HAL
open science

A service-oriented approach for model management

Jorge-Luis Pérez-Medina, Dominique Rieu, Sophie Dupuy-Chessa

To cite this version:

Jorge-Luis Pérez-Medina, Dominique Rieu, Sophie Dupuy-Chessa. A service-oriented approach for model management. 12th Iberoamerican Conference on Software Engineering (CIbSE'2009), 2009, Medellin, Colombia. pp.115-120. hal-00953679

HAL Id: hal-00953679

https://inria.hal.science/hal-00953679v1

Submitted on 28 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Service-Oriented Approach for Model Management

Jorge Luis Pérez Medina, Dominique Rieu, Sophie Dupuy-Chessa

LIG Laboratory, Grenoble University
385 rue de la Bibliothèque, B.P. 53 3841 Grenoble Cedex 9, France
{Firstname.Lastname}@imag.fr

Abstract. In the Software Engineering (SE) domain, the Model Driven Engineering (MDE)[1] paradigm focuses on using models as main software artifacts to provide a full description of software systems and on automating model manipulation with tools. Model management concerns a set of features allowing representing, creating, storing and manipulating models. Nowadays, the needs of models designers in terms of management process and products are diverse. Modeling tools are not complete because there is no consensus about models needs and uses. To remedy the heterogeneity and the functional limitations of models management tools, we propose a service-oriented approach for model management for the creation of modeling environments adapted to the needs of designers. The considered needs are related in two abstract levels: the operational and the organizational level.

Keywords: Model, service, model management, MDE, modeling tools, operational service, organizational service.

1 Introduction

Software development is a collaborative process where teams of developers work together to design solutions and produce quality code [1]. In order to cover the variety of software viewpoints (business, human-computer interaction, security…), multiple modeling languages are used by model designers in a software project. In particular, UML covers many concerns for modeling software systems; however, there exists domain-specific concerns, such as the Human Computer Interaction (HCI) domain [2], that model designers cannot express using UML. Due to the needs to describe the system through different viewpoints, software development projects involve actors with different types of skills that play different roles (e.g. designers, ergonomists,…). Project managers need to create environments that support a useful coordination of activities and tasks between actors, and actors need to create models to resolve assigned activities. To assist them in their work, the model-based approaches aim at helping designers in understanding user needs and in designing solutions in an effective way. Consequently, realizing a model-based software project requires an

[1] Official Site of Model Driven Engineering, http://www.actionidm.org/.

environment that could assist designers to collaboratively manage models and to coordinate all Software Engineering (SE) activities involving differents models.

Nowadays Computer-Aided Software Engineering (CASE) tools and Model-Driven Engineering (MDE) tools are used to describe a set of concepts and technologies around model management. Tools can manipulate different models and perform different actions on models. They are limited in features and often reduced in the use of some meta-models and models. They do not permit the coordinated and cooperative models management required by different specialists, that work together in a context of collaborative design [3]. In conclusion, models management tools are not complete because there is no consensus on the needs and the uses of models.

At the same time, using models permits a new form of interoperability in terms of model-based tool integration. It focuses on the integration through metamodeling tools [4]. Several works have treated the problem of model-based tool integration [5]. However, none of them answer the questions of the choice of modeling tools according to the needs of actors and design processes. The question that we try to resolve is how to build a model management environment that solve technical needs of companies in conformity with theirs organizational needs.

We want to propose a modeling environment that allows model designers to build a modeling environment that suits her needs, whatever the role she plays in a design process. To find a solution to this problem, we propose to adapt the services based approach [6]. Our approach considers the models management as services for creating modeling environments adapted to the needs of different stakeholders, whose goals are: a) Documenting services to facilitate their research and their use; b) Integrating model management tools as services in an open platform to add, edit and delete services; c) Defining language modeling development process to describe the roles and needs of each actor that participates in a software development process in terms of models management.

In this paper, we propose two types of services: "operational" services and "organizational" services. This paper is organized as follows. Section 2 presents the needs of model designers and the features offered by models management tools. Section 3 presents the concepts of service in an overview of our approach based on two modeling levels. Models of the operational layer and the organizational layer are presented. Finally a conclusion and some perspectives are presented in section 4.

2 Features offered by models management tools

Model designers need to manipulate models along the software project lifecycle. At the organizational level, the system design consists in guiding modeling activities using human resources. Consequently, project managers need to create environments that support a useful collaboration of activities and tasks among actors. This means that it is necessary to define the roles that are played by model designers and the activities realized. It is also necessary to define all the modeling languages used by the stakeholders of a project. Considering these concerns, the Business Process Management (BPM) deals with the design, the automation, and the optimization of business processes [7]. In our work, we focused on identifying those functional

activities that produce or use models. Our aim is to offer modeling environments based on the reuse of modeling activities.
A development process consists in collaborative activities (e.g. cooperation or coordination) among designers. In a cooperation, designers must work together to produce a common product: specialists filling different headings in a common form are for instance realizing a cooperation activity. Coordination occurs when tasks are decomposed into common activities. A common planning is set up, actors are dispatched on these activities and a proper goal is assigned to each actor or activity. Drawing semantic relationships between models is a typical coordination activity example.The issue is to take into account any existing development method, ith its organizational specificities. Finally, an efficient modeling management cannot be realized without automated support. Consequently model designers need some software engineering activities to be automated.

In order to automate model management activities, many models management tools are available. With regard to our approach, the features offered by model management tools can be considered at two modeling levels. At the operational level, the model management functionalities offered by tools are model management, model transformation, etc. At the organizational level, the functionalities offered by tools consist in guiding modeling activities using human resources.

3 Two levels of service

Service-Oriented Computing (SOC) uses services as the constructs to support the development of rapid, low-cost and easy composition of distributed applications. A service is a set of self-contained software modules and auto-descriptive applications than can be described, published, discovered, composed, and negotiated on demand by a customer [8]. Services perform functions that can be anything from simple request to sophisticated business processes requiring multiple layers of services consumers and providers [8]. Services are offered by service providers. In our context, a service provider is the entity that offers any editor which realizes treatments on models and tools for models management. The clients are users with needs in model management. They represent any Model Designers. They must able to find the description(s) of the modeling services adapted to their needs.

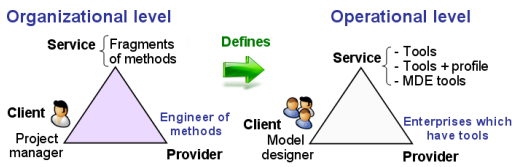

Fig. 1. Two levels of service.

Our approach based on services does not limit itself in considering each model management tool as a "service". It relies on two modeling levels (see Figure 1) where providers, clients and services are different. The first level corresponds to the

operational layer. This layer offers services for model designers, to facilitate the creation of their modeling environment. The client is a model designer who wants to manage models in an individual or collaborative way (with other designers). So, she needs to define her modeling environment adapted to her favorite functions and use context. The organizational layer permits to model the design process of a system in terms of modeling activity. This layer offers a support based on services for project group. Therefore, in this layer, roles and activities are expressed in terms of simplified development processes. The organizational layer reuses, in a coordinated way, the operational services defined in the first level. The customers is, in this case, project managers who need to define and manage roles and activities for their process.

3.1 The operational layer

The operational layer describes the static structure supporting the research and the coupling of services that are required by model management. An operational service (Fig. 2) corresponds to an executable application composed of other model management services. A model management service consists in a tool offered by a provider. An operational service is associated with various types of features: a) different types of collaboration; b) some factors of software quality; and c) the contexts of utilization of model designer.

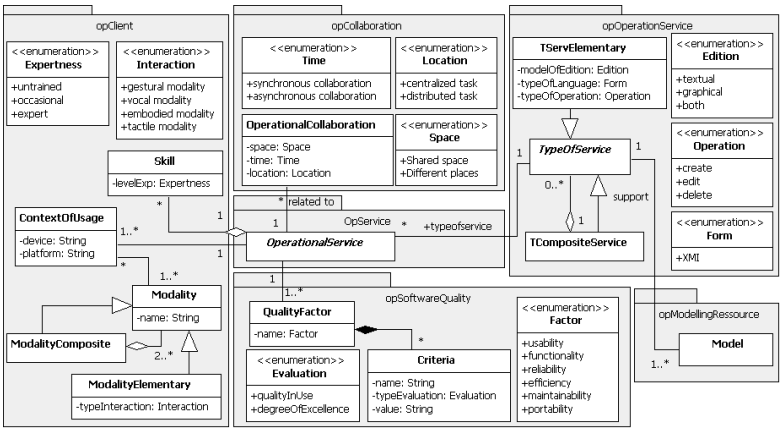

Fig. 2. Operational model service.

An operational service is associated to various types of services which are represented by a composition of elementary services. An elementary service defines how a model designer can make operations on models, the supported languages to store the information and the possible operations of service. To take into account the reuse of modeling, a type of service supports the use of one or several models. Modeling operational services is not new. We extend the classical aspects (e.g. collaboration, context of usage) in response to the needs identified by [3]. This study

lead us to propose an original operational model service characterized by: 1) aspects related to collaborative design; 2) and the user's context.

To support the needs of creating collaborative environments, an operational service is connected to several collaboration forms. A collaboration form is defined according to three axes: location, space and time. Location represents either if the task is centralized or distributed. Space is the area where tasks are performed. Finally, time is the collaboration moment (synchronous or asynchronous).

To guarantee the quality in the use of a model management tool, we introduced the notion of quality factor. It consists in the composition of several criterias that assure the software quality. The evaluation can be expressed as a function or a degree of excellence. Related to the notions of users profile and context of use of an operational service, Fig. 2 considers that an operational service can be executed in different contexts of use by specialists who have diverse levels of expertness. The context of use is a structured information space that includes the hardware-software platform required by the service. It is related to a variety of interaction modalities, multimodality referring to a composition of multiple modalities simultaneously.

3.2 The organizational layer

Our organizational model service is inspired by several works. Ralyté et al. [9] propose a method engineering process model approach, which permits to represent any method as an assembly of the reusable method fragments. In our work, we use the notion of service to support the construction of modeling process environment by assembling method fragments of design processes adapted to the functional activities of model designers. Fig. 3 presents the organizational model service.

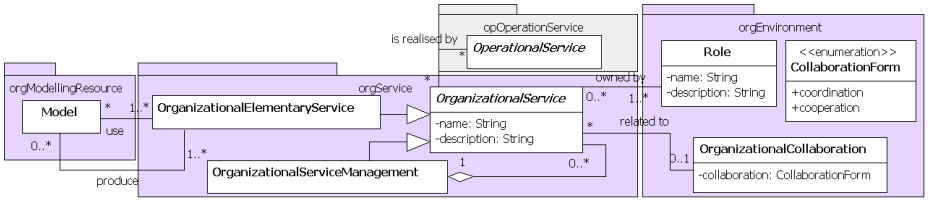

Fig. 3. Organizational model service.

An organizational service consists in a composition of development method fragments of that can be reused by model designers. A method fragment is represented by an organizational elementary service that is defined in terms of model manipulation. An organizational service is carried out by one or more roles. A model designer who plays a role can define and reuse several organizational services. At this level, the "collaboration" term is used for coordination and cooperation tasks between designers. Another aspect considered by our organizational model is the fact that the operational services are the mechanisms that realize the organizational services. It means that organizational elementary services must use operational service to support the management of modeling activities.

4 Conclusion and perspectives

This article presents the principles of our service-oriented models management approach. It relies on two modeling levels where providers, clients and services are different. The first level corresponds to the operational layer. An operational service considers aspects such as: the type of services, modeling resources, collaboration forms, context of use and user profiles. At the second level, the organizational model permits the reuse of operational services in a coordinated way, but also the creation and the management of model process fragments. This organizational model will allow project managers to model design processes.
Afterward, we consider necessary to specify an intentional layer. This layer will consist in modeling the goals that can be proposed to represent the intention of a specialist, a group of specialists, a unity of work or any organization involved in the development process. Therefore, this intentional level will permit to justify the existence of fragments of design processes, actors and roles. Then, we must develop a modeling process language in order to unify the concepts of intentional and organizational levels (expressed in terms of models management) with operational services. This modeling language must take into account coordinated and cooperative model management required by several specialists. Finally, we must consider the use of a platform for the implementation of our three levels of services. The existing service-oriented tool such as DoCoSOC [10] will be important references in this area.

Acknowledgments. We are grateful to the Foundation "Gran Mariscal de Ayacucho" and the university "UCLA-Venezuela" for their financial support.

References

1. Booch G., Brown A., Collaborative Development Environments, in Advances in computers Vol. 59, Academic Press, August 2003.
2. Paterno, F., Mancini, M., ConcurTask Tree: a diagrammatic notation for specifying taskmodels, Proc. of 6th Int. Conference Interact'97. (1997). pp. 362-369.
3. Godet-Bar G., Rieu D., Dupuy-Chessa S., Juras D., Interactional Objects : HCI concerns in the analysis phase of the Symphony method, Proc. of the 9th Int. Conference ICEIS'07, Madeira (2007).
4. Kapsammer E., Reiter T., Schwinger W., Model-Based Tool Integration - State of the Art and Future Perspectives, Proc. of 3rd Int. Conference CITSA'06, Orlando, USA, (2006).
5. Wicks M., Tool Integration within Software Engineering Environments: An Annotated Bibliography, Technical Report. Ref. HW-MACS-TR-0041, 4th August 2006.
6. Bieber, G., Carpenter, J. Introduction to Service-Oriented Programming (Rev 2.1), 2001.
7. Wil van der H., Hofstede A., Weske M., Business Process Management: A survey, Proc. of the Int. Conference BPM'03, Eindhoven, Netherlands, June 2003, pp. 26-27.
8. Papozoglou M-P., Traverso P., Dustdar S., Leymann F., Service-Oriented Computing: A Research Roadmap, Dahstuhl Seminar 05462, 2005, pp. 1 – 29.
9. Ralyté J., Rolland C., An Approach for Method Reengineering. Proc. of the 20th Int. Conference ER'01, Yokohama, Japan, November 2001.
10. Yu J., Lalanda P., Chollet S. Development Tool for Service-Oriented Applications in Smart Homes, IEEE SCC (2) 2008: pp. 239-246.