Research Journal of Applied Sciences, Engineering and Technology 7(17): 3514-3521, 2014
DOI:10.19026/rjaset.7.703
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: October 19, 2013 Accepted: November 21, 2013 Published: May 05, 2014

Research Article

A Qualitative Study of Domain Specific Languages for Model Driven Security

Muhammad Qaiser Saleem
College of Computer Sciences and Information Technology, Al Baha University, Al Baha, Kingdom of Saudi Arabia

Abstract: In Model-Driven development, software system design is represented through models which are created using general purpose modeling languages e.g., UML. Later on system artifacts are automatically generated from these models. Model-Driven Security is a specialization of Model-Driven paradigm towards the domain of security, where security objectives are modeled along the system models and security infrastructures are directly generated from these models. Currently available general purpose modeling languages like UML do not have capability to model the security objectives along the system models. Over the past decade, many researchers are trying to address these limitations of the general purpose modeling languages and come up with several Domain Specific Modeling Languages for Model Driven Security. In this study, a comparative study is presented regarding the security Domain Specific Modeling Languages presented by the most prominent researchers for the development of secure system. A success criteria has been defined and these DSLs are critically analyzed based on it to obtain the qualitative results.

Keywords: Domain specific language, model driven security, model driven software development, software modeling languages

INTRODUCTION

During software modelling, concentration is towards modelling the functional correctness in the model; usually notion of security is often neglected. It may happen due to many reasons, one of the prominent reason is that the currently available software system modelling languages e.g., UML, do not have ability to capture security objectives (Christian *et al*., 2008). In practice security objectives are specified in a non-formalize way by the business department normally as an unstructured text. If these security specification are not understood by the IT security department; a complicated and error prone coordination process between both departments arises; resulting loss of requirement sovereignty by the business department which is owner of the application (Klarl *et al*., 2009). As ggeneral purpose modelling languages like UML; do not have capability of modelling the security objectives along modelling of the software system. There should be some formal means through which security would be modelled in a software model for secure software application development. Having this very essential aspect in mind, several researching are proposing Domain Specific Languages for modeling security along the software model. Different researches focused different security objectives for their DSLs. Normally they proposed a meta model for the definition of the abstract syntax of the DSL. Later on they used different general purpose modeling languages for the definition of concrete syntax. After the definition of the DSL, the general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotations.

BACKGROUND MATERIAL AND RESEARCH METHOD

In this section, initially background literature is presented in which two basic concepts i.e., Model Driven Security (MDS) and Domain Specific Language (DSL) are discussed which are necessary to build the background knowledge of the current work. Later on; discussions are provided regarding the research method used during this study.

Model driven security: MDSD is one of the most promising software engineering approaches. Object Management Group (OMG) has presented a framework known as Model Driven Architecture (MDA) (OMG, 2011), which is considered as implementation of Model Driven Engineering (MDE). In MDA framework, software systems are modeled using general purpose modeling language like UML, as a Platform Independent Model (PIM) and then it is transformed into other Platform Independent Model (PIM), Platform Specific Model (PSM) or Implementation Specific Model (ISM). In MDA framework, rather than just a visual aid, models are considered as essential part of software definition (Alam, 2007a; Rodríguez *et al*., 2007b).

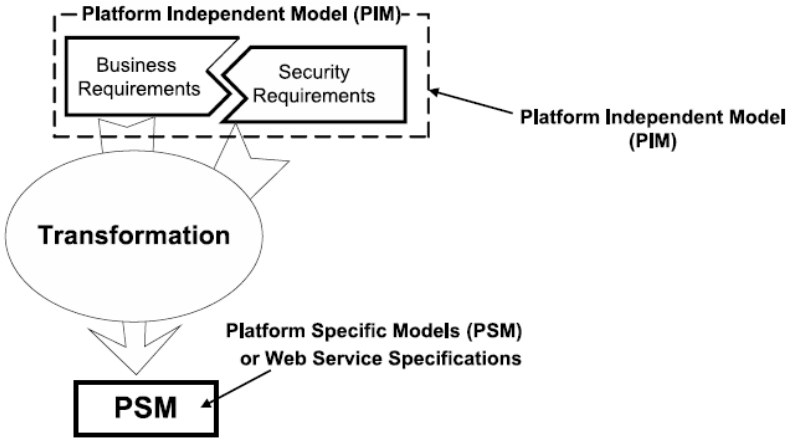

Fig. 1: MDA pattern with security extension (Alam, 2007a)

"Model driven security is an engineering paradigm that specializes Model Driven Software Development towards Information Security" (Michal and Ruth, 2009). The MDS is based on the MDSE and MDA where security requirements are realized at the model level and kept separate from the underlying security architecture. The MDS is an engineering discipline which is concerned with the integration of security requirements in all system development phases e.g., analysis, design, implementation, testing etc., (Alam, 2007b). The vision of the MDS is to provide a way for software engineers to bridge the gap between the system design requirements and security requirements by taking a model-centric approach. This in turn necessitated bridging the gap between security modelling languages and design modelling languages, leading to the notion of security-design modelling languages, such as the Secure UML (Basin *et al*., 2011). In the MDS, security requirements are defined as a model during the designing phase and concrete security configuration files can be generated by the model transformation e.g., security concepts are modelled side by side with the business process modelling at the PIM level of abstraction and step-wise refined to further levels of abstraction i.e., Platform Specific Model (PSM) and Implementation Specific Model (ISM) (Basin *et al*., 2006; Alam, 2007b; Satoh *et al*., 2008; Christian *et al*., 2009; Michal and Ruth, 2009). Figure 1 illustrates the whole process.

These security objectives are defined in the model with the help of a DSL and transformed into enforceable security rules with a little human intervention. The MDS is a critical component of future Information Assurance (IA) architectures, especially for agile IT environments such as SOA (Ulrich Lang, 2009).

Doman specific language: Application structure, requirements and behavior according to a specific domain are formalized in the form of a DSL which is one of the components of the Model Driven Software Development (MDSD) (OMG, 2011).

A domain can be defined as "a field of application delimitated by a specific area of interest" (Michal and Ruth, 2009). A DSL is defined as "A concise, precise and process-able description of a viewpoint, concern or aspect of a system, given in a notation that suits the people who specify that particular viewpoint, concern or aspect" (Michal and Ruth, 2009).

The DSL is used to formalize a modelling language capable of formalizing different business domains (like e-government, e-health, e-education), system aspects (like security, real-time) or concrete technologies (such as EJB or .NET) (Basin *et al*., 2006). A DSL consists of constructs that capture information regarding the domain it describes (France and Rumpe, 2007). DSLs are small and provide a basis for domain-specific formal analysis and use those notions which are familiar to domain experts (Achim and Brucker, 2007). A DSL may also be called a Domain Specific Modelling Language (DSML) (Tomaž Lukman, 2008).

DSLs are developed using the OMG's (Object Management Group) MOF (Meta Object Facility) meta-modelling mechanism (France and Rumpe, 2007; Giovanni *et al*., 2009). These extension techniques are metamodel based techniques and known as heavy weight extension mechanisms. The metamodel based technique of defining the DSL is mostly used when the "domain" is well defined and has an 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 (Basin *et al*., 2006).

There is no universal approach for the integration of security and design modelling languages (Menzel and Meinel, 2010; Saleem *et al*., 2012a; Lodderstedt, 2004). The current practice of defining a DSL by different researchers (Jürjens, 2002; Basin *et al*., 2006; Rodríguez *et al*., 2006a, 2007a; Selic, 2007; Saleem

et al., 2012b; Christian *et al*., 2008) is that the abstract syntax of the DSL is represented by a metamodel and the concrete syntax is represented by a UML Profile.

Following are the basic concepts of a DSLs (Michal and Ruth, 2009).

Abstract syntax: Defines the basic concepts, their relationships and the integrity constraints of a DSL (Kai *et al*., 2005) e.g., in the OMG's metamodel architecture, the UML Class diagram at the M2 level of abstraction (Atkinson and Kuhne, 2003). Normally abstract syntax is defined through a metamodel (Michal and Ruth, 2009).

Concrete syntax: Defines the notion of the language, which will be used during modelling i.e., the front end of the DSL. These notions may be visual or textual (Michal and Ruth, 2009). For example UML notations (Atkinson and Kuhne, 2003).

Semantic: A modeling language defines its meaning in context. Semantics are either defined formally or should at least be documented in an informal way (Michal and Ruth, 2009). For example, the natural language specification (Atkinson and Kuhne, 2003).

Research method used during this study: During this study, qualitative methods are used to collect and analyze the qualitative data. Qualitative data is normally in the form of pictures, words, statements, description and diagrams. The process followed to collect them are ethnographies, case studies and interviews (Abbas and Charles, 1998). In qualitative methods the focus is more towards the collecting and analyzing the non-numeric data and information are explored in depth rather than in breath (Loraine *et al*., 2001). Qualitative data is analyzed using categorization and sorting (Runeson and Höst, 2009). Qualitative research explores attitudes, behavior and experiences and the research methodologies used are: Phenomenology, Ethnography, Case studies, Interviews, Action Research, Grounded Theory (Dawson, 2002; Cresswell, 2009).

During this study, an explanation building technique Yin (2003) is adopted that support in comparative analysis of existing research work. In explanation building, many different kinds of evidence, figures, statements, documents etc., are linked together to support a strong and relevant conclusion (Runeson and Höst, 2009). During this research work, figures (security annotated business process diagrams) and statements are used as an evidence to support the conclusion. In order to analyze the results, a comparative study has been conducted. The outcome of the comparative study is represented in the form of statements as well as table.

QUALITATIVE STUDY OF DSLS FOR MDS

From through literature review, it is revealed that, the different DSLs proposed by different researchers, adopted different approaches to annotate the business process with security. Some are proposing icons (Rodríguez *et al*., 2006b; Christian *et al*., 2009), while on the other hand, some are just proposing security stereotypes (textual description) (Michal *et al*., 2006; Alam, 2007a; Mukhtiar *et al*., 2008). Furthermore, few are proposing multiple diagrams to represents the business process model (Michal *et al*., 2006; Christian *et al*., 2008). Moreover, the number of security objectives presents in different security DSLs, are also different. These are the factors which affect a business process model and regarding them the data is collected and analysis is performed.

In this study a comparative study has been conducted by following the guidelines presented by (Roy Grønmo, 2004). In this study, DSLs presented by different researchers are compared by considering the numbers of factors such as: simplicity and readability of business process model, ease of use by business process expert, use of icons in a DSL to represent the security objectives and sufficient numbers of security objectives for SOA applications. The outcome of the comparative study is represented in the form of statements as well as table. Thorough discussion is provided below according to specific factors and Table 1 depicts the summary of the discussion.

Simplicity of the business process model: In this section, discussions are provided regarding the approaches followed in different research work for security annotation to find out that the model created using a particular approach is either simple or complex.

A business process model should be simple; it should not be messy with lots of technical details. If it contains lots of technical details then the model would be complex for a common business process expert.

In the study presented by Saleem *et al*. (2012a) a single business process model is developed and security is annotated within the model with the help of their DSL named "UML-SOA-Sec". In their approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. Security objectives are represented in the business process model as stereotypes. In this way only one diagram is created. It does not make the business process model such a mess; instead, it keeps the model simple so a business process expert can easily understand and work with it.

The approach presented by Ruth *et al*. (2003), Michal *et al*. (2006) and Michal and Ruth (2009) proposes total of five diagrams to represent the security

Table 1: Qualitative comparison of DSLs based on success criteria

Researchers	Work	Simplicity of business process model	Readability of business process model	Ease of use by business process expert	Use of icons for security objective	Sufficient number of security objectives for SOA applications
Saleem *et al*. (2012a)	A DSL called UML-SOA-sec. focusing confidentiality, integrity, availability, auditing, non-repudiation	Yes	Easy	Easy	Yes	Yes
Michal and Ruth (2009), Ruth *et al*. (2003), Michal *et al*. (2006) and Michal and Ruth (2009)	Security policies regarding confidentiality, integrity and availability	Little bit complex (they are preparing total of five models, two models under workflow and three under interface model)	Little bit difficult	Little bit harder for a business process expert	No	Almost yes (however they are using three broad categories of generic security objectives)
Memon (2011)	Security pattern for authentication and non-repudiation	Little bit complex (they are preparing five models, two models under workflow and three under interface model)	Little bit difficult	Little bit harder for a business process expert	No	No
Rodríguez *et al*. (2006b), Rodríguez *et al*. (2006a) and Rodríguez *et al*. (2007a)	A DSL focusing different security objectives non-repudiation, attack harm detection, integrity, privacy access control	Yes	Easy	Easy	Yes	Almost yes (however they did not mentioned the SOA architecture)
Christian *et al*. (2008)	Security policies for confidentiality, integrity, authentication, authorization, availability and audit	Little bit complex (they are presenting a business process model in term of three layers; business process, organization and integration)	Little bit difficult	Little bit harder for a business process expert	Yes	Almost yes
Menzel *et al*. (2009)	Security policies for authentication, authorization, trust, data integrity and data confidentiality, system integrity and system availability	Yes	Little bit difficult	Easy	Yes	Almost yes

annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two diagrams are created called Global Workflow Model and Local Workflow Model. While in Interface View; three diagrams are created known as Interface Model, Role Model and Document Model. It means a total of five diagrams are created. Mukhtiar *et al*. (2008) and Memon (2011) also used the same approach in his work. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. Furthermore, if someone spend too much time in documentations and especially focusing only one non-functional requirement i.e., "Security", then it would be very difficult to cope with the other functional and non-functional requirements. Hence, a security annotated business process model created using their approaches is little bit complex for a common business process expert to understand and work with it.

In their approach, Rodríguez *et al*. (2006b, 2007a) are constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e., BPMN (Rodríguez *et al*., 2007a) as well as the UML (Rodríguez *et al*., 2006a, b). In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is simple.

Christian *et al*. (2008), in their approach is also constructing a single business process model and annotate it with security requirement. However, in their work the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. Although, they are constructing a single business process model and annotate it with security; however, in their approach security has to be defined in terms of three layers i.e., business process layer, organization layer and integration layer. Hence, a security annotated business process model created using their approaches is little bit complex for a common business process expert to work with it.

Menzel *et al*. (2009), in their approach is also creating a single business process model and security is annotated with their proposed DSL Secure SOA. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is simple.

Readability of the business process model: In this section, discussions are provided regarding the approaches followed in different research work for security annotation to find out that the model created using a particular approach is either easily readable or difficult to read.

A business process model should be easily readable. It should not be messy with lots of technical details. If it contains lots of technical details then it will affect its readability for a common business process expert.

In the work presented by Saleem *et al*. (2012a) a single business process model is developed and security is annotated within the model with the help their DSL

named "UML-SOA-Sec". In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. Security objectives are represented in the business process model as stereotypes. Hence, the business process model developed using their approach is easily readable.

The approach presented by Ruth *et al*. (2003), Michal *et al*. (2006) and Michal (2009) proposes total of five diagrams to represent the security annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two diagrams are created called Global Workflow Model and Local Workflow Model. While in Interface View; three diagrams are created known as Interface Model, Role Model and Document Model. It means a total of five models are created. Mukhtiar *et al*. (2008) and Memon (2011) also used the same approach in his work. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. Hence, a security annotated business process model created using their approaches is little bit difficult in terms of readability.

Rodríguez *et al*. (2006a, b, 2007a), in their approach are also constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e., BPMN (Rodríguez *et al*., 2007a) as well as the UML (Rodríguez *et al*., 2006a, b). In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily readable for a common business process expert.

Christian *et al*. (2008), in their approach is also constructing a single business process model and annotate it with security requirement. However, in their approach, the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model little bit difficult in terms of readability.

Menzel *et al*. (2009), in their approach is also creating a single business process model and security is annotated with their proposed DSL Secure SOA. However, their focus is security policies which contain more technical details and make the business process model little bit difficult in terms of readability.

Ease of use by business process expert: In this section, discussions are provided regarding the approaches followed in different research work for security annotation to find out that a common business process expert can easily create a security annotated business process model using a particular approach or not.

The business process model should be easily understandable for a common business process expert so that he/she should easily work with it. A business process expert is not a security expert. Although he/she is familiar with common security notions, it is not reasonable to expect too much security knowledge from him/her to build a security policy or a security pattern.

In the work presented by Saleem *et al*. (2012a) a single business process model is developed and security is annotated within the model with the help of their proposed DSL "UML-SOA-Sec". In their approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. It does not require the deep understanding of how the security objectives would be realized through which security policy or security pattern. Hence, a common business process expert can easily use their proposed DSL for security annotation.

The approach presented by Ruth *et al*. (2003), Michal *et al*. (2006) and Michal and Ruth (2009) proposes total of five diagrams to represent the security annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two diagrams are created called Global Workflow Model and Local Workflow Model. While in Interface View; three diagrams are created known as Interface Model, Role Model and Document Model. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. Mukhtiar *et al*. (2008) and Memon (2011) also used the same DSL in his work. Furthermore, it also required a business process expert to have a strong knowledge of security patterns as well. Hence, it is also harder for a common business process expert to work with it as it required a business process expert to create five models to represent the security annotated business process model.

Rodríguez *et al*. (2006a, b, 2007a), in their work are also constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e., BPMN (Rodríguez *et al*., 2007a) as well as the UML (Rodríguez *et al*., 2006a, b). In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily useable for a common business process expert.

Christian *et al*. (2008) in their work are also preparing a single business process model and annotate it with security requirement. However, in their approach, the business process model is constructed in

terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model little bit difficult to use for a common business process expert. Furthermore, it also required the knowledge of security policies to work with it. Hence it is difficult for a common business process expert to work with their approach.

Menzel *et al*. (2009), in their approach are also creating a single business process model and security is annotated with their proposed DSL Secure SOA. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily useable for a common business process expert.

Use of icons to represent the security objectives: In this section, discussions are provided to represent that either icons are used or not to represent the security objectives in the research works.

Two kinds of approaches are adopted by the researchers to represent the security objectives, icons (graphical notation) and textual description. Few researchers are just using text to represent the security objectives. Although in this way a business process model can be annotated with security objectives; however, graphical representation of security objectives i.e., icons, facilitate the business process expert to incorporate the security objectives in the business process model in an easier way.

Meaningful icons are provided in the approaches proposed by the Saleem *et al*. (2012b), Rodríguez *et al*. (2006a, b, 2007b), Christian *et al*. (2008) and Menzel *et al*. (2009) to represents the security objectives which facilitate a common business process expert to add security in the business process model. These icons are available at design time to incorporate security objectives in the business process models.

While, Ruth *et al*. (2003), Michal *et al*. (2006), Michal and Ruth (2009), Mukhtiar *et al*. (2008) and Memon (2011) are not using icons to represents the security objectives, they are just using textual description to represents the security objectives.

Sufficient number of security objectives for SOA applications: In this section, discussions are provided regarding the number of security objectives presentedin the different research work is sufficient for SOA environment or not.

SOA applications are basically distributed applications which required securing both data as well as service. Number of security objectives present in a DSL is very important because it represents that how much a DSL satisfy the security requirements of an SOA environment. A thorough discussion is presented by the Saleem *et al*. (2012b) regarding the necessary security objectives of SOA environment.

In the work presented by Saleem *et al*. (2012b), five security objectives which are necessary for modelling along the business process modelling of SOA applications are picked. The five security objectives are Confidentiality, Integrity, Availability, Auditing and Non-repudiation. Afterwards two security mechanisms, Authentication and Authorization are described through which these security objectives would be realized. The security objectives present in their proposed DSL are sufficient for a SOA application.

The work presented by Rodríguez *et al*. (2006a, b, 2007b), contains the five security objectives naming Non-repudiation, Attack Harm detection, Integrity, Privacy, Access Control. These security objectives can be sufficient of SOA environment, however authors did not mentioned anything regarding the target architecture weather it is an SOA environment or not.

Michal *et al*. (2006), Ruth *et al*. (2003) and Michal and Ruth (2009) are dealing with three security objectives namely Confidentiality, Integrity and Availability. These security objectives are sufficient for SOA environment as authors kept themselves at very abstract level.

The DSLs presented by the Christian *et al*. (2008) contains the six security objectives naming Confidentiality, Integrity, Authentication, Authorization, Availability and Audit. These security objectives are sufficient for SOA environment.

Menzel *et al*. (2009) Authentication, Authorization, Trust, Data Integrity and Data Confidentiality, System Integrity and System Availability. These security objectives are also sufficient for SOA environment.

Memon (2011) is dealing with only two security objectives, i.e., Authentication and Non-repudiation. They have defined the patterns for these security objectives. These two security objectives are not sufficient for securing the SOA environment.

CONCLUSION

This study tried to compile the work of different researchers which are working in the area of MDS and presented different DSLs for security modeling along modeling of the different aspect of software systems. A critical evaluation of these DSLs are presented discussing the weaknesses and strength of these DSLs. Afterwards qualitative results are presented to show the comparison of these DSLs. We believe our efforts will facilitate the practitioners in selecting the most suitable DSL for their work. Our efforts will also facilitate the beginners in this area to get a picture of already work done in this area, which will serve him/her as a basis for understanding the area of MDS and DSL and provide basis for further improvements in the said areas.

REFERENCES

Abbas, T. and T. Charles, 1998. Mixed Methodology Combining Qualitative and Quantitative Approaches. SAGE Publications, Thousand Oaks, Calif.

Achim, D. and J.D. Brucker, 2007. Metamodel-based UML notations for domain-specific languages. Proceeding of 4th International Workshop on Language Engineering (ATEM, 2007).

Alam, M., 2007a. Model driven realization of dynamic security requirements in distributed systems. Ph.D. Thesis, University of Insbruck, Austria.

Alam, M., 2007b. Model Driven Security Engineering for the Realization of Dynamic Security Requirements in Collaborative Systems. In: Kuhne, T. (Ed.), MoDELS 2006 Workshops, LNCS 4364, Springer-Verlag, Berlin, Heidelberg, pp: 278-287.

Atkinson, C. and T. Kuhne, 2003. Model-driven development: a metamodeling foundation. IEEE Software, 20(5): 36-41.

Basin, D., J. Doser and T. Lodderstedt, 2006. Model driven security: From UML models to access control infrastructures. ACM T. Softw. Eng. Meth., 15(1): 39-91.

Basin, D., C. Manuel and E. Marina, 2011. A decade of model-driven security. Proceedings of the 16th ACM Symposium on Access Control Models and Technologies (SACMAT '11), pp: 1-10.

Christian, W., M. Michael and M. Christoph, 2008. Modelling security goals in business processes. Proceeding of GI Modellierung 2008. GI LNI 127, Berlin, Germany, pp: 197-212.

Christian, W., M. Michael, M. Christoph, S. Andreas and M. Philip, 2009. Model-driven business process security requirement specification. J. Syst. Archit., 55(4): 211-223.

Cresswell, J.W., 2009. Research Design: Quantitative, Qualitative and Mixed Methods Approaches. 3rd Edn., SAGE, Los Angeles.

Dawson, C., 2002. Practical Research Methods: A User Friendly Guide. How to Books, Oxford.

France, R. and B. Rumpe, 2007. Model-driven development of complex software: A research roadmap. Proceedings of Future of Software Engineering (FOSE '07), pp: 37-53.

Giovanni, G., M. Beatriz and P. Oscar, 2009. Integration of domain-specific modeling languages and uml through uml profile extension mechanism. Int. J. Comput. Sci. Appl. Technomath. Res. Found., 6(5): 145-174.

Jürjens, J., 2002. UMLsec: Extending UML for secure systems development. Lect. Notes Comput. Sci., 2460: 412-425.

Kai, C., S. Janos and N. Sandeep, 2005. Toward a semantic anchoring infrastructure for domain-specific modeling languages. Proceedings of 5th International Conference on Embedded Software (EMSOFT '05), pp: 35-43.

Klarl, H., C. Wolff and C. Emig, 2009. Identity management in business process modelling: A model-driven approach. Proceeding of 9th International Conference on Business Computer Science Concepts, Technologies, Applications. Vienna, (German), Feb. 25 -27.

Lodderstedt, T., 2004. Model driven security: From UML models to access control architectures. Ph.D. Thesis, Albert-Ludwig University of Freiberg, Germany.

Loraine, B., H. Christina and T. Malcolm, 2001. How to Research. 2nd Edn., Open University Press, Buckingham, Philadelphia.

Memon, M., 2011. Security Modeling with pattern refinement for security-as-a-service architecture. Ph. D. Thesis, University of Insbruck, Austria.

Menzel, M. and C. Meinel, 2010. SecureSOA modelling security requirements for service-oriented architectures. Proceeding of IEEE International Conference on Services Computing (SCC), pp: 146-153.

Menzel, M., I. Thomas and C. Meinel, 2009. Security requirements specification in service-oriented business process management. Proceeding of International Conference on Availability, Reliability and Security (ARES '09), pp: 41-48.

Michal, H. and B. Ruth, 2009. Security Engineering for Service-oriented Architectures. Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-79538-4.

Michal, H., B. Ruth, A. Berthold and N. Andrea, 2006. SECTET: An extensible framework for the realization of secure inter-organizational workflows. Internet Res., 16(5): 491-506.

Mukhtiar, M., H. Michael and B. Ruth, 2008. SECTISSIMO: A platform-independent framework for security services. Proceeding of Modeling Security Workshop (MODSEC, 08).

OMG, 2011. OMG Model Driven Architecture. Retreieved form: http://www.omg.org/mda/. (Accesed on: Oct. 30, 2011).

Rodríguez, A., F.M. Eduardo and P. Mario, 2006a. Security requirement with a UML 2.0 profile. Proceeding of the 1st International Conference on Availability, Reliability and Security (ARES 2006).

Rodríguez, A., F.M. Eduardo and P. Mario, 2006b. Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes. In: Fischer-Hubner, S. *et al*. (Eds.), TrustBus, 2006. LNCS 4083, Springer-Verlag, Berlin, Heidelberg, pp: 51-61.

Rodríguez, A., F.M. Eduardo and P. Mario, 2007a. A BPMN extension for the modeling of security requirements in business processes. IEICE T. Inf. Syst., E90-D(4): 745-752.

Rodríguez, A., F.M. Eduardo and P. Mario, 2007b. Towards CIM to PIM transformation: From secure business processes defined in BPMN to use-cases. Lect. Notes Comput. Sci., 4714: 408-415.

Roy Grønmo, I.S., 2004. Towards modeling web service composition in UML. Proceeding of the 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure. Porto, Portugal.

Runeson, P. and M. Höst, 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng., 14(2): 131-164.

Ruth, B., B. Klaus, H. Michael, J. Jan, W. Guido and L. Volkmar, 2003. Key issues of a formally based process model for security engineering. Proceedings of the 16th International Conference on Software and Systems Engineering and their Applications (ICSSEA, 03).

Saleem, M.Q., J. Jaafar and M.F. Hassan, 2012a. Secure business process modelling of SOA applications using UML-SOA-Sec. Int. J. Innov. Comput. I., 8(4): 2729-2746.

Saleem, M.Q., J. Jaafar and M.F. Hassan, 2012b. Model-based security engineering of SOA systems using modified "UML-SOA-Sec. Adv. Inf. Sci. Serv. Sci. (AISS), Int. J. Res. Innov., 4(9): 79-88.

Satoh, F., Y. Nakamura, N.K. Mukhi and M. Tatsubori, 2008. Methodology and tools for end-to-end SOA security configurations. Proceeding of IEEE Congress on Services- Part I. Honolulu, HI, pp: 307-314.

Selic, B., 2007. A systematic approach to domain-specific language design using UML. Proceeding of 10th IEEE International Symposium on Object and Component-oriented Real-time Distributed Computing (ISORC '07), pp: 2-9.

Tomaž Lukman, M.M., 2008. Model-driven engineering and its introduction with metamodeling tools. Proceeding of 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint. Izola, Slovenia.

Ulrich Lang, R.S., 2009. Top SOA Security Concerns and OpenPMF Model-driven Security. Object Security White-paper, Topics Cloud Computing and Security Management.

Yin, R.K., 2003. Case Study Research Design and Methods. 3rd Edn., Sage, Thousand Oaks, CA.