among these UML and BPMN are considered as industry standards [4].

3.4. **Model driven approach (MDA) and model driven security (MDS).** Currently, software engineering is greatly influenced by a new paradigm presented by OMG (Object Management Group) known as MDA; which work at model and meta-model level [34]. In MDA approach, software systems are specified and developed through models; transformation functions are automatically performed between models at different levels of abstractions as well as between models to code [8]. Model based design methodology is being widely accepted in the development of electronics systems due to their flexibility and tool support. To organize landscape of model, meta-modelling techniques are emerged; theories and methods are provided for the development of coordinated representation suitable for heterogeneous environment such as SOA [35].

MDS specializes MDSD towards information security [36]. MDS is a technology where security requirement are defined as a model during designing phase and concrete security configuration files can be generated by model transformation [37].

4. **Extending a Modeling Language according to a Particular Domain and Definition Process of a DSL.** There are three main limitations a general purpose modelling languages have: lack of semantics, lack of visualization and lack of abstraction for modelling a specific domain while preparing a business process model [38].

As compared with the general purpose modelling languages, DSLs offer substantial gain in ease of use and expressiveness according to the specific domain according to which they are developed. DSLs results in considerable gain in productivity, reduction in maintenance cost and reducing the required domain specific expertise. DSLs are also called application oriented, special purpose, specialized or task specific language. Appropriate notions related to the specific domain are usually beyond the notation offered by general purpose modelling language. DSL development requires language development expertise as well as domain knowledge [39].

A domain can be defined as "*a field of application delimitated by a specific area of interest*" [36]. Application structure, requirements and behavior according to a specific domain are formalized in the form of DSL which is one of the components of MDSD. DSL is defined as "*Concise, precise and processable description of a viewpoint, concern or aspect of a system, given in a notation that suit the people who specify that particular viewpoint, concern or aspect*" [36]. A DSL consists of constructs that capture information regarding the domain it describes [40]. DSL may also called Domain Specific Modelling Language (DSML) [41].

DSL development requires language development expertise as well as domain knowledge [39]. Following are the three alternatives for defining a DSL for modelling [8,42].

1. The easiest way of defining a DSL is the usage of the extensions points provided by the language itself [42]. DSL can be defined directly in UML in a lightweight way by using stereotypes and tagged values known as "labels" resulting *UML profile*. A UML profile describes how UML model elements are extended to support usage in a particular domain [40]. A profile is a lightweight extension mechanism and thus cannot be used to add new model elements or delete existing model elements [40], to introduce new language primitives, *stereotypes* are used by extending the semantics of existing model elements present in the UML meta-model [43]. Stereotypes are represented by double angle brackets, e.g., $\ll$*stereotype*$\gg$. To formalise the properties of these new language primitive, *tagged values* are used which are written within curly brackets, e.g., {Tag, Value} [43], which associate data with model elements. Model elements are assigned to these new language primitives and labelled them with corresponding stereotype. If some additional restrictions are required on the syntax of these new language primitives; Object Constraints Language (OCL) constraints is used. OCL is a specification language provided by UML, based on first order logic. Normally, OCL expressions are used for various purposes such as invariant for classes, pre and post conditions for methods and guards for state diagram. Set of such definitions, i.e., stereotype, tagged values and OCL constitutes the UML profile [8]. Most of the current UML modelling tools can readily use because they support the definition of custom stereotypes and tagged value. Because of having tool support this approach is widely used [8,23,25]. Normally, DSLs are defined by UML-Profiles when the "domain" may be combined with other domains, in an unpredictable way and the model defined under the domain may be interchanged with other domains [23].

Limitation is that; UML 2.0 profiling mechanism do not support the semantics associated with extensions that's why it cannot be used to develop domain specific UML variant that support the formal model manipulation required in an MDE environment [40]. It is very clumsy to add domain-specific restrictions in large languages like UML; furthermore for formal analysis, large languages usually lack detailed formal semantics [23]. Visualization of the complicated security intents might be confusing; furthermore, many modelling languages do not provide extension points [42].

Remaining two extension techniques are *meta-model based* techniques and known as heavy weight extension mechanisms. The meta-model based technique of defining DSL is mostly used when the "domain" is well defined and has accepted set of concepts; there is no need to combine the domain with other domains and the model defined under the domain is not transferred into other domains [23].

2. DSL can be defined by using MOF by extending the meta-model of existing modelling languages like UML. Concept of stereotype is used to formally extend the meta-model of an existing modelling language. At modelling level, stereotypes are manipulated as annotation on model elements. In this way of DSL definition, an existing meta-model is reused and specialized.

The limitation is that the extended and customized meta-model is based on the entire meta-model of existing modelling languages and may be complex [8,23,35]. During this approach, manually changes are applied to the metamodel of an existing modelling language is tedious and error prone due to many reasons: (1) difficulty in ensuring that the changes are made consistently across the metamodel; (2) difficulty in determine the impact of change on other model elements; and (3) difficulty to ensure that the resulting modified meta model is complete and sound [40]. Furthermore, to support the DSL; CASE (Computer Aided Software Engineering) tool may also require extension to accommodate

these new language primitives in particular storage component (repository) and visualization component [8,23,35]. Furthermore, extensions are defined and integrated according to a particular domain into a specific modelling language based on its meta-model [42].

3. A new DSL for modelling the domain of interest or particular problem is created by a fully dedicated meta-model using MOF having no dependency on existing modelling languages. The resulting DSL have much more concise vocabulary than the vocabulary of general purpose modelling languages, e.g., UML or BPMN. For querying and manipulating meta-data of these DSL, interface would be more simple than the UML Interfaces. Abstract syntax is represented by the meta-model and notions (concrete syntax) of the DSL are specified with the UML profile [8]. This way of extension is optimally suited for the problem at hand [35]. Example of such a language is CWM (Common Warehouse Metamodel) [16].

Limitation of this techniques is, sometime it does not provide the well defined mapping between the UML model with which developer work, to the instances of meta-model of DSL that define the meaning of this model [23].

Unless there is a real need to deviate from the UML metamodel, the benefits of using UML Profiles undoubtedly outweigh their limitations [16].

5. **Essential Security Goals to be Modeled for SOA Applications.** Generally, security is consider as a state of freedom from risk or danger, however, in computer sciences, it is a fields which deals with the risks, threats and mechanism to the use of computing system [36]. Computer security can be defined as "*A computer is secure if you can depend on it and its software to behave as you expect*" [44]. However, security is not just like a state only, it also describes the other things, e.g., the measures to preserve this state, in [45], author define security as "*Computer security deals with the techniques employed to maintain security with in a computer system*". These two definitions for the computer security can be correct for the isolated host, however, they short fall to the modern computing system where we have loosely coupled components distributed over a network, heterogeneous and interconnected, e.g., SOA environment. Computer systems are no more conceived as a centralized architecture. A system which is connected to other systems is exposed to many additional security threats such as the case of SOA environment. That is why a comprehensive security definition is required which also covers the environment to which the system is belongs to. Going beyond the traditional monolithical computing system, a very comprehensive security definition is given by [36] "*the sum of all techniques, methods, procedures and activities employed to maintain an ideal state specified through a set of rules of what is authorized and what is not in a heterogeneous, decentralized and inter-connected computing system*".

Security is an abstract concept which can be defined precisely by specifying set of security goals or objective [13]. Security objectives describe the most basic security need of an asset [36] and they can be defined as "*a statement of intent to counter identified threats and/or satisfy identified organizational security policies and assumptions*" [46]. These security goals can be further subdivided, specialized or combined [13]. Many names can be found in literature for security objectives like security properties, security aspects, security concern, security intents or security states [47]. Massive literature can be found on software security, however, we would like to able to specify basic security intents which can be easily understandable, modeled in the business process and can be used for the identification of specific security implementation [48]. During our work we mainly focus security measures to encounter the threats related to: use of identity information and associated rights (authentication, authorization), information in different forms, i.e., stored, transferred or processed (confidentiality and integrity of data) and service function

(availability and integrity of a system) [5]. Although there has been a lot of research on security issues and concerning technologies, however, we want to address business-level security intents which are easy to understandable for a business process expert [6], who is not well versed with the technical security details, however, he/she is able to model it at very abstract level [48].

5.1. **Security objectives in related work.** There are numerous SOA security concerns that may differ for each stakeholder like vendors, security experts, consultants and business process experts. Similarly, security concerns can be about some specific business case, technology, governance, deployment, etc. Unclear security intents results in unclear security implication which is cited as one of the most important issue that limit the SOA benefits and hence slow down its adoption [1].

5.1.1. *Security intents general.* These are the security intents irrespective of the deployment environment it may be considered in any of the environment. Here authors did not mention anything about the deployment environment.

N. Nangaratnam et al. in [12] specify the five security primitives for a business process model naming audit, authenticate, authorize, confidentiality and integrity.

Firesmith in [2] have very comprehensive discussion about the general security of a software application and identified eleven security objectives and corresponding thirteen security requirements. The eleven security objectives are: identification, authentication, authorization, immunity, integrity, intrusion detection, non-repudiation, privacy, security auditing, survivability and physical protection.

In [4,11], A. Rodríguez et al. extended the UML and BPMN by defining DSLs and focusing on five security goals: access control, integrity, privacy, attack-harm detection and non-repudiation.

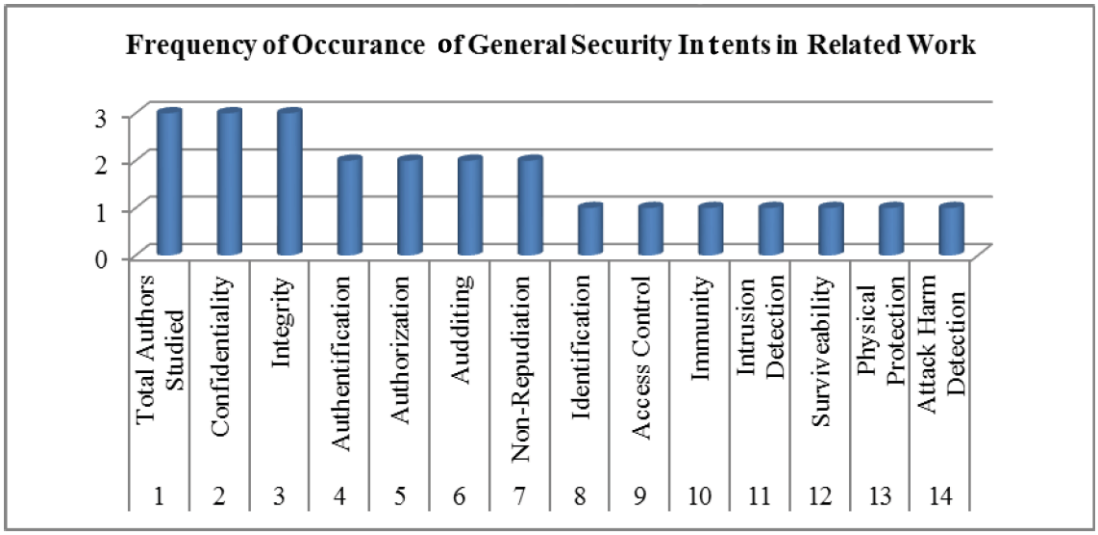
The whole discussion is summarized in Table 1 representing the general security intent in related work and Table 2 represents them in graphical form.

TABLE 1. General security intents focused by different authors in their work

S/No	Security Objectives	1	2	3	4	5	6	7	8	9	10	11	12	13
	Research Groups	Access Control	Identification	Authentication	Authorization	Confidentiality	Integrity	Non-Repudiation	Auditing	Immunity	Intrusion Detection	Survivability	Physical Protection	Attack Harm Detection
1	Nangaratnam et al.			√	√	√	√		√					
2	Firesmith		√	√	√	√	√	√	√	√	√	√	√	
3	A. Rodríguez et al.	√				√	√	√						√

5.1.2. *Security intents of SOA applications.* These are the security intents where authors specifically mentioned about the SOA environment. Different research groups are focusing different security goals for their DSL for SOA applications [4,6,11,13,40].

TABLE 2. General security intents in graphical form



M. Hafner et al. [36] defined the three security goals naming confidentiality, integrity and availability. They defined access control under the umbrella of confidentiality; and availability is used in the meaning of no-repudiation.

C. Wolter et al. [13] presented a security policy model by focusing six security goals: authentication, authorization, confidentiality, integrity, availability and auditing. M. Menzel et al. used the same security policy model specified by C. Wolter et al. in their work [5] and defined security extensions to the BPMN.

M. Menzel et al. [49] specified the four security goals necessary for the SOA architecture: authorization, authentication, integrity and confidentiality.

Y. Nakamura et al. [7] defined three security intents for their work: authentication, integrity and confidentiality and defined a UML profile. In [6], Y. Nakamura et al. addressed four business level security intents as they are easy to be understood by business user naming: authentication, integrity, non-repudiation and confidentiality.

S. Johnston [48] described seven security intents which are essential for SOA environment: identification, authentication, authorization, privacy, audit, data integrity and non-repudiation.

U. Lang and R. Schreiner [1] described the five security objectives in their work naming Confidentiality, Integrity, Availability, Auditing and Manageability.

T. Erl [50] presented an overview of the security intents for the WS Security and presented a framework containing five security requirements naming Identification, Authentication, Authorization, Integrity and Confidentiality.

T. Phan et al. [51] introduced a method for design and implementation of SOA Business Security Engineering. The security objectives they focused in their work are confidentiality, integrity, non-repudiation, audit, authentication, authorization and message freshness.

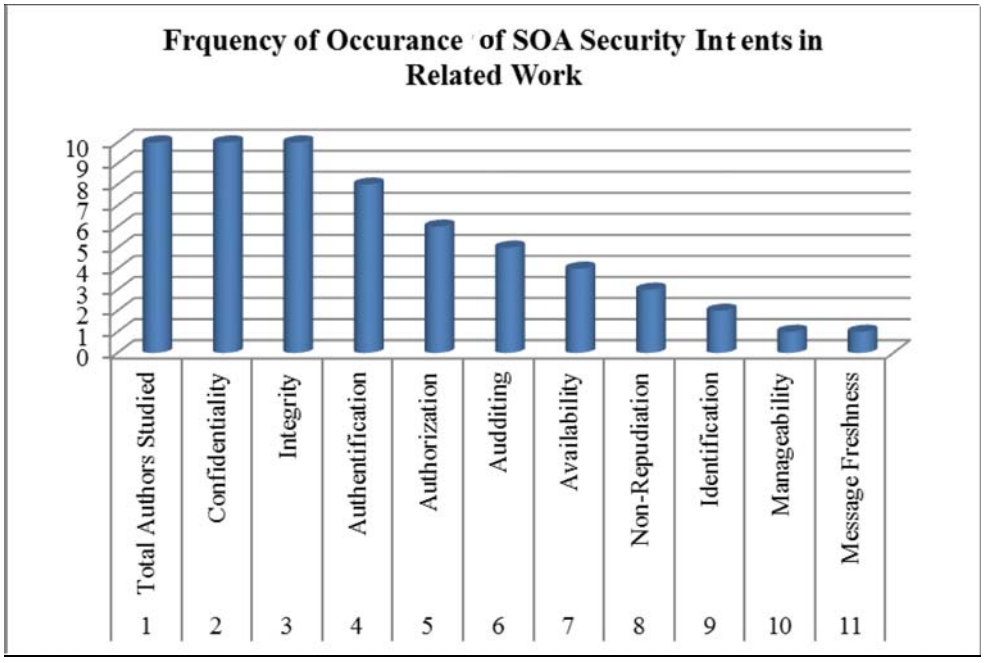
The whole discussion is summarized in Table 3 representing the security intent for SOA environment in related work and Table 4 represents them in graphical form.

5.2. **Discussion.** Among the eleven security objectives discussed by Firesmith in [2]; four are out of scope of our work; *physical protection*, *intrusion detection*, *survivability and immunity*; because we are focusing only those security objectives which are essential to be modelled along the business process modelling for SOA application.

TABLE 3. Security intents focused by different authors in their work for SOA environment

S/No	Security Objectives	1	2	3	4	5	6	7	8	9	10
	Research Groups	Identification	Authentication	Authorization	Confidentiality	Integrity	Availability	Non-Repudiation	Auditing	Manageability	Message Freshness
1	M. Hafner et al.				√	√	√				
2	C. Wolter et al.		√	√	√	√	√		√		
3	M. Menzel et al.		√	√	√	√	√		√		
4	M. Menzel et al.		√	√	√	√					
5	Y. Nakamura et al.		√		√	√					
6	Y. Nakamura et al.		√		√	√		√			
7	S. Johnston	√	√	√	√	√		√	√		
8	U. Lang and R. Schreiner				√	√	√		√	√	
9	T. Erl	√	√	√	√	√					
10	T. Phan et al.		√	√	√	√		√	√		√

TABLE 4. SOA security intents in graphical form



"*Identification*" is specified as a separate security intent in [2,48]. Identification and authentication are assumed when we are trying to model: "who are you?". If we take the example of ATM with drawl; ATM card is a bank-issued identification, whereas the PIN-code allows the ATM to authenticate the person as an account holder. It is far more important to model the notion of authentication than identification [48]. We believe

identification is a part of authentication process and no need to model it explicitly in a business process diagram.

"*Access control*" is an abstract term used in [4,11] by A. Rodríguez et al. for the description of identification, authentication and authorization security intents. We believe if we model only "access control" then it would hide many details and it is better to model authentication and authorization separately in a business process model.

"*Attach-harm-detection*" specified by A. Rodríguez et al. in [4,11]; is a security mechanism which will allow to detect, register and notify an attack attempt or a successful attack. It is a same kind of mechanism as described by Firesmith in [2] under the name of "immunity" security objective. Every organization is supposed to have some security protections mechanisms such as anti-virus or firewall; therefore no need to model it explicitly in a business process diagram.

U. Lang and R. Schreiner in [1] taking accountability in the sense of auditing. They also mentioned a security intent "manageability", i.e., IT Security should be manageable. It is a concept of overall security related to the SOA environment; therefore, no need to model it explicitly in a business process diagram.

5.3. **Security objectives of UML-SOA-Sec.** Among the fifteen security objectives as discussed, we believe following seven are the essential security objectives which should be modeled in a business process model of SOA applications; which are focused by different authors either as it is or with some different name or by merging them.

1. *Data Confidentiality:* It specifies the system's state where only authorized entities can access the information [36]. In [5,13], confidentiality is defined as "*It provides protection against the unauthorised notice of stored, processed or transferred information*". To represent the data confidentiality security requirement some authors use the term "Privacy" instead of confidentiality [2,4,48]. It is also think in the term of "secrecy" [49].

The typical objective of the data confidentiality is to ensure that [2]:

- Unauthorized individuals and programs do not gain access to sensitive data and communications.
- Access to data and communications is provided on a "need to know" basis.

2. *Data Integrity:* It identifies an authorized subject to alter information in authorized ways [36]. It ensures the integrity of data (properness, intactness, correctness and completeness of information) as well as integrity of origin [5,36,49]. It ensure that the transferred, processed or stored data can only be modified with proper rights [13]. Basically, it ensures that the transferred data between parties must be guaranteed to reach the recipient in the same form and with the same content [6]. Typical objective of the data integrity is to make the data and communication trust worthy [2].

3. *Authentication:* It ensures the credibility of information by confirming them as authentic [5,13]. It is a mechanism to verify the identity of an entity [36]. It establish the trust relationship between a subject and a party that relies on claim stated by the subject [49]. The typical objective of the authentication is to ensure that "*externals are actually who or what they claim to be and thereby to avoid compromising security to an impostor*" [2].

4. *Authorization:* Authorization is based on some specific security model, how to grant various privileges to various entities on different resources [36]. Basically, it is a process of granting rights to participants to perform interaction or task [5,13]. It determines the rights which will be granted to the subject based on the trust relationship and properties of the subject's identity [49].

The typical objective of authorization is to ensure that [2]:

- Person (Administrator of your system) are able to are able to authorize specific authenticated users and client applications to access specific application or component capabilities or information.
- Authenticated externals (users or client application) can access specific application or component or information if and only if they have been explicitly authorized to do so by a properly appointed person(s).

5. *Availability:* It ensure that the data, resources and services which are needed for the proper functioning of a system, are available at each point in time regarding the requested quality of service [5,13].

6. *Non-Repudiation:* A user may use a resource or call a service and this usage or service call must not deniable [36]. Basically, it ensures that the information must include the digital signatures of the parties related to the document [6].

Typical objective of the non-repudiation security requirements are [2]:

- Proper temper-proof record keeping is going regarding the interactions of the parties to prevent them from denying that it have taken place.
- To minimize any potential future legal and liability problems that might causes due to someone denying one of their interactions.

7. *Traceability and Auditing:* It is a process of verification of all actions performed in an information processing system [13]. Basically, auditing is to verify all operations in an information system [13]. It underlies with each security requirement; and will automatically be understood when a security requirement is specified in a model [4].

The typical objective of traceability and auditing security requirements are to ensure that the software application will collect, analyse and report information about the status (e.g., enabled vs. disabled) and use (e.g., change in properties) of its security mechanism [2].

6. Proposed DSL: UML-SOA-Sec.

To gain the benefits of DSL and general purpose modelling language, DSLs are defined in terms of general purpose modelling language like UML or BPMN [23]. In our research work, our domain is "*modelling the security in SOA system*". General purpose modelling language like UML can easily be customized by the extension mechanism provided by the language itself and DSL can be defined according to the domain of interest by extending the general purpose modelling language. In case of UML, the extension mechanism is known as *UML Profile*. Tools are available for the general purpose modelling languages which support the definition and usage of DSL. In our case, we have focus the domain of "*SOA Security*" and we have extended the general purpose modelling language UML by providing a profile and we named it as "*UML-SOA-Sec*". We have used *MagicDraw* tool for UML modelling which support the definition and usage of DSL. The whole phenomenon can be explained by Figure 1.

Current practice of defining a DSL by different researchers [4,8,11,13,25] is; abstract syntax is represented by a meta-model and concrete syntax is represented by a UML Profile. We are also working along this approach and have defined the abstract syntax of our DSL by a meta-model; and concrete syntax by a Profile.

6.1. Abstract syntax.

Abstract syntax of our DSL is presented by a metamodel. The UML profile that describes our metamodel is described as UML package with the stereotype ≪profile≫ as shown in Figure 2. We are using package for the creating of our DSL as discussed in [52]. Our DSL is based on the security intents disused in previous section. The most difficult task is the identification of elements of the meta-model of a modelling language which must be extended for example in case of UML, identification of UML metaclasses for which the stereotypes will be defined [35]. In our case, we have extended

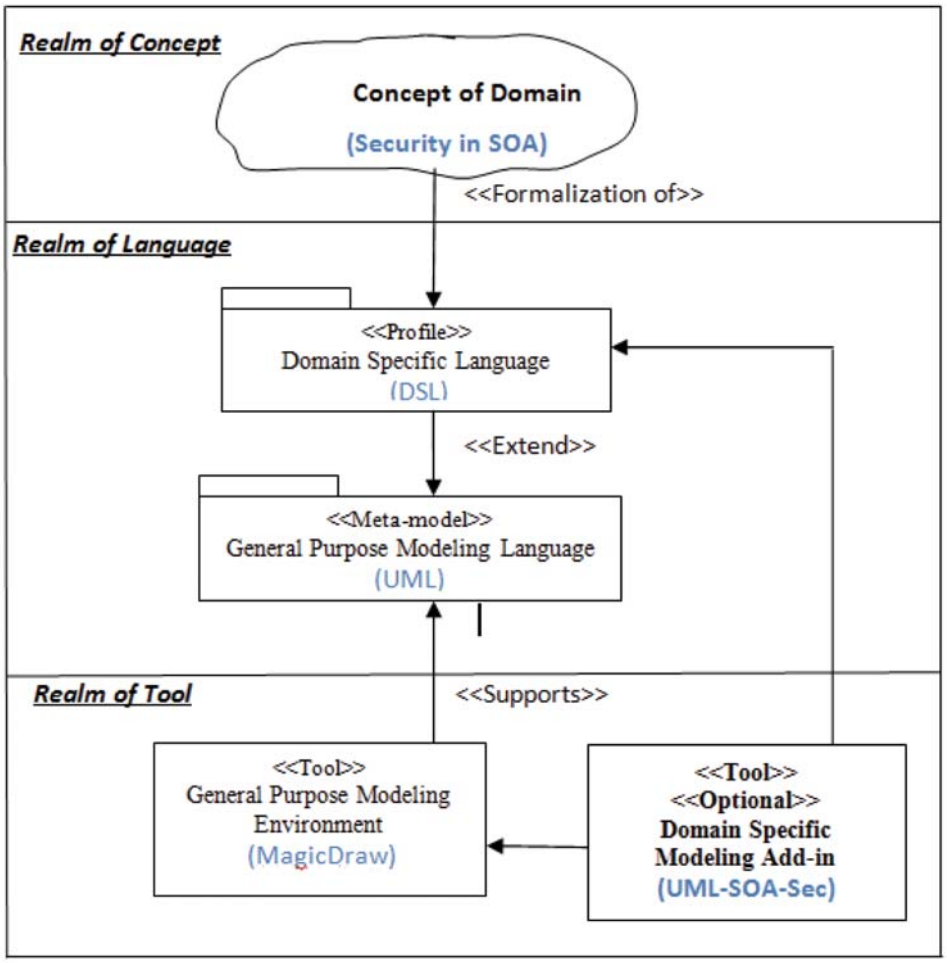


FIGURE 1. Definition process of a domain specific language [35]

UML meta-classes *ObjectNode and ActivityNode*, i.e., these are the metaclasses to which stereotypes will be assigned.

6.2. **Concrete syntax.** Each extension of the elements of UML meta-model is formally captured under the concept of *stereotypes*. Properties and/or modelling constraints of the target domain are associated with the stereotypes which results the *UML profile*. After the definition of domain specific UML-profile, general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotation [35]. For concrete syntax, we have presented stereotypes as shown in Table 5.

7. **Case Study.** To demonstrate our work, a case study of "Online Flight Booking System" is presented. It describes the web services based interaction between the participants and enables them to work through the Internet. The whole process has to be realized in a peer-to-peer fashion and would integrate security requirements.

7.1. **Business scenario.** Nowadays travel agencies provide online services to travelers for booking the flights. Traveler submits the trip information to the travel agency, containing the personal information of travelers; start date, end date, origin, destination and price rang, etc. After having this information travel agency search for the suitable airline and routes accordingly and prepare itinerary and send it to traveler. If traveler accepts the itinerary then he/she will make payment into the bank specified by the travel agency.

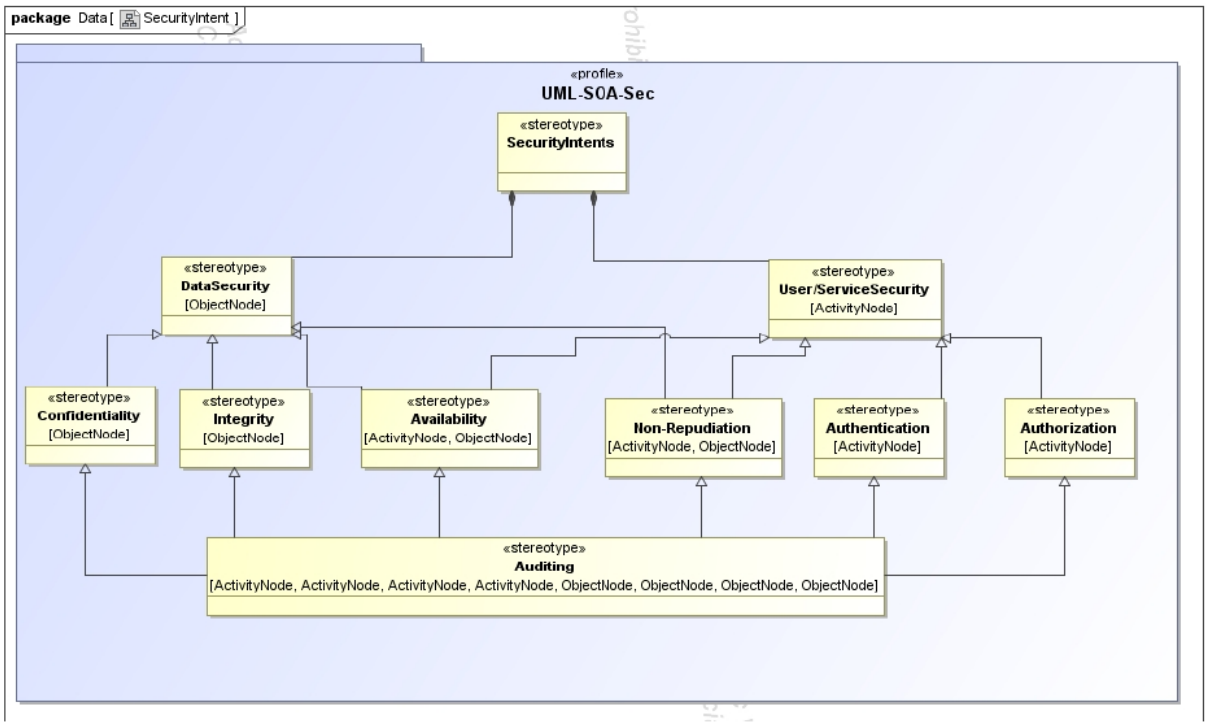
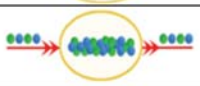


FIGURE 2. Abstract syntax of proposed DSL

TABLE 5. Concrete syntax (notions) of proposed domain specific language

S/No	Stereotype Name	Meta Classes	Symbols	Description
1	Data Confidentiality	ObjectNode		It is applied on the data communicated to ensure that this data is private.
2	Data Integrity	ObjectNode		It is applied on the object node to represents the integrity of data.
3	Non-repudiation	ObjectNode		It is used to represent that the data transferred contain the information about the originating party.
4	Availability	ObjectNode		It is used to represents the availability of a data, resource or service.
5	Authentication	ActivityNode		It is applied on a party who want to initiate a collaboration
6	Authorization	ActivityNode		This stereotype is used to denote the communication between the two parties where requester has to gone through the authentication check.
7	Traceability and Auditing	ActivityNode		When this stereotype is applied, it represents the auditing of some action.

The bank; upon receiving payment send receipt of payment to both, i.e., traveler as well as travel agency. After receiving conformation of payment, travel agency will order ticket from airline, which will send the ticket to the traveler.

7.2. **Stakeholders.** In the case-study, services from the four stakeholders are involved, i.e., traveler, travel agency, airline and bank.

7.3. **Security requirements of the system.** In online flight booking system, a traveler needs to perform different tasks, i.e., fill in the trip information form, viewing the itinerary, make payment into the bank and view the ticket. Necessary permissions are assigned to him/her on different objects to perform these tasks, i.e., travels require update information on trip information payment form, read permission on itinerary information and ticket. To perform these operations traveler's personal information are involved at different places, e.g., passport number while filing the trip information and credit card information while making payment to bank. Therefore, confidentiality is required, i.e., proper access control mechanism with authentication and authorization is required to access this information. Furthermore, traveler has to submit the trip order to the travel agency, traveler must sign it with his/her signature so he/she may not be able to deny that he/she has not submitted the trip order. Availability (Non-repudiation) is required in this use-case between the traveler and travel agency. Travel order form is submitted online, therefore secure information flow, i.e., Integrity is required to successfully perform this use-case. These three security requirements, i.e., Confidentiality, Availability and Integrity are identified and modeled for other stakeholders of the case-study like travel agency, airline and bank. Figure 3 shows the security enhanced business process model of the flight booking system use case.

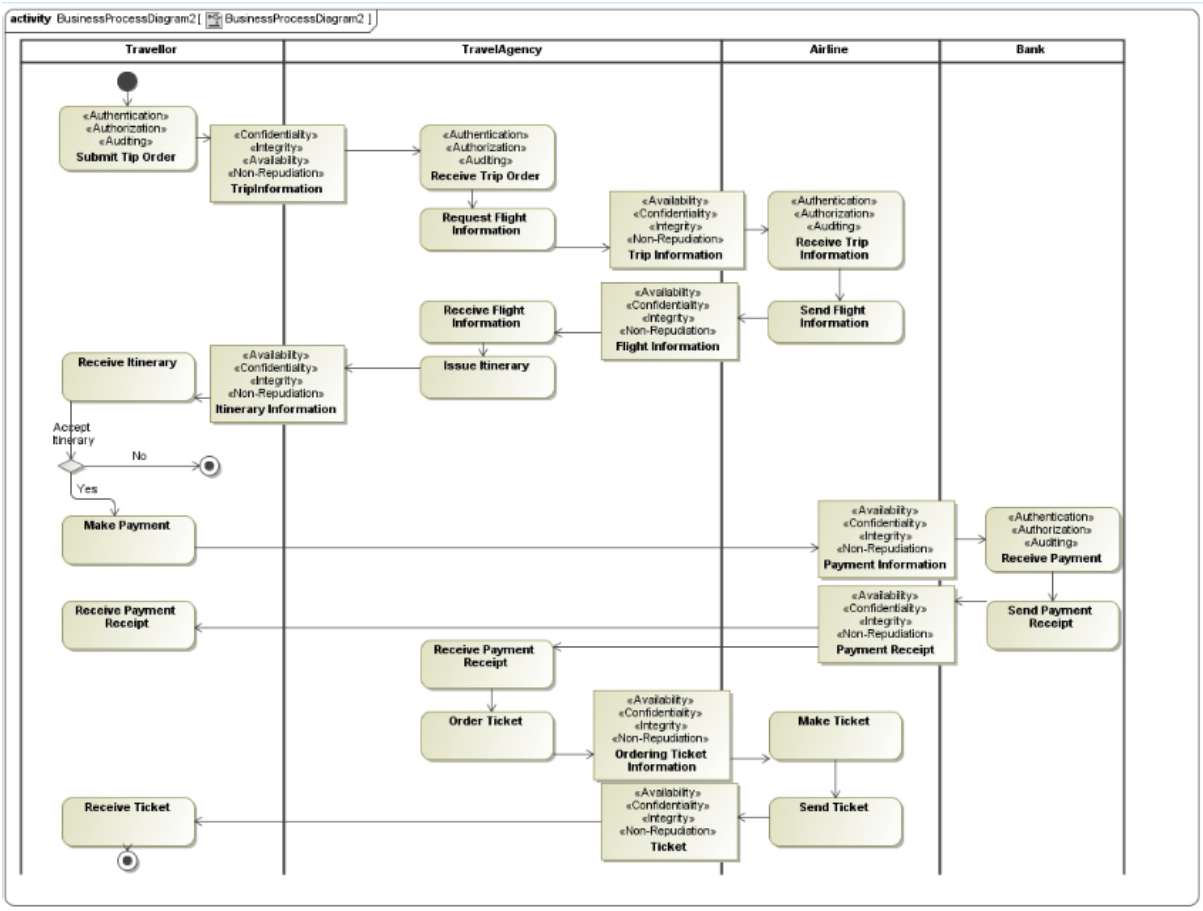


FIGURE 3. Security enhanced business process model of case study

Meaning of a particular security symbol at a specific place is discussed below.

Confidentiality: Whenever some information are sent or received they are consider as confidential, i.e., we show confidentiality requirement on data objects. In UML activity diagram, it would be modeled along the Objectnode.

Integrity: This security requirement is modeled whenever some transmission of information is takes place. It represents integrity of the information transmitted over the Internet. In the case, study whenever stakeholders interact with each other through sending messages; integrity symbols would be modeled over the message flow to ensure the integrity of information flow. In UML activity diagram, it would be modeled along the Objectnode.

Availability: It ensures the availability of data as well as service, i.e., whenever some data or service is requested then it would be available. In UML activity diagram, it would be modeled along Objectnode and activitynode.

Non Repudiation: Whenever some information would be sent or received between the stakeholders; then sending person would include additional information like digital signature, time and date along with the message, so the interactions cannot be denied. Same is the case with service, if some service is requested then this service request cannot be deniable. This security requirement is applicable to both data as well as service and in UML activity diagram it would be modeled along Objectnode and activitynode.

Authentication: Whenever some user wants to access some service then he/she has to authenticate himself/herself. In UML activity diagram, it would be modeled along the activitynode.

Authorization: After the authentication process of user, if he wants to perform some operation then his/her authorization would be checked. In UML activity diagram, it would be modeled along the activitynode.

Traceability/Auditing: Basically, it is used to monitor the overall security of the system and report generation. In UML activity diagram, it would be modeled along the activitynode.

8. Conclusion.

Incorporating security requirements during early stages of software development improve the "Security" of Web Services based SOA applications. A security DSL; containing essential security objectives for SOA applications, is presented to model the security along with the business process model at PIM level of abstraction. We have used UML-profiling mechanism to extend the UML and security objectives will be modeled as stereotypes. For our work, we have used Magic Draw modeling tool. A business domain expert is facilitated to use modeling language which is equipped with the vocabulary for specifying security objectives. Hence, he/she only need to understand the security concepts in the UML-based security design language and do not have to expertise in target security technologies. Specifying security requirements at abstract level help the architectural team in choosing different, and potentially better, security mechanisms, e.g., biometric devices such as retina scanners and fingerprint readers to meet the real underlying security requirements. Being able to express security requirements in a widely used design notation likes UML; for SOA systems, helps to save time and effort during the implantation and verification of security in system.

The next step we are focusing is how to define the security requirements using our symbolic language for Service Composition Modelling. Now the focus is different, to compose one advanced service (service composition) out of atomic basic services and we will define/model the security requirements for those atomic services and then for the composed advanced service.

REFERENCES

[1] R. S. Dr. Ulrich Lang, Top SOA security concerns & OpenPMF model-driven security, *ObjectSecurity White Paper, Topics Cloud Computing and Security Management*, 2009.

[2] D. G. Firesmith, Engineering security requirements, *Journal of Object Technology*, vol.2, no.1, pp.53-58, 2003.
[3] D. Xie, S. Ying, T. Zhang, X.-Y. Jia, Z.-Q. Liang and J.-F. Yao, An approach for describing SOA, *International Conference on Wireless Communications, Networking and Mobile Computing*, pp.1-4, 2006.
[4] A. Rodríguez, E. Fernández-Medina and M. Piattini, A BPMN extension for the modeling of security requirements in business processes, *IEICE – Transactions on Information and Systems*, vol.E90-D, no.4, pp.745-752, 2007.
[5] M. Menzel, I. Thomas and C. Meinel, Security requirements specification in service-oriented business process management, *International Conference on Availability, Reliability and Security*, pp.41-48, 2009.
[6] Y. Nakamura, M. Tatsubori, T. Imamura and K. Ono, Model-driven security based on a Web services security architecture, *IEEE International Conference on Services Computing*, vol.1, pp.7-15, 2005.
[7] F. Satoh, Y. Nakamura, N. K. Mukhi, M. Tatsubori and K. Ono, Methodology and tools for end-to-end SOA security configurations, *IEEE Congress on Services – Part I*, pp.307-314, 2008.
[8] D. Basin, J. Doser and T. Lodderstedt, Model driven security: From UML models to access control infrastructures, *ACM Transactions on Software Engineering and Methodology*, vol.15, pp.39-91, 2006.
[9] C. Woltera, M. Menzelb, A. Schaada, P. Miseldinea and C. Meinel, Model-driven business process security requirement specification, *Journal of Systems Architecture*, vol.55, no.4, pp.211-223, 2009.
[10] M. Alam, Model driven security engineering for the realization of dynamic security requirements in collaborative systems, *Models in Software Engineering*, pp.278-287, 2007.
[11] A. Rodríguez et al., Towards a UML 2.0 extension for the modeling of security requirements in business processes, *Trust and Privacy in Digital Business*, pp.51-61, 2006.
[12] N. Nagaratnam et al., Business-driven application security: From modeling to managing secure applications, *IBM Systems Journal*, vol.44, pp.847-867, 2005.
[13] C. Wolter et al., Modelling security goals in business processes, *Proc. of GI Modellierung*, Berlin, Germany, vol.127, pp.197-212, 2008.
[14] A. Rodriguez, E. Fernandez-Medina and M. Piattini, Security requirement with a UML 2.0 profile, *The 1st International Conference on Availability, Reliability and Security*, 2006.
[15] M. Menzel and C. Meinel, A security meta-model for service-oriented architectures, *IEEE International Conference on Services Computing*, pp.251-259, 2009.
[16] L. Fuentes-Fernández and A. Vallecillo-Moreno, An introduction to UML profiles, *UPGRADE, the European Journal for the Informatics Professional*, vol.5, no.2, pp.5-13, 2004.
[17] J. Jurjens, UMLsec: Extending UML for secure systems development-tutorial, *Proc. of the 5th International Conference on the Unified Modeling Language*, 2002.
[18] T. Lodderstedt, D. Basin and J. Doser, SecureUML: A UML-based modeling language for model-driven security, *Proc. of the 5th International Conference on the Unified Modeling Language*, 2002.
[19] M. Hafner, R. Breu, B. Agreiter and A. Nowak, Sectet: An extensible framework for the realization of secure inter-organizational workflows, *Emeral Internet Research*, vol.16, no.5, pp.491-506, 2006.
[20] M. Memon, M. Hafner and R. Breu, SECTISSIMO: A platform-independent framework for security services, *Modeling Security Workshop*, 2008.
[21] M. Q. Saleem, J. Jaafar and M. F. Hassan, Security modelling of SOA systems using security intents DSL, *The 2nd International Conference in Software Engineering and Computer Systems (CCIS) Series of Springer, LNCS*, 2011.
[22] J. Jürjens, Developing secure system with UMLsec from business process to implementation, *Computing Laboratory University of Oxford GB*, 2001.
[23] A. D. Brucker and J. Doser, Metamodel-based UML notations for domain-specific languages, *The 4th International Workshop on Language Engineering*, 2007.
[24] I. C. Mikael Åkerholm and G. Mustapić, *Introduction for Using UML*, 2004.
[25] J. Jürjens, UMLsec: Extending UML for secure systems development, *The Unified Modeling Language*, pp.1-9, 2002.
[26] S. Hanna and M. Munro, Fault-based web services testing, *The 5th International Conference on Information Technology: New Generations*, pp.471-476, 2008.
[27] G. A. Lewis, E. Morris, S. Simanta and L. Wrage, Common misconceptions about service-oriented architecture, *The 6th International IEEE Conference on Commercial-off-the-Shelf-Based Software Systems*, pp.123-130, 2007.
[28] A. Dan and P. Narasimhan, Dependable service-oriented computing, *IEEE Internet Computing*, pp.11-15, 2009.

[29] R. K. P. Bianco and P. Merson, Evaluation of service-oriented architecture, *Technical Report, CMU/SEI-2007-TR-015*, Software Engineering Institute/Carnegie Mellon, 2007.
[30] L. O'Brien, L. Bass and P. Merson, Quality attributes and service-oriented architectures, *Technical Note, CMU/SEI-2005-TN-014*, Software Engineering Institute/Carnegie Mellon, 2005.
[31] A. Bucchiarone and S. Gnesi, A survey on services composition languages and models, *International Workshop on Web Services Modeling and Testing*, 2006.
[32] W. M. P. van der Aalst et al., Web service composition languages: Old wine in new bottles? *Proc. of the 29th Euromicro Conference*, pp.298-305, 2003.
[33] N. Damij, Business process modelling using diagrammatic and tabular techniques, *Business Process Management*, vol.13, pp.70-90, 2007.
[34] A. Rodríguez et al., Towards CIM to PIM transformation: From secure business processes defined in BPMN to use-cases, *Business Process Management*, pp.408-415, 2007.
[35] R. Passerone et al., Metamodels in Europe: Languages, tools, and applications, *Design & Test of Computers, IEEE*, vol.26, pp.38-53, 2009.
[36] M. Hafner and R. Breu, *Security Engineering for Service-Oriented Architectures*, Springer-Verlag, Berlin, 2009.
[37] F. Satoh et al., Methodology and tools for end-to-end SOA security configurations, *IEEE Congress on Services – Part I*, pp.307-314, 2008.
[38] S. Brahe and K. Østerbye, Business process modeling: Defining domain specific modeling languages by use of UML profiles, *Model Driven Architecture – Foundations and Applications*, pp.241-255, 2006.
[39] M. Mernik et al., When and how to develop domain-specific languages, *ACM Comput. Surv.*, vol.37, pp.316-344, 2005.
[40] R. France and B. Rumpe, Model-driven development of complex software: A research roadmap, *Future of Software Engineering*, pp.37-54, 2007.
[41] T. Lukman and M. Mernik, Model-driven engineering and its introduction with metamodeling tools, *The 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, Izola, Slovenia, 2008.
[42] M. Menzel and C. Meinel, SecureSOA modelling security requirements for service-oriented architectures, *IEEE International Conference on Services Computing*, pp.146-153, 2010.
[43] M. Q. Saleem et al., Model driven security frameworks for addressing security problems of service oriented architecture, *International Symposium in Information Technology*, Kuala Lumpur, Malaysia, pp.15-17, 2010.
[44] T. Garfinkel et al., Flexible os support and applications for trusted computing, *Proc. of the 9th Conference on Hot Topics in Operating Systems*, pp.25-25, 2003.
[45] D. Gollmann, *Computer Security*, John Wiley & Sons, 1998.
[46] *Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model*, CCIMB-99-031, Version 2.1, 1999.
[47] M. Schumacher et al., *Security Patterns: Integrating Security and Systems Engineering (Wiley Software Patterns Series)*, John Wiley & Sons, 2006.
[48] S. Johnston, Modeling security concerns in service-oriented architectures, *IBM DeveloperWorks*, 2004.
[49] M. Menzel et al., SOA security-secure cross-organizational service composition, *Proc. of Stuttgarter Softwaretechnik Forum*, Stuttgart, Germany, pp.41-53, 2007.
[50] T. Erl, *An Overview of the WS Security Framework*, http://www.soaspecs.com/ws-security.php, 2011.
[51] T. Phan, J. Han, I. Müller, M. Kapuruge and S. Versteeg, SOABSE: An approach to realizing business-oriented security requirements with web service security policies, *IEEE International Conference on Service-Oriented Computing and Applications*, pp.1-10, 2009.
[52] T. OMG., Meta object facility (MOF) 2.0 core specification, *OMG Available Specification*, 2005.