A MODEL DRIVEN APPROACH SUPPORTING MULTI-VIEW SERVICES MODELING AND VARIABILITY MANAGEMENT

Boutaina Chakir and Mounia Fredj
ENSIAS, Mohammed V Souissi University, Rabat, Morocco

Keywords: MDD, MDA, SOA, SOC, Services modeling, Separation of concerns, Variability.

Abstract: Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve their business goals. Moreover, the growing of information systems increases the need of agility which implies the ability of a system to be adaptable to the changes in requirements and context of use. Managing variabily is considered as new leading edge concept for improving interoperability and reuse. Indeed, variability refers to the characteristic of a system to adapt, specialize and configure itself with the context of use. Several proposals have been proposed in this sense, but they are still immature and incomplete. Consequently, in this paper we propose a model driven method for managing variability in SOA based on MDA (Model Driven Architecture). In fact, through MD, the method enables the automation of service's realization regardless of supporting platforms. Our representation of variability is based on the extension of SOAML which is the future standard for modeling services. In addition, we adopt the separation of concerns theory by integrating modeling views, to better organize the various modeling artifacts.

1 INTRODUCTION

Service Oriented Architecture (SOA) is considered as another concept of maximizing agility and interoperability between different organizations. It is based on the description of services and their interactions.

Moreover, while technology and standards are important to support SOA, they are not sufficient on their own. Indeed, with the growing of SOA and the multiplicity of implementation platforms, the emphasis is more and more on the services modeling techniques. In fact, service's modeling aims to define the formalisms and notations needed to describe SOA solutions regardless of the technologies and standards. Numerous preliminary methodologies for service's modeling have been proposed. However, they are still immature and many issues haven't been deeply addressed in these approaches. In fact, with the continuing evolution of information systems, the demanding customer requirements increase the cost involved in designing and generating variants of a service in an ad-hoc manner. Hence, the need of mechanisms that support changes and encourage reuse in SOA systems is unavoidable. Techniques supporting reuse and adaptability rely on identifying and managing service's variability. Indeed, the aim of service's variability is to provide one central technique for better supporting the reusability of services in different application scenarios and to simplify the service consumption by consumers (Narendra, Ponnalagu, Srivastava. And Banavar, 2008).

In this paper, we present our model driven service modeling method. This method allows the management of variability, by dividing modeling process into two sub process. The first one is for commonalties and the second is for the variability aspects of a system. The proposed modeling method is based on the model driven architecture (MDA), witch enables the automation and the formalization of the method. To model variability, we propose to extend the SOAML profile (SOAML, 2009) by variability's stereotypes. Finally, in order to facilitate and organize our modeling artifacts, we divide the PIM layer of our method into multiple modeling views for better separation of concerns.

The remainder of the paper is organized as follows: Section 2 provides an overview of related works. In Section 3, we present the modeling approach and the different views integrated into our method. A case study is presented in Section 4 to illustrate our approach. The conclusion is reported in Section 5.

Chakir B. and Fredj M..
A MODEL DRIVEN APPROACH SUPPORTING MULTI-VIEW SERVICES MODELING AND VARIABILITY MANAGEMENT .
DOI: 10.5220/0003495502470252
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 247-252
ISBN: 978-989-8425-55-3

2 RELATED WORKS

In Service Oriented Computing (SOC) (Papazoglou, 2007), services modeling and designing approaches are an active area of research that aims to specify systems in a high level of abstraction regardless of implementation technologies. Among the studied approaches, we distinguish between those that use existing development processes (like XP and RUP) and those that propose their own processes. In the first category, we include the SOUP method (Mittal, 2006), which proposes two slightly different variations: one adopting RUP for initial SOA projects and the other adopting a mix of RUP and XP for the maintenance of existing SOA projects. In the second category, we mention the following methods: Thomas Erl's (ERL, 2005), IBM Service-Oriented Analysis and Design (SOAD) (Zimmermann, 2004), IBM Service Oriented Modeling and Architecture (SOMA) (Arsanjani, 2004), which act at provider side and support the services analysis and design.

To provide agility and reuse in response to the requirements and context changes in SOA solution, the variability management in SOA is an important issue. Some works are beginning to address this aspect under two perspectives: i) the variability for reuse, where we try to make explicit the variability in design artifacts, to reuse the service at provider level, ii) variability for adaptation, which focuses on the resolution mechanisms of variability according to the context of use. In this paper we were interested by the representation of variability in the purpose of improvement of services reuse. Among the works reviewed, we cite Robak and Franczyk (2003), which introduce the concept of modeling the variability of Web services using feature diagrams. Also, the work of (Segura, Benavides, Ruiz-Cort´es and Trinidad, 2008) focuses on the classification of variation points for Web Service Flows as a starting point for handling variability through services. Besides, Chang and Kim (2007) give a general classification of variation point in SOA solution: Workflow, Composition, Interface and Logic according to SOAD architecture's layers. Narendra and Ponnalagu (2007) propose an approach called Variation-Oriented Requirements Analysis based on a traceability model. All these approaches are focused on the representation of variability at just

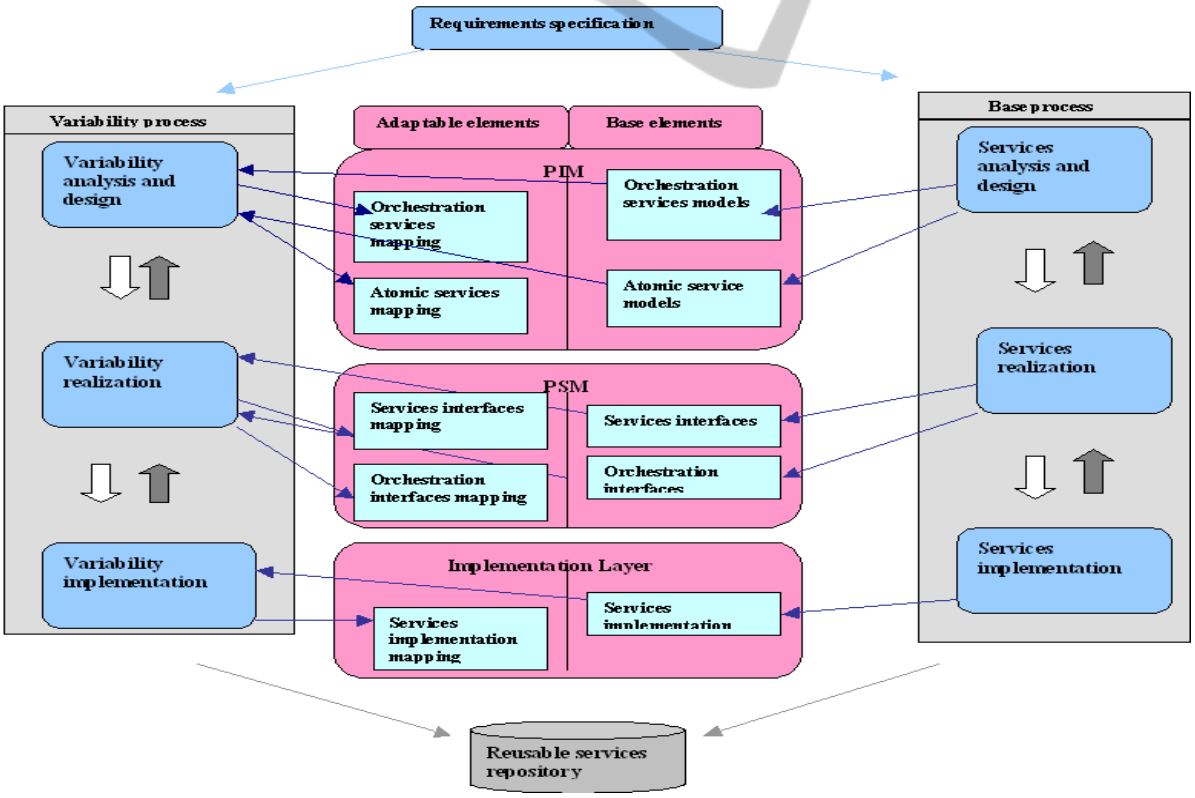

Figure 1: The proposed modeling method.

one level: requirement or analysis and not at all stages of services development lifecycle. To address this issue, Narendra et al. (2008) propose an end-to-end approach for variability modeling called (VOE), which consists of three steps: Variation-Oriented Analysis (VOA), which is concerned with analyzing the solution with respect to its static and changing parts; Variation-Oriented Design (VOD), where a variation model for the solution design is instantiated based on the results of VOA; and Variation-Oriented Implementation (VOI), which is responsible for producing an implementation based on the results of VOD. But this approach is orthogonal to service design, and also it is still not detailed and immature.

By reviewing these works, we found the absence of a comprehensive approach for modeling services, supporting the variability in different stages of lifecycle development process. Also, with the increasing number of platforms supporting SOA, it is interesting to use MDA in the development approaches. This justifies our approach, that consists of proposing a services modeling method, managing variability and following the MDA approach.

3 TOWARDS A MULTI VIEW MODEL DRIVEN METHOD FOR SERVICES MODELING

3.1 Overview of the Method

The main steps of the method are the following (see figure 1):

- Requirements specification: it consists of the identification of requirements sources, the elicitation and analysis of requirements. Also, it allows the identification of business process. Finally, it allows the development of the features model that contains common features and variable ones. The fixed elements trigger the base process of the method and the variable elements are the subject of the variability process;
- Base process: the main activities are: services analysis and design, services realization and services implementation;
- Variability process: the main activities of the process are: variability analysis and design, variability realization and variability implementation.

3.2 Extension of SOAML for Variability Modelling

For services modeling, we adopt the SOAML language, because it is the first standard language proposed by the OMG (OMG, 2011)for modeling SOA solutions. However, in order to model variability, it is important to provide mechanisms that permit the representation of variation point and variant which are important for the variability identification (Jacobson, Griss and Jonsson, 1997). Hence, we propose to enhance the SOAML profile by stereotypes that represent variability in SOAML models. For this purpose we use the UML package merge relationship to create our profile called VarSOAML.

So, our profile is a package merging SOAML (SOAML, 2009) with extensions defined in the variability package (see Figure 2).

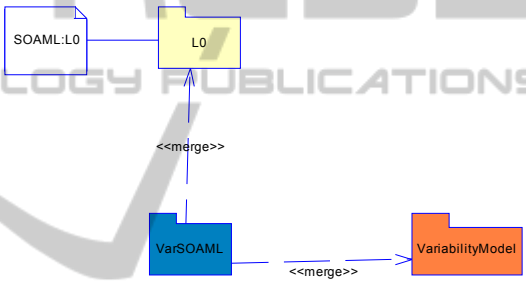

Figure 2: Variability profile.

The VariabilityModel package is composed of the following stereotypes (see Figure 3):

- variableOperation: expresses a variant for the variation point representing a service operation;
- VariableMessage: used at the SOAML messageType level;
- ComplexVariableType: used at the DataType level, that contains attributes;
- SimpleVariableType: applied at the DataType level that does not contains attributes.

The application of these stereotypes is explained in (Chakir and Fredj, 2011).

Moreover, nowadays, design approaches increasingly use the software engineering theory known as the separation of concerns. This theory asserts that a major problem is best solved when decomposed into a series of small problems or concerns. Thus, to facilitate the modeling of services, we adopt the separation of concerns in our method. In this perspective, we divide our PIM

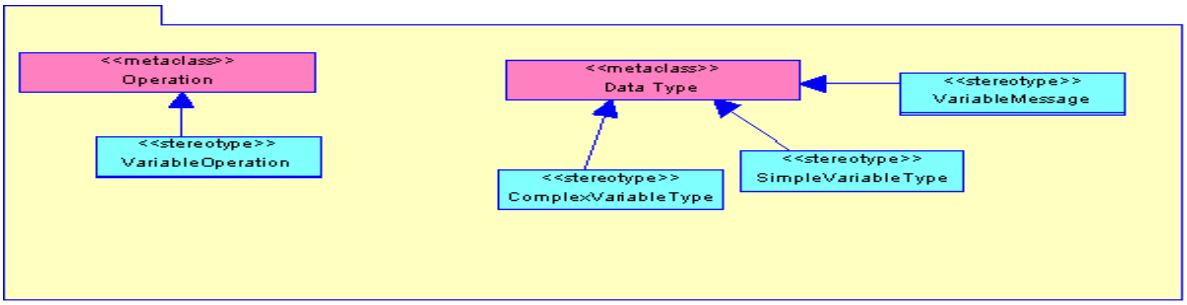

Figure 3: Variability stereotypes.

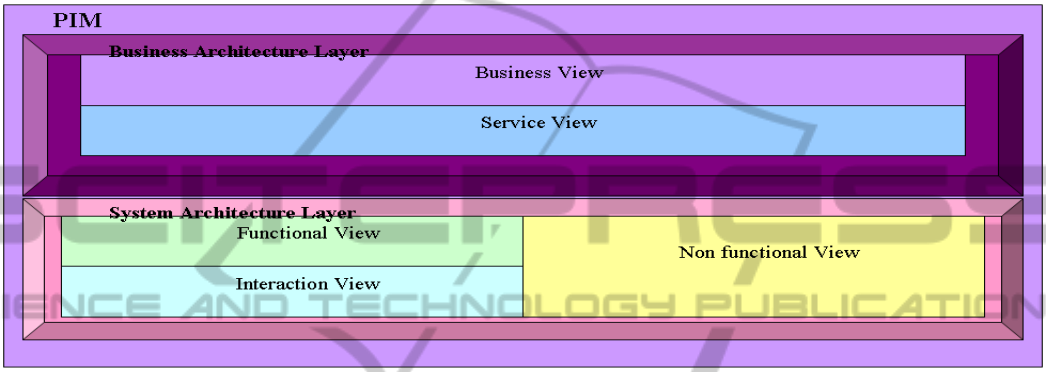

Figure 4: Multi views modeling integration.

(Platform Independent Model) layer into several modeling views.

3.3 Integration of Multi View Modeling in Our Approach

In order to adopt the separation of the concerns in our method, we divide our model PIM layer into two sub layers, which are in turn divided into several modeling views (see Figure 4):

- Business architecture layer: it captures the business requirements as well as it allows the identification of services from business processes. The models produced in this layer can be divided into two modeling views:
  - Business view: contains the various business processes models of the system, and the entities diagrams that capture the semantics of the solution;
  - Service View: allows the identification of services. It contains the architecture, contract diagrams and capability diagrams.
- System architecture layer: develops business architecture layer models. It is composed of the following views:
  - Functional view: contains services interface diagrams and messages exchanged between services;
  - Interaction view: allows the representation of services compositions. among the diagrams used in this view, we mention the service choreography diagram that models the interactions between services;
  - No functional view: represents the non-functional properties applied to services.

4 CASE STUDY

To validate our approach, we use the application proposed by (WS-I, 2007), describing "Supply Chain Management System" (SCMS). SCMS denotes the process of planning, implementing, and controlling the operations of the supply chain with the purpose to satisfy efficiently the customer requirements. In the requirements specification stage, we identify the following participants: Client, Retailer, Warehouse, Shipper and Manufacturer. To illustrate the functional variability of the SCSM services, we consider that in the case of a foreigner client, some retailers may offer the possibility of checking whether the customer is eligible or not. By

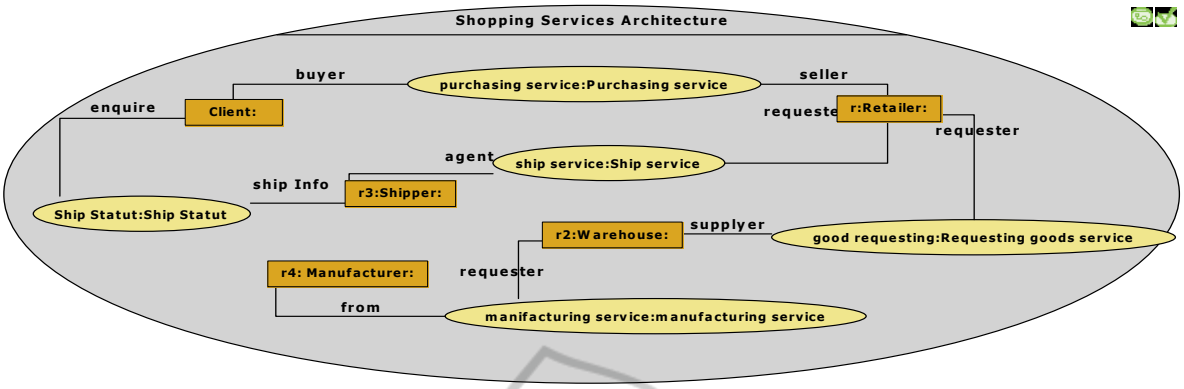

Figure 5: Service view-architecture diagram of SCMS.

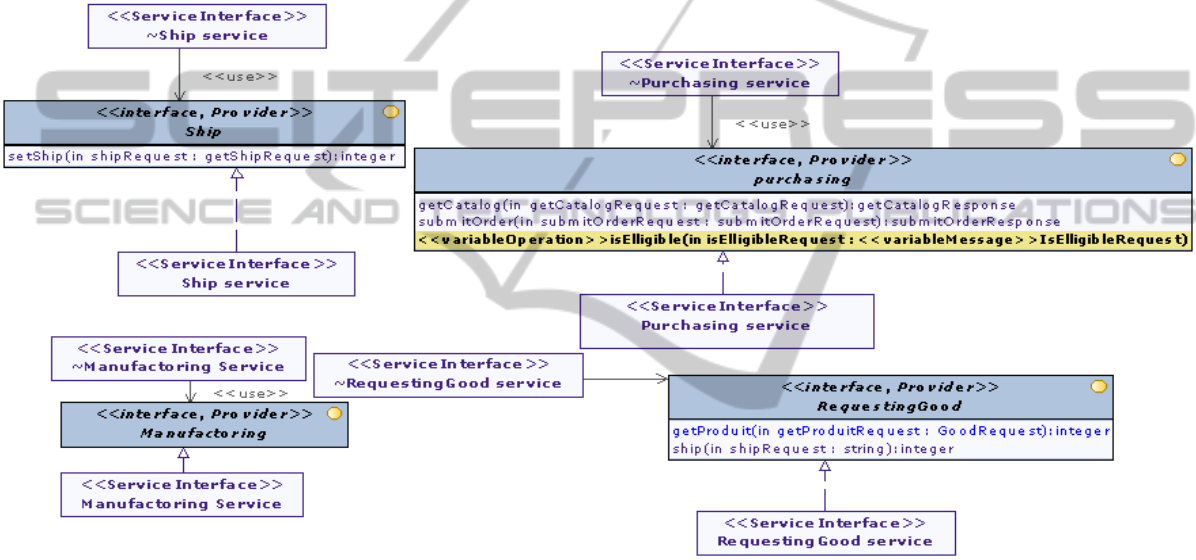

Figure 6: Functional view-interface diagram of SCMS.

applying our method, the analysis of the SCMS system produce a set of models related to each modeling view:

- Business view: in this view, we represent all the business processes of the system by the UML activity diagram. These processes are used in the identification of services;
- Service view: the Figure 5 illustrates a simplified architecture diagram of our system. The SCMS system architecture binds the roles of participants: Client, Retailer, Shipper, Warehouse and Manufacturer. Also, the participants participate to the following services: “purchasing service”, “ship service”, “ship status”, “requesting goods service” and “manufacturing service”. To specify each service we use the contract diagrams. For example the service “Purchasing service” links two roles “buyer” who is the consumer and “seller” who is the supplier;
- Functional view: in this view, we construct the interface diagrams and messages diagrams of the system, but because of the pages limitation, we just present a sample of the interface diagram in Figure 6. The SOAML stereotype «ServiceInterface» defines the interface and the responsibilities of a participant to produce and consume a service. The interface with a name beginning with ~ indicate the conjugate interface (consumer level), using the operations offered by the interface of the service producer;
- Interaction view: for a better explanation of the interactions between the various

participants of our system, it is interesting to enrich the model with the interaction diagrams which is represented by the service choreography diagram.

5 CONCLUSIONS

Nowadays, services modeling approaches become very important. In order to improve services modeling, it is interesting to use MDA, which brings automation ability, increasing reuse and productivity and reducing cost. Among the important issues that should be considered in modeling approach, we mention the variability management, which permits service reuse and service adaptability. In this regard, we have presented in this paper a model driven services modeling method that allows the variability management. The design artifacts of the method are organized into multiple modeling views for better separation of concerns.

Since our work is still ongoing, future work would involve detailing non functional view, presenting method's iterations, detailing the process of variability resolution and defining transformation rules in order to allow automatic generation of services. We will then be using this to provide tool support for our approach.

REFERENCES

Arsanjani, A. (2004). Service-oriented modeling and architecture. *IBM Corporation*, available in http://www-128.ibm.com/developerworks/webservices /library/ws-soa-design1/.

Chakir, B. and Fredj M. (2011). Towards a modelling method for managing variability in SOA. *Proceedings of the IADIS International Conference Information Systems 2011*, Avila, Spain.

Chang, S. H. and Kim, S. D. (2007). A variability modeling method for adaptable services in service-oriented computing. *Software Product Line Conference*, 261-268.

Jacobson, I. Griss, M. Jonsson, P. (1997). Software reuse: architecture process and organization for business Success. *ACM Press*, New York.

Erl, T. (2005), Service-Oriented Architecture: Concepts, Technology, and Design, *Prentice Hall PTR*.

Mittal, K. (2006). *Service Oriented Unified Process (SOUP)*, available from http://www.Kunalmittal.com/ html/soup.shtml, 2006.

Narendra, N. Ponnalagu, K. (2007). Variation-Oriented Requirements Analysis (VORA). *IEEE Congress on Services*, 159-166.

Narendra, N. Ponnalagu, K. Srivastava., B. and Banavar, G. S. (2008). Variation-oriented engineering (VOE): enhancing reusability of SOA based solutions. *SCC'08*, Vol. 1. IEEE, 257–264.

OMG. (2011). *Object Management Group*. Available in: http://www.omg.org/.

Papazoglou, M. Traverso, P. Dustdar, S. Leymann, F. (2007). Service-Oriented Computing: State of the Art and Research Challenge. *IEEE Computer Society*, Vol. 40, No. 11, 38-45, ISSN 0018-9162.

Robak, S. Franczyk, B. (2003). Modeling Web services variability with feature diagrams. *In: Revised Papers from the NODe 2002 Web and Database-RelatedWorkshops on Web, Web-Services, and Database Systems*, Springer,Verlag, pp. 120-128.

Segura, S. Benavides, D. Ruiz-Cort´es, A. and Trinidad, P. (2008). A taxonomy of variability in Web service flows. *In first workshop on Service-Oriented Architecture and Software Product Lines*.

SOAML, (2009). Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS). *FTF Beta 2*, Available in: http://www.omg.org/docs/ad/08-08-04.pdf.

WS-I, (2007). *Web Services Interoperability Organization (WS-I) Web site*, Available in: http://www.ws-i.org/.

Zhang, L. J. Arsanjani, A. Allam, A. Lu, D. Chee, Y. M. (2007). Variation oriented analysis for SOA solution design. *Proceedings of the 2007 Salt Lake City*, 560-568.

Zimmermann, O. Krogdahl, P. and Gee, C. (2004). Elements of Service-Oriented Analysis and Design. *IBM developerWorks*.