A Model Driven Engineering Approach to Support the Development of Secure Software as a Service

Pablo Matos, Denivaldo Lopes*, Zair Abdelouahab
Federal University of Maranhão - UFMA, CCET, PPGEE, São Luís — MA, Brazil.

* Corresponding author. Tel.: +55 98 3272 8243; email: denivaldo.lopes@ufma.br
Manuscript submitted September 22, 2015; accepted December 5, 2015.
doi: 10.17706/jsw.11.2.118-132

Abstract: The development and use of software based on cloud computing have been highlighted more and more nowadays. Software as a Service (SaaS) is considered as a trend for small, medium and large enterprises. However, SaaS brings some challenges concerning information security. In this context, we propose an approach based on Model-Driven Engineering (MDE), weaving models and security techniques for developing secure software as a service. An illustrative example helps to understand our proposed approach and demonstrates its feasibility and benefits.

Key words: Model driven engineering, model transformations, weaving model, cloud computing.

1. Introduction

In the context of cloud computing, Software as a Service (SaaS) provides some benefits such as needless to install a software locally and resource provision on demand [1]. Furthermore, it guarantees the use of updated releases.

SaaS is widely used by enterprises and customers in different applications scenarios, for instance, the Gmail mail service provided by Google. However, security is an important issue in SaaS. In this regard, we can talk about the lack of sophisticated resources for cloud service providers in order to integrate its platform and identity verification services placed behind firewalls of a corporate network [2]. Another relevant point in security of SaaS is the integrity of data that must be guaranteed by the SaaS provider, e.g. assuring integrity based on encryption algorithms and authentication systems [3]. Furthermore, security of SaaS must take into account the data transmission security over the Internet. In the literature, solutions for security in cloud computing are concentrated only in final source code, e.g. algorithms for security and security Application Programming Interface (API). However, there is an absence of discussion about approaches to take into account security in SaaS from the design to the final code.

In this paper, we provide an approach that takes care of security in SaaS from design to code. This approach is based on Model Driven Engineering (MDE) and security mechanism to support SaaS development. MDE is an approach where software development is based on models [4]. Here we present the idea to aggregate security mechanisms, for instance, role-based access control systems, into SaaS applications conceived as models. The security concerns are usually managed as an isolated feature in the software design process, but in our approach we take care of security from the beginning thanks to models and model transformation definitions. Security concerns are modeled from the beginning together with the business model following a similar Y development process presented in [5], and these security concerns are

propagated in other low level models thanks to transformation definitions until to obtain the source code.

This paper is organized as follows. Section 2 presents our approach based on MDE, weaving models and security techniques, including a framework and a methodology. Section 3 shows a prototype implementing the proposed approach. Section 4 depicts an illustrative example that helps to understand the proposed approach. Section 5 shows related works and compares them with our approach. Section 6 contains a conclusion and trends for future research works.

2. An Approach Based on MDE to Support the Development of Secure SaaS

The proposed approach aims to meet some security goals for SaaS as shown in Table 1. Security models can define aspects such as authentication, access control, encryption and persistence. Due to page limits, we present only authentication and access control in this paper.

Table 1. Security Goals for SaaS Applications (Based on [6])

Security goals	Security Requirements	Policy type
Confidentiality and integrity of stored information	Access to resources; properties and operations must be controlled	Access control policies based on user roles
	To use a system, a user must be authenticated	Authentication policies
Confidentiality and integrity of information in transmission	The contents of exchanged messages must be kept confidential	Message-level confidentiality policies
	The content of messages must be verified whether it has been modified during its transmission	
Confidentiality of intercepted information	The contents of exchanged messages must be transformed from its original form to another unreadable	Data encryption policies
Persistence of the information stored	The contents must be stored persistently	Data serialization policies

Fig. 1 presents a Framework for Developing Secure SaaS (FD3S) that is based on MDE, Weaving and Y development process. At the top, we find the Platform Independent Model (PIM) and Platform Description Model (PDM) that are related by the weaving model following a Y development process suggested in [5]. A PIM conforms to a metamodel as UML metamodel, and a PDM conforms to a metamodel as our proposed Security metamodel. In the center, a weaving tool aids with the creation and edition of a weaving model that is conform to a Weaving metamodel. Once the weaving model is done, an Intermediate Model Generator takes as inputs a PIM, a Weaving Model and a PDM and gives as output an Intermediate Model. This last model merges the information from the PIM, Weaving Model and PDM. An Intermediate Model is needed before the generation of a PSM, because the former contains the merged information from PIM, weaving model and PDM. Afterward, a Transformation Engine takes as input a Model-to-Model Transformation Definition, an Intermediate Model and gives as output a PSM. This Model-to-Model Transformation Definition relates at the top an Intermediate Metamodel and at the bottom another metamodel for instance a Web Service Metamodel. Then, a Transformation Engine takes as input another Model-to-Model Transformation Definition and an abstract PSM and gives as output a concrete PSM. This Model-to-Model Transformation Definition relates at the top to a Web Service Metamodel and at the bottom a Java Metamodel and the Pattern for applying Java API of Web Service. Then, another Transformation Engine takes as input a Model-to-Text Transformation Definition and a concrete PSM and gives as output a Source Code. This Model-to-Text Transformation Definition relates to a Java Web Service and Java API for Web

Service and an EBNF grammar of the Java programming language.

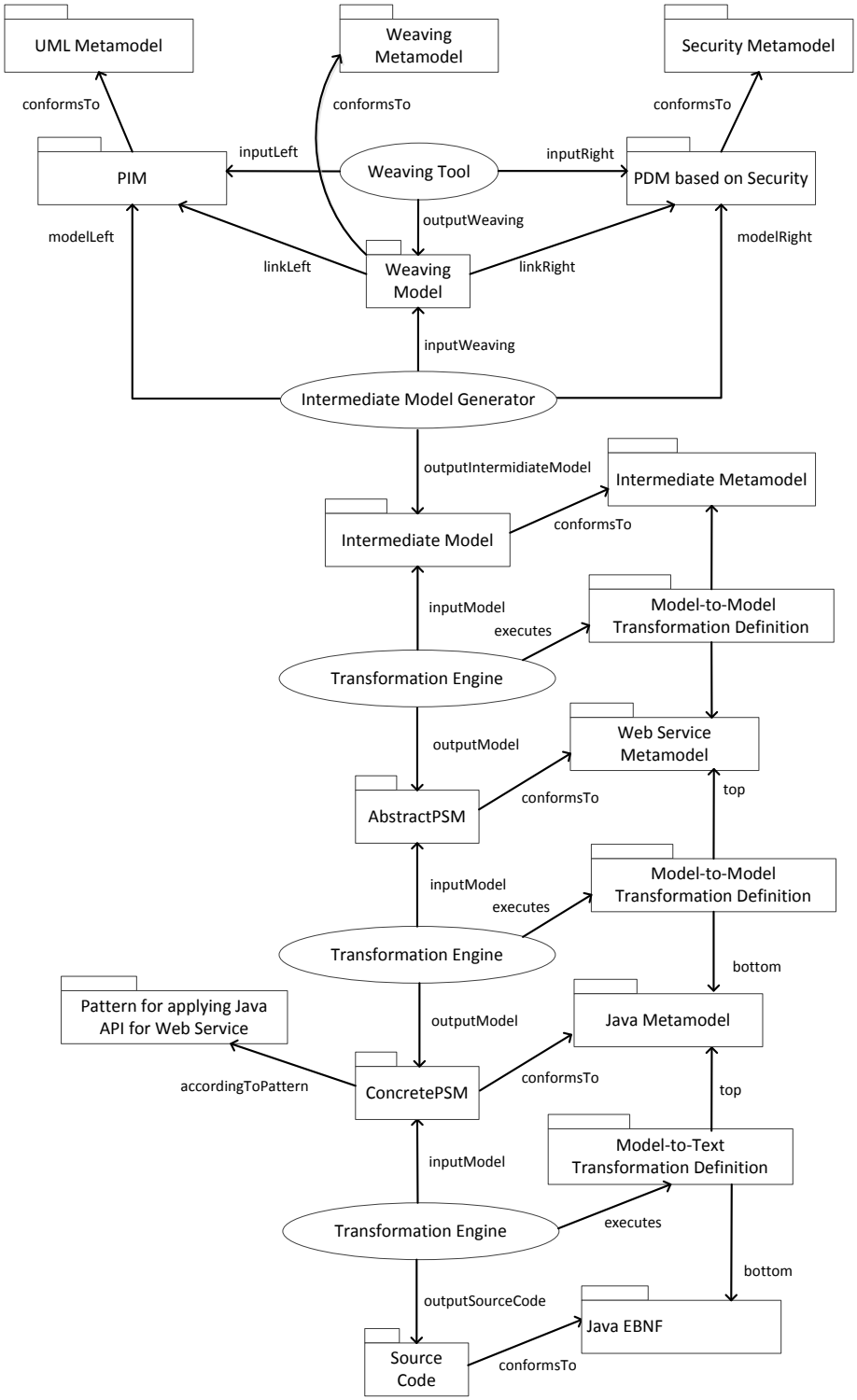

Fig. 1. Framework for developing secure SaaS (FD3S).

Our approach contains a methodology to develop secure SaaS as presented in Fig. 2. The steps of this methodology are presented as follows:

- Create a PIM: A business model (PIM) is created.
- Create a PDM (based on Security): This model contains security information, in other words, this PDM is a pattern for a security which may be reused many times.

- Create a Weaving Model: In order to relate a PIM and a PDM.
- Generate an Intermediate Model: That contains the merged information from PIM, weaving model and PDM.
- Execute a Transformation Definition M2M: In order to generate a PSM from an Intermediate Model. This is organized as follows. First, an M2M transformation takes as input an Intermediate Model and generates an Abstract PSM. The latter contains information about a platform, e.g. Web services. Second, another M2M transformation takes as input an abstract PSM and generates as output a concrete PSM. The generated model contains details about the implementation of a platform in a specific programming language. For this purpose, the concrete PSM reuses the model of an API, e.g. the model of Web services API in Java, and a pattern of the API usage.
- Edit a PSM: In order to include information specific to a platform that is not present in Intermediate Model, i.e. information which is not present in the PIM or in the PDM.
- Execute a Transformation Definition M2T: For creating a source code conforming to an EBNF grammar of a programming language.
- Edit a source code: In order to complete it and be ready for compilation.
- We provide a metamodel for creating Weaving Models, a metamodel for creating PDM and a metamodel for creating PSM. A PIM can be conform to UML metamodel that is available in UML specification [7].

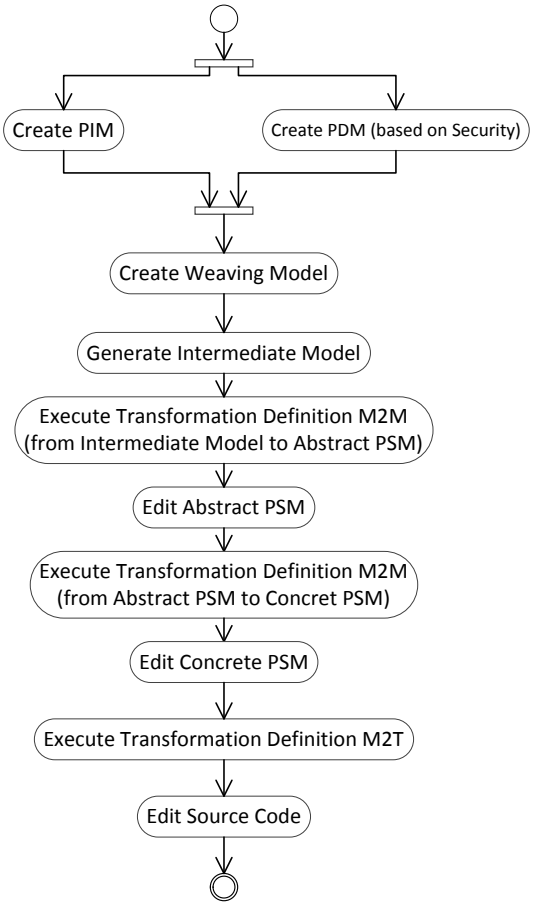

Fig. 2. A methodology to develop secure SaaS.

Fig. 3 presents a metamodel to create a weaving model. The metamodel extends and adapts the one presented in [8] for mapping specification in order to support the creation of a weaving model. Each

element of the metamodel is described as follows:

- Historic: Contains *WeavingDefinitions* and has the attributes update, note, version and timestamp in order to indicate the current *WeavingDefinition* and the versions.
- Weaving Definiton: Is a container for elements that define the weaving. For example, it contains *ModelHandlers* that point to models that are related, WeavingNode that relates two or more elements pointed by *linkLeft* and *linkRight*, i.e. *ElementModelHandler*.
- *ModelHandlers*: Is a pointer to models that are related by a weaving model. It contains *ElementModelHandler* that points to elements of a metamodel related to other elements of another metamodel.

Fig. 4 presents a metamodel to create a PDM for security aspects (focused on authentication and authorization) that is based on concepts presented in [9], [10]. This metamodel is described as follows:

- *RoleSpace*: is a container for Roles that are actions performed by *Users* according to *Permission*.
- *User*: contains *Property* and *Operation*. A *Property* defines static features and *Operation* defines behavior features.
- *Permission*: is related to constraints on *Actions* that *Resource* needs to perform. An *Action* can be read, updated, deleted and executed.

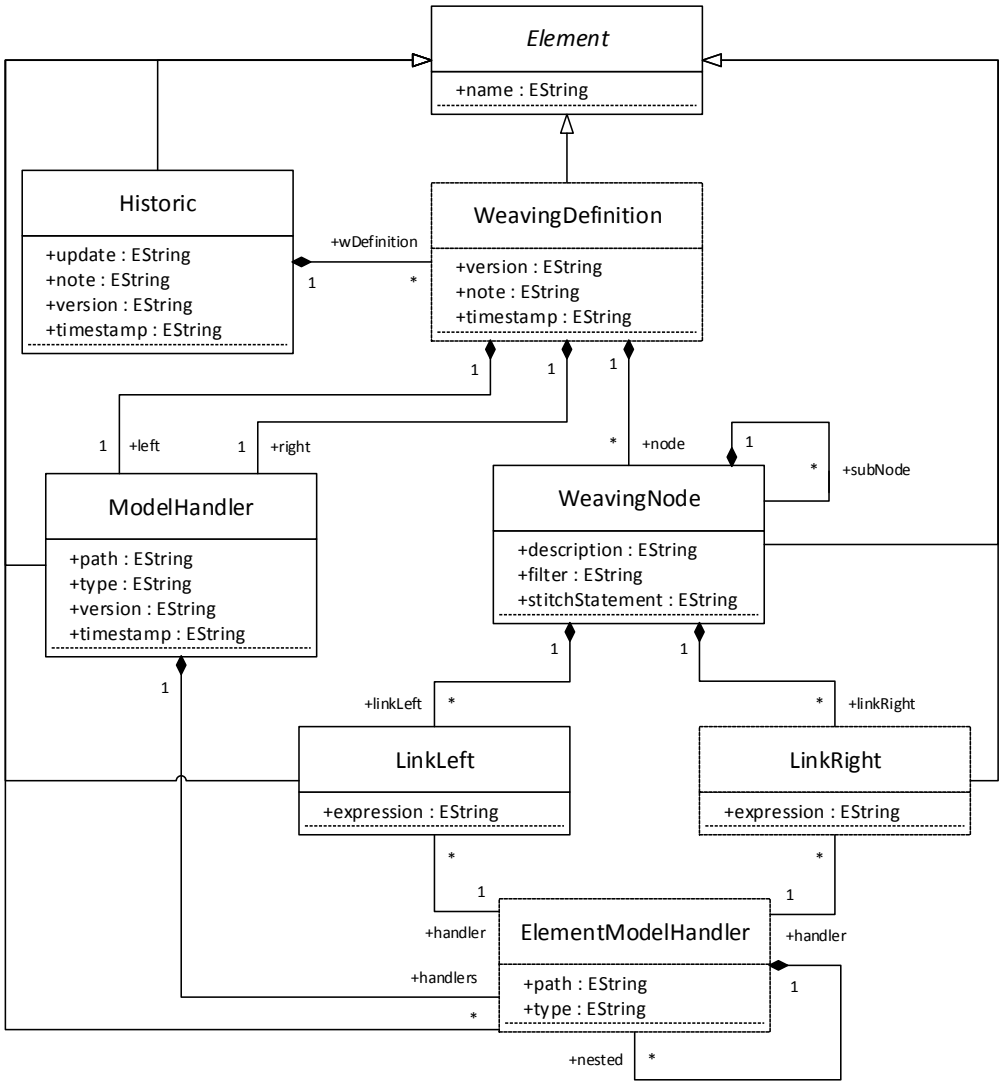

Fig. 3. A metamodel for weaving (based on [8]).

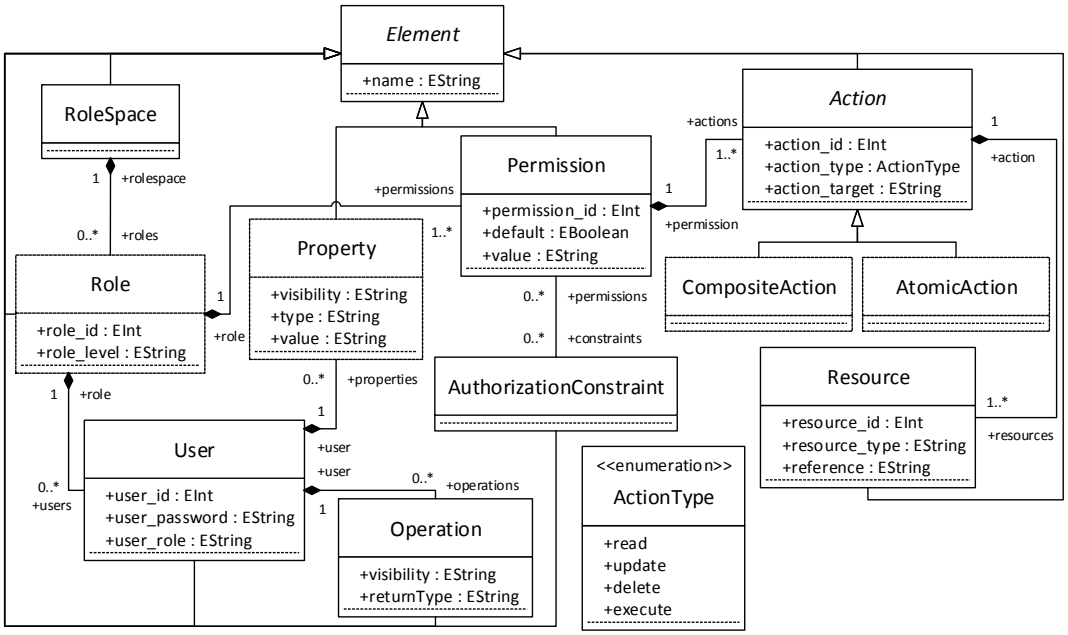

Fig. 4. A metamodel for defining a PDM based on security aspects: authentication and authorization (fragment) (based on [9], [10]).

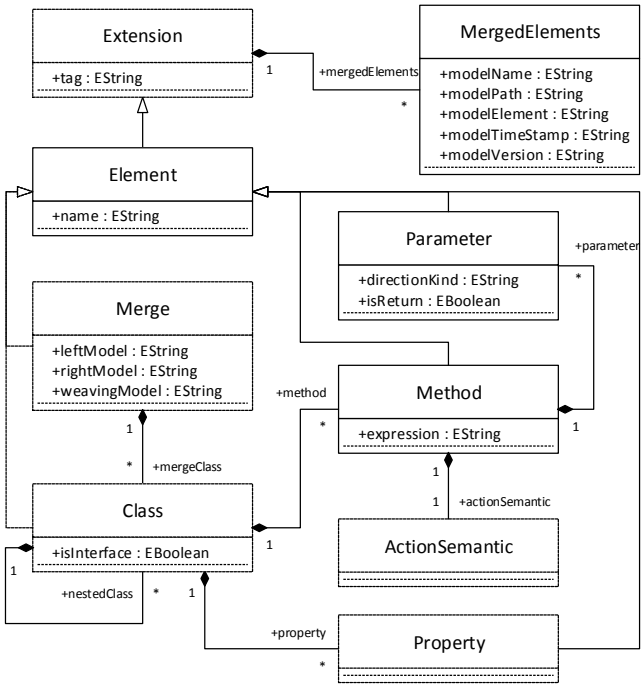

Fig. 5. Intermediate metamodel (fragment).

We propose the metamodel presented in Fig. 5 for creating an Intermediate model. The metamodel supports the creation of *Class*, *Property* and *Method* like to UML metamodel. We have enhanced the metamodel with the ability to define extensions containing *MergedElements* that are pointers to elements of other metamodels. In this way, we can create an intermediate model by merging a PIM, Weaving Model and PDM, but maintaining the link with the original elements from where intermediate elements were derived. Thus, *MergedElement* has a *name*, *path*, *timestamp* and *version* of a metamodel.

In order to create Abstract PSM based on Web Services, and the Concrete PSM based on Java plus API for

Web Services, we have reused, extended and adapted the metamodels proposed by D. Lopes *et al* [8] and J. Bézivin *et al.* [11].

3. Prototyping

Fig. 6 presents the architecture of the prototype that implements our proposed framework FD3S, methodology and metamodels.

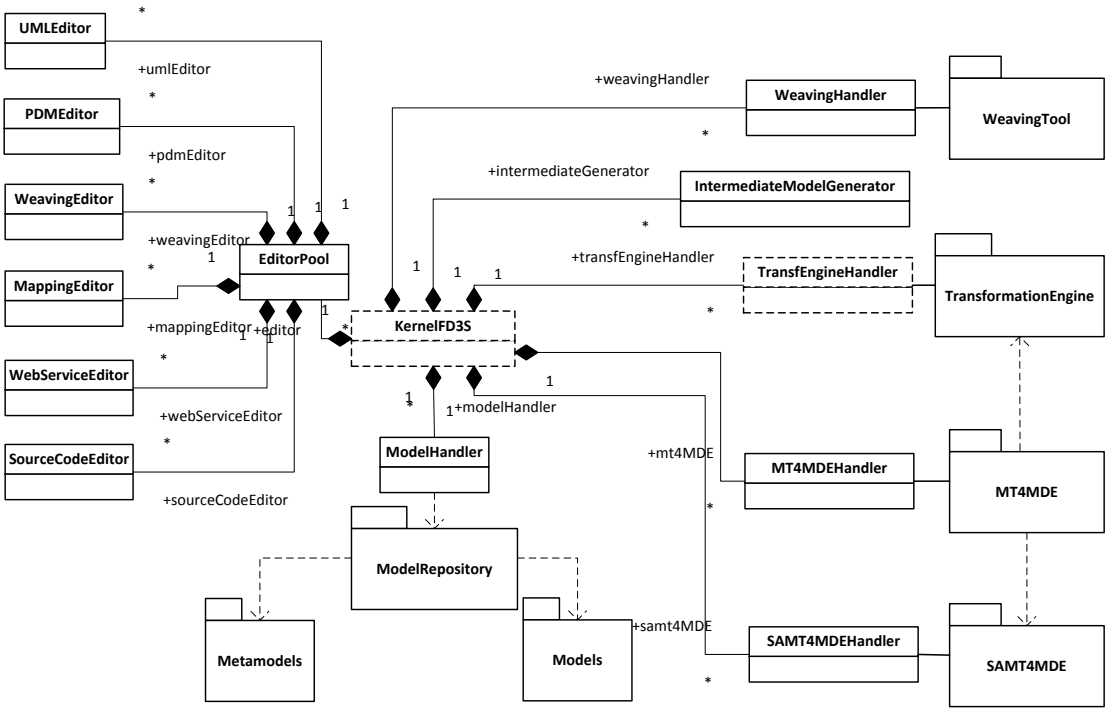

Fig. 6. Architecture of security weaving tool for MDE (SWT4MDE).

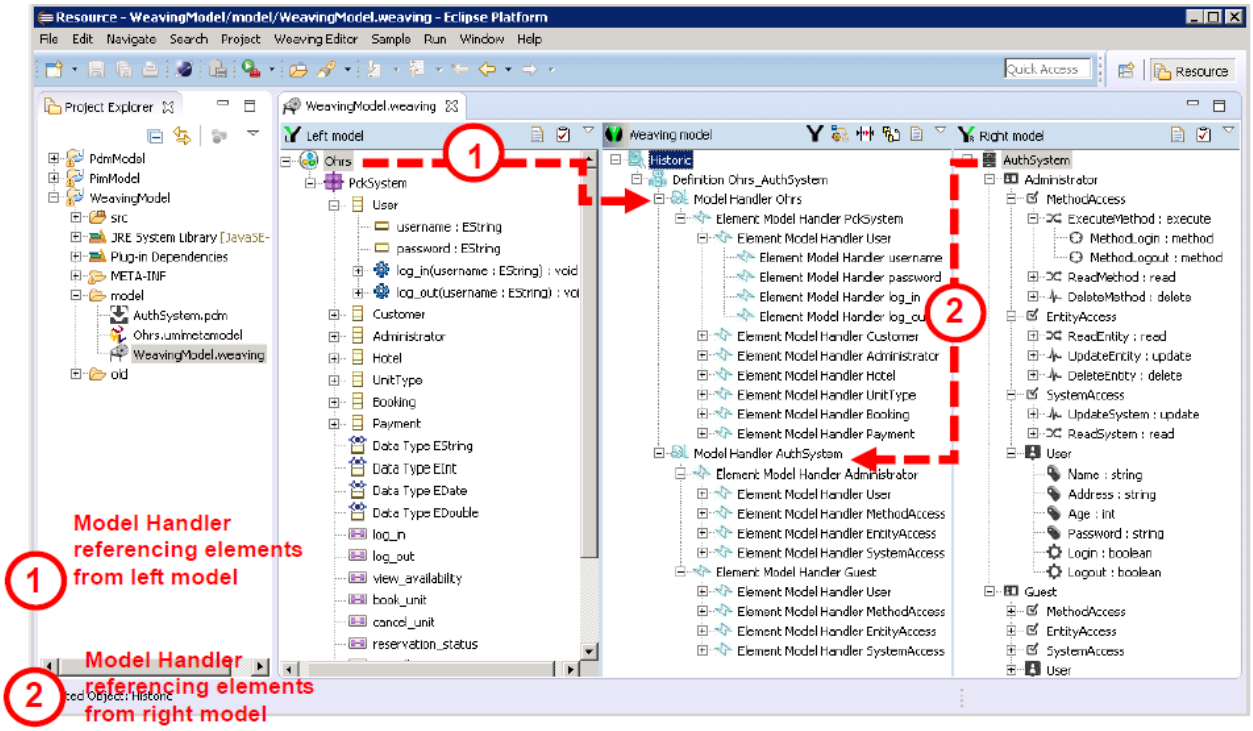

Fig. 7. Screenshots of the prototype of SWT4MDE.

The prototype Security Weaving Tool for MDE (*SWT4MDE*) implements our proposed FD3S as follows. *KernelFD3S* has the basic functionalities to start and contains handlers to integrate other tools such as weaving tools (i.e. *WeavingHandler*), transformation tools (i.e. *TransfEngineHandler*), mapping tool (i.e *MappingHandler*), matching tool (i.e. *MatchingHandler*). It integrates an editor pool containing editors for creating and editing UML models, Platform Description Model (PDM), Weaving models, mapping models, web services and source code in Java. *The KernelFD3S* manages the *ModelRepository* that includes metamodels and models.

The prototype *SWT4MDE* was developed combining the following software: Java SE Development Kit (J2SDK) version 7, Eclipse IDE version Juno Service Release 1, Eclipse Modeling Framework version 2.8.3 [12], ATL SDK for Eclipse version 3.3.1, Papyrus UML editor version 0.9.2, Netbeans version 8.0.2 (including Glassfish Server Open Source Edition 4.1), *MT4MDE* version 0.5.1[13] and *SAMT4MDE* version 0.8.2 [14].

We have developed the following parts of the prototype SWT4MDE: Weaving editor version 0.1.0, Weaving Tool version 0.1.0, PDM editor based on the metamodel presented in Fig. 4, Web Service editor based on the metamodel presented in [8], intermediate model generator based on the metamodel presented in Fig. 5, and transformation definitions presented in Listing 1.

Fig. 7 presents a screenshot of the prototype *SWT4MDE* implemented as a plug-in for IDE Eclipse. The PIM of OHRS, the weaving model and the PDM of security are presented on the left, center and right respectively. The weaving model has handlers to manipulate the left and right model elements.

4. Illustrative Example

In this section, we present the development of a software system as an illustrative example following our proposed approach. Fig. 8 presents the PIM for this illustrative example that is an "Online Hotel Reservation System (OHRS)" which is extracted and adapted from the works of Ritu Sharma and Manu Sood [15], [16].

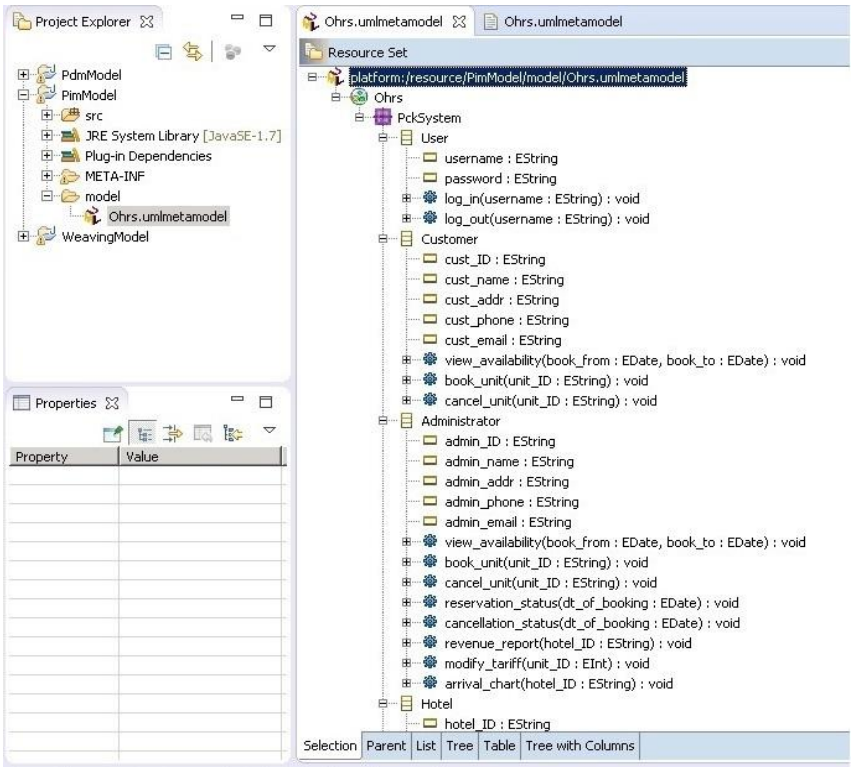

Fig. 8. PIM for OHRS (extracted and adapted from [15], [16]).

OHRS allows its customers to make reservation, payment and cancelation of hotel services. This software

system also allows a manager to perform procedures, for instance, modify prices and generate reports. The PIM for OHRS is the first input model. The PDM for security is presented in Fig. 9. The PIM is the business model while the PDM represents the security aspects.

Fig. 9 presents the PDM conforms to a metamodel based on security aspect described in Fig. 4. The PDM model depicts the *RoleSpace AuthSystem*. The *RoleSpace* contains the *Administrator* and *Guest* roles. These roles contain their respective users and permissions, e.g. permissions to access a method or entity. The permissions enable to perform actions, such as read, execute, and so on, over resources that can be entities, methods or attributes from the system.

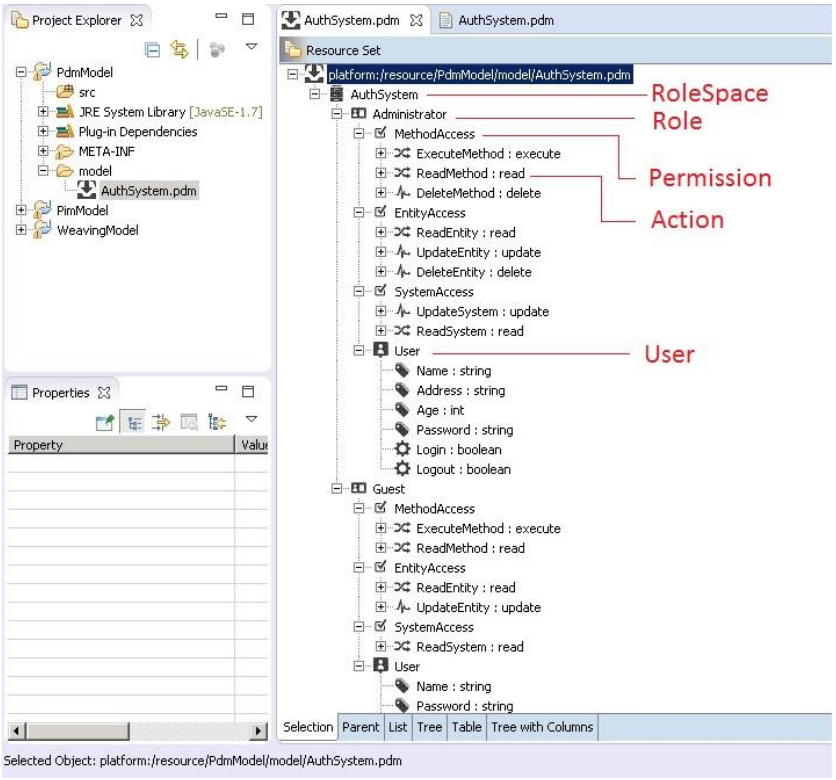

Fig. 9. PDM for security.

Once we have the PIM and the PDM, a weaving tool is used to create a weaving model that relates PIM and PDM. Fig. 10 presents the Weaving model relating the OHRS model (PIM) and the Security Model (PDM). At this point, the weaving model contains references to OHRS model and references to Security Model. The weaving model is created combining the elements from the PDM and the elements from the PIM.

An Intermediate Model is generated containing the aspects from business model (PIM), security aspect (PDM) and weaving model. Figure 11 presents the Intermediate Model that is the merging of OHRS model and Security model. For example, it associates features of the Roles *Administrator*/*Guest* from PDM with the classes Administrator/Customer from PIM, assigning them their features. It also associates features of Resources from PDM with the classes from PIM, such as Hotel, Booking and Payment, by merging them. An intermediate model is transformed in an abstract PSM using a model-to-model transformation definition executed in transformation engine like ATL [17].

Listing 1 presents a fragment of the transformation definition based on ATL from Intermediate Model to Abstract PSM. This transformation definition takes as input a model based on Intermediate Metamodel and gives as output a model based on Web Services Metamodel. The *Merge2Definition* rule transforms the *Merge* elements from the Intermediate Model to Definition elements of Web Services Model. The *Class2Service* rule

catches information of *Class* elements from the *Intermediate* Model to produce elements *Service*, *Port*, *Binding*, and *PortType* of Web Services Model. In the same way, other rules are created but they are omitted due to page limits.

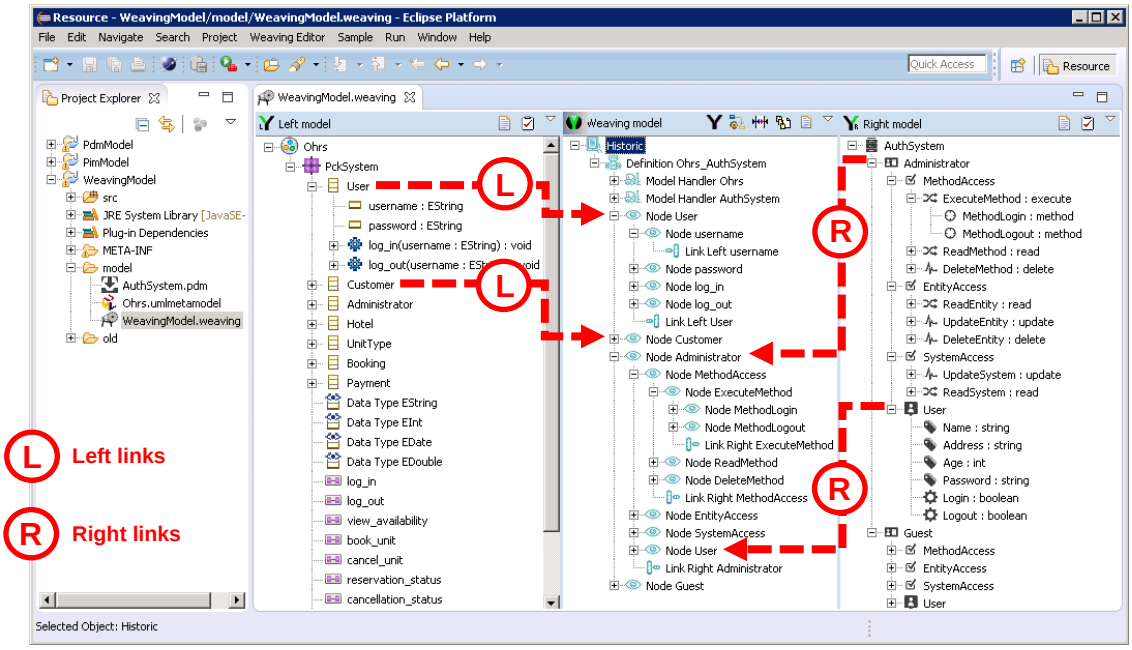

Fig. 10. Use of SWT4MDE for creating weaving model between PIM and PDM.

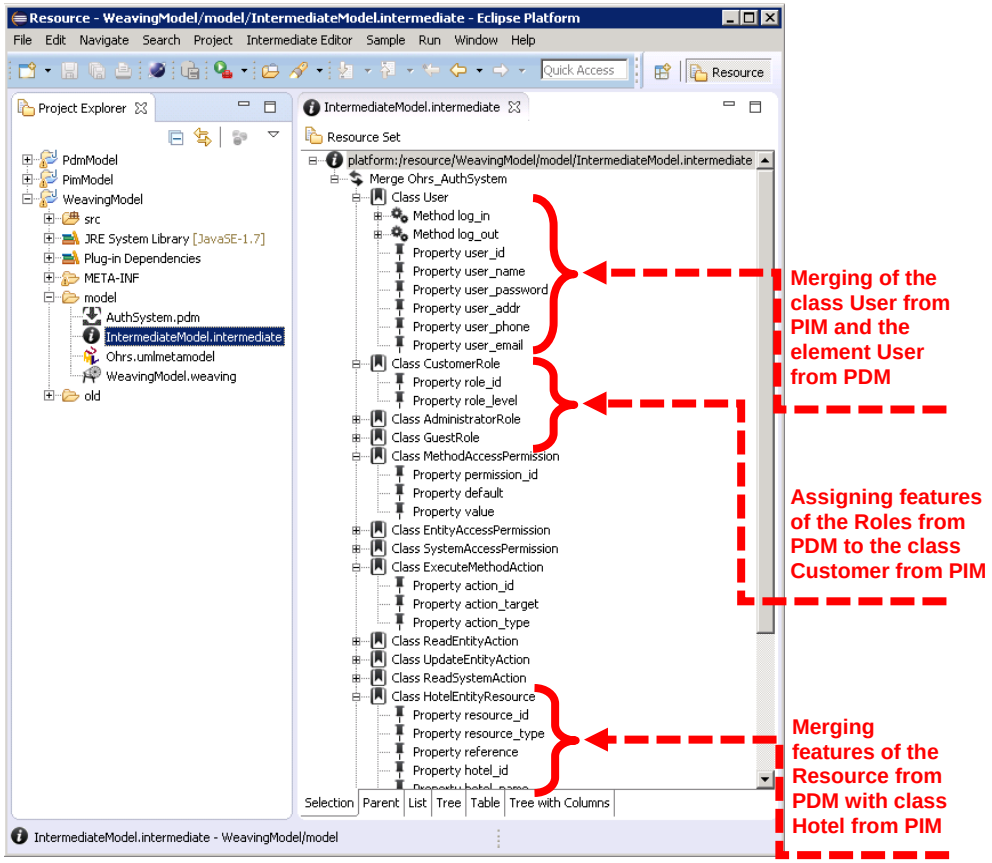

Fig. 11. Intermediate model of the illustrative example.

```
1  -- @path INTERMEDIATE=/Intermediate2WSDL/metamodels/Intermediate.ecore
2  -- @path WSDL=/Intermediate2WSDL/metamodels/WebServices.ecore
3  module intermediate2wsdl;
4  create OUT : WSDL from IN : INTERMEDIATE;
5  rule Merge2Definition {
6       from merge : INTERMEDIATE!Merge
7       to definition : WSDL!Definition (
8            name <- 'Service_' + merge.name,
9            targetNameSpace <- 'urn://' + merge.name + '.wsdl'
10           )
11 }
12 rule Class2Service {
13      from class : INTERMEDIATE!Class
14      to service : WSDL!Service (
15           name <- class.name,
16           ownerServ <- class.merge,
17           port <- port
18           ),
19           port : WSDL!Port (
20                name <- class.name + 'Port',
21                binding <- binding
22           ),
23           binding : WSDL!Binding (
24                name <- class.name + 'Binding',
25                ownerBind <- class.merge,
26                type <- portType
27           ),
28           portType : WSDL!PortType (
29                name <- class.name,
30                ownerPTyp <- class.merge,
31                binding <- binding
32           )
33 }
```

Listing 1. Transformation Definition written in ATL to generate an Abstract PSM from an Intermediate Model

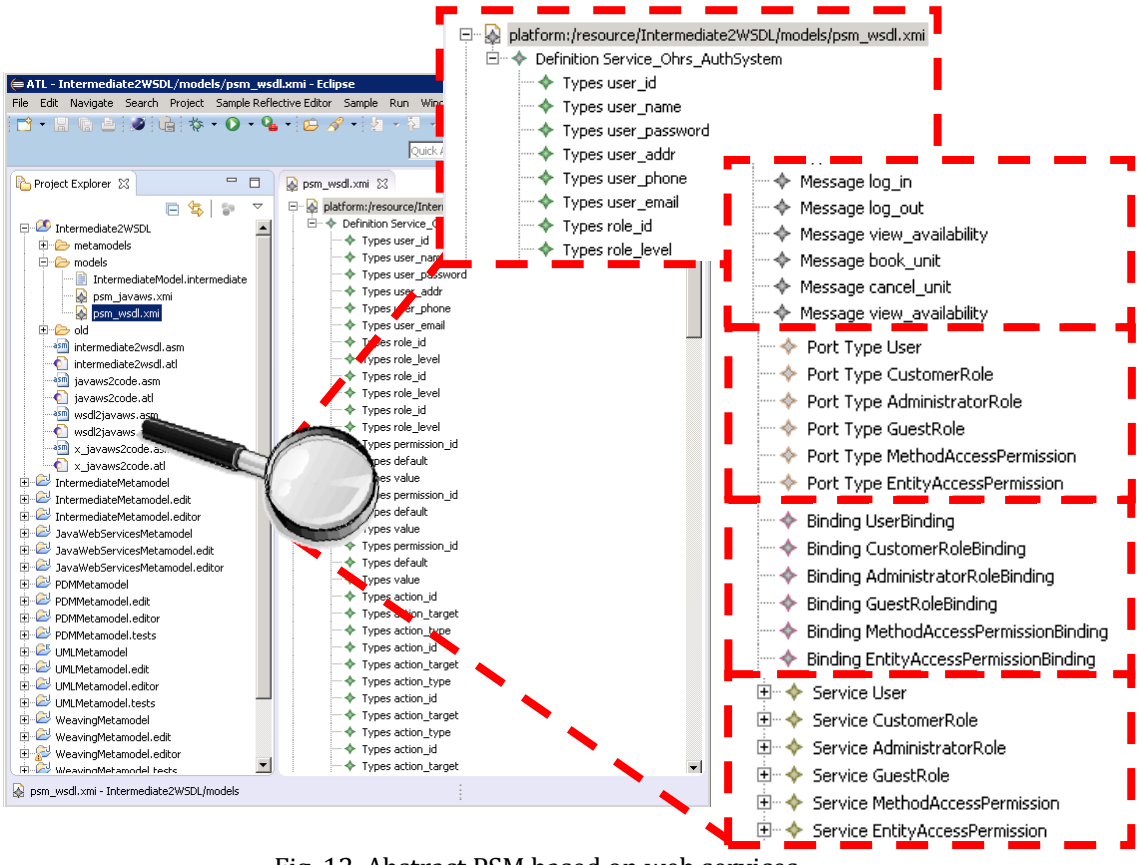

Fig. 12. Abstract PSM based on web services.

Fig. 12 presents the abstract PSM based on Web Services generated by the transformation definition described in Listing 1. The abstract PSM presents the Definition for Web Services Ohrs_AuthSystem. This Definition contains elements representing Types, Messages, Port Types, Bindings and Services. The elements are directly related to *Intermediate Model*, in which Properties, Methods, Classes, and some *Class* features are transformed into *Types, Messages, Services, Port Types and Bindings* respectively.

The framework presented in Fig. 1 and the methodology presented in Figure 2 are applied until generating the final source code. Listing 2 presents a source code of one generated file "*AdministratorOperationResource.java*".

The class *AdministratorOperationResource* has attributes and methods resulting from the weaving model. At this point, it is noted that the weaving tool has merged the two input models. The references of the element *Resource* from the security model is merged with the *Administrator* operations of the business model. This class is created taking a new structure in which specific operations abstracted from business model are defined as resources to a specific role (*Administrator*) from security model. The same happens with the other elements of business model that are abstracted as users, roles and resources, and have their actions controlled by the respective granted permissions based on user roles.

```
1  package Service_Ohrs_AuthSystem;
2
3  import javax.jws.WebMethod;
4  import javax.jws.WebParam;
5  import javax.jws.WebService;
6
7  @WebService()
8  public class AdministratorOperationResource {
9      @EJB
10     public /*type*/ resource_id;
11     @EJB
12     public /*type*/ resource_type;
13     @EJB
14     public /*type*/ reference;
15
16     @WebMethod
17     public /*returnType*/ view_availability() { //method body }
18
19     @WebMethod
20     public /*returnType*/ book_unit() { //method body }
21
22     @WebMethod
23     public /*returnType*/ cancel_unit() { //method body}
24 ***
25 }
```

Listing 2. Source code for the AdministratorOperationResource.java

5. Related Works

Ritu Sharma and Manu Sood promoted the MDA approach for developing cloud software applications in order to ensure that software solutions are more robust, flexible and agile, following constant evolving technologies [15], [16]. However, Ritu Sharma *et al* do not take care of aspects like security.

Edwark Willink proposed a graphical transformation language which involves small UML extensions to build a high-level language for model transformations [18]. Edwark Willink proposed the merging operation to bring together a PIM and additional information platform provided by a PDM in order to generate a PSM. He does not provide support to integrate aspects like security and his approach is not applied to cloud computing.

Abdessamad Belangour *et al* proposed a software development process called M2T (MDATM 2 Tracks) that provides an implementation of the MDA approach based on the Y development process [19]. M2T is also based on Y development process like our approach, but we also provide a framework and a methodology in order to include some aspects like security and we consider PDM as a model to recurrent models like design patterns.

Marcos Fabro *et al* proposed the weaving models to improve the database schema evolution in the context of MDE and provided Eclipse AMW plugin [20-21]. AMW plugin is a generic tool that was developed and applied to database systems. We also provide a solution based on weaving models, but we include a methodology and transformation definitions to take advantage of Y development process focusing in security aspects present in PDMs.

6. Conclusions

In this paper, we have presented an approach to develop a Secure SaaS based on Web Service. We have presented the PIM (i.e. business model), representing the SaaS application, and PDM model representing the access control application model. The weaving model relates the PIM and PDM and allows generating an Intermediate Model. We then take this Intermediate Model and transform it to an Abstract PSM conforms to Web Services platform. The abstract PSM is transformed into a concrete PSM conforms to Java and Java API for Web Service. Transformation definition takes as input the concrete PSM and generates source code based on Java language plus API for Web Services in Java.

We have presented a solution to take care of security concerns that are usually managed as an isolated feature of the software design process. In our proposed approach, we take care of security from the beginning thanks to models and model transformation definitions.

The proposed approach brings some benefits, because it improves the software quality, enhancing security, robustness and flexibility. In this approach, we introduced access control policies based on user roles for a SaaS application.

Our proposed approach has the following limitations:

- The approach does not cover encryption and data serialization issues. These aspects can be included in other PDMs, weaving models and model transformation definitions.
- The merging process between the business model and the security model cannot occur completely in an automated form. It is always necessary for a user intervention in order to identify the related aspects between models (i.e. PIM and PDM) and analyze if the correspondences between model elements are valid.

Our proposed approach was presented and an illustrative example was proposed to help understand how the things are going in the development process according to our methodology.

Acknowledgment

The work described in this paper was supported by CAPES, CNPq (Grant 560231/2010-5) and FAPEMA (Grant UNIVERSAL-00568/14).

References

[1] Baun, C., Kunze, M., Nimis, J., & Tai, S. (2011). *Cloud Computing Web-Based Dynamic IT Services*. Berlin, Heidelberg, Bertia: Springer.

[2] Ko, R., & Choo, R. (2015). *The Cloud Security Ecosystem: Technical, Legal, Business and Management Issues*. Waltham: Syngress.

[3] Vemulapati, J., Mehrotra, N., & Dangwal, N. (2011). SaaS security testing: Guidelines and evaluation

framework. *Proceedings of the 11th Annual International Software Testing Conference*.

[4] Favre, J. M. (2004), Foundations of model (driven) (reverse) engineering: Models – episode I, stories of the fidus papyrus and of the solarus. *Post-Proceedings of Dagsthul Seminar on Model Driven Reverse Engineering*. Dagstuhl, Germany.

[5] Bézivin, J., Hammoudi, S., Lopes, D., & Jouault, F. (2005). B2B applications, BPEL4WS, web services and. *Knowledge Sharing in the Integrated Enterprise*, *183*, 225-236. New York: Springer.

[6] Delessy, N. A., & Fernandez, E. B. (2008). A pattern-driven security process for SOA applications. *Proceedings of the Third International Conference on Availability, Reliability and Security* (pp. 416-421).

[7] Object Management Group. (2015). *Unified Modeling Language (OMG UML) (Version 2.5)*. OMG Document Number: Formal.

[8] Lopes, D., Hammoudi, S., Bézivin, J., & Jouault, F. (2005). Generating transformation definition from mapping specification: Application to web service platform. *Advanced Information Systems Engineering - Lecture Notes in Computer Science*, *3520*, 183-192.

[9] Basin, D., Doser, J., & Lodderstedt, T. (2006). Model driven security: From UML models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology*, *15(1)*, 39-91.

[10] Alalfi, M. H., Cordy, J. R., & Dean, T. R. (2012). Automated verification of role-based access control security models recovered from dynamic web applications. *Proceedings of the 14th IEEE International Symposium on Web Systems Evolution (WSE)* (pp. 1-10).

[11] Bézivin, J., Hammoudi, S., Lopes, D., & Jouault, F. (2004). Applying MDA approach for web service platform. *Proceedings of the Eighth IEEE International Enterprise Distributed Object Computing Conference (EDOC)* (pp. 58-70).

[12] Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF — Eclipse Modeling Framework*. Second Edition. Addison-Wesley Professional.

[13] Lopes, D., Hammoudi, S., & Souza, J., & Bontempo, A. (2006). Metamodel matching: Experiments and comparison. *Proceedings of International Conference on Software Engineering Advances*.

[14] de Sousa, J., Lopes, D., Claro, D., & Abdelouahab, Z. (2009). A step forward in semi-automatic metamodel matching: Algorithms and tool. *Enterprise Information Systems*, 137-148.

[15] Sharma, R., & Sood, M. (2011). Enhancing cloud SaaS development with model driven architecture. *International Journal on Cloud Computing: Services and Architecture*, *1(3)*, 89-102.

[16] Sharma, R., & Sood, M. (2011). Cloud SaaS: Models and transformation. *Communications in Computer and Information Science*, 305-314.

[17] Jouault, F., & Kurtev, I. (2005). Transforming models with ATL. *Proceedings of the 2005 international Conference on Satellite Events at the MoDELS* (pp. 128-138). Berlin: Springer Verlag.

[18] Willink, E. D. (2003). UMLX — A graphical transformation language for MDA. *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of Model Driven Architecture*. Anaheim.

[19] Belangour, A., Bézivin, J., & Fredj, M. (2006), Towards a new software development process for MDA. *Proceedings of the European Workshop on Milestones, Models and Mappings for Model-Driven Architecture* (pp. 1-15). Spain.

[20] Del, F. M. D., Bézivin, J., Jouault, F., & Valduriez, P. (2005). Applying generic model management to data mapping. *Proceedings of Base de Données Avancées*. Saint-Malo.

[21] Del, F. M. D., Bézivin, J., & Valduriez, P. (2006). Weaving models with the eclipse AMW plugin. *Eclipse Modeling Symposium*. Eclipse Summit Europe.

Pablo Matos was born in São Luís, MA, Brazil. He received his B.Sc. in information systems from CEUMA, Brazil in 2009. He received his M.Sc. in computer science from Federal University of Maranhão (UFMA), Brazil in 2015. His research interests are model driven engineering, cloud computing and information security.

Denivaldo Lopes is Ph.D. in computer science by University of Nantes, France. He is professor at Federal University of Maranhão (UFMA) and his lectures are about system engineering and development, model driven engineering and computer architecture. His research subjects are model driven engineering, embedded systems, cloud computing, security and software testing.

Zair Abdelouahab is Ph.D. in computer science by University of Leeds, United Kingdom. He is professor at Federal University of Maranhão (UFMA) and his lectures are about concurrent programming and distributed systems. His research subjects are model driven engineering, distributed systems, cloud computing and information security