Defining Service-Oriented Software Architecture Models for a MDA-based Development Process at the PIM level

Marcos López-Sanz, César J. Acuña, Carlos E. Cuesta, Esperanza Marcos
Kybele Research Group
Rey Juan Carlos University
C/Tulipan S/N – CP 28933
Móstoles, Madrid (Spain)
{marcos.lopez, cesar.acuna, carlos.cuesta, esperanza.marcos}@urjc.es

Abstract

Although the development of service-based software has evolved greatly in the last years, there is still a lack of accurate development methodologies applied to service orientation. To solve this, we focus on the Model-Driven Development approach, and more specifically in MDA, the proposal by the OMG, as it is one of the main trends in nowadays Software Engineering. Accordingly, we propose to define a Service-Oriented Software Architecture (SOSA) model within a model-driven methodological framework called MIDAS, for the adaptable and flexible development of modern systems. In the present work we study the concepts and elements that need to be defined at a platform independent and neutral abstraction level. This paper reflects our proposal of PIM-level architecture model for Service-Oriented Software Architectures within MIDAS.

Keywords. Service-Oriented Software Architectures, Model-Driven Architecture, PIM-level modelling

1. Introduction

Although much research has been made in defining with precision service orientation within the Architecture field, there are still some gaps to be filled. Model-Driven Development (MDD) and, in particular, the proposal by the OMG, MDA (Model-Driven Architecture) [13] is one of the current leading trends in Software Engineering which can help to fill some of these gaps.

We aim at the definition of service-oriented software architectures using a model-driven approach. Our work is framed in a methodological framework called MIDAS [5], which follows an Architecture-Centric Model-Driven Development (ACMDD) approach [11].

In this paper we centre our attention in the definition of the architectural model at the PIM level of MDA that will be incorporated to the MIDAS methodological framework. To correctly define this model, an analysis of the main concepts involved in the SOC paradigm is made, taking into account that no aspect related to the underlying platform or implementation technology should be included in this model. To achieve this, we first study other works related to service-orientation, model-driven development of services, service architectures and UML modelling. We assemble in our proposal the common SOSA principles identified together with the aspects needed to fill the gaps found in the bibliographical study.

2. Related Works

Many approaches define which elements should be identified in a service-oriented architecture. Proposals can be broadly divided into two groups: those taking their principles from the underlying technology (Web Services mainly, such as [1], [2] or [10]); and, on the other hand, those which adopt a more conceptual and technology-neutral point of view.

Among the latter, and taking into account the notation used to define the architectural model and properties, we see that most of them rely on UML and create metamodels and profiles to be used in the description of the architecture ([7], [8], [15], [17]).

The extensive use of UML as notation demonstrates how MDD (Model-Driven Development) is considered as one of the leading trends in current service-oriented software development. Because of that, many initiatives follow this approach. For example, in [16] the usage of models to integrate process-driven service-oriented architecture models is shown. In [7] a proposal with the separation of PIM and PSM levels proposed by MDA is presented. Moreover, the inclusion of SOSA within a software development process has been also introduced in other works [9].

It is considering the elements defined for the SOSA model when most of the differences come up. All the
proposals support the definition of services, of course, and even identify roles or types of services but without a global agreement. [4] consider service clients, providers and service discovering agents; [7] define a UML profile which only allows to define two distinct kinds of services (provider and registry) and does not include any facility to model semantic or constraint issues; [17] enact service contracts, the roles given to the services in the contracts and the identification of pre- and post- conditions, etc. However, they don’t deal with the problem of user interaction as they do not define any modelling component to support it. There are other proposals whose set of concepts is more complete and adequate for the definition of a general SOSA model ([3] or [14]).

In summary, there are not many proposals which gather together all the concepts involved in SOSA: the enactment of service composition, the description of service boundaries with the environment and interactions, the definition of semantic aspects and architectural constraints within the model, a level of abstraction sufficient to be independent of the underlying technologies and a solution in which the SOSA model play a guiding role throughout the entire software development process. In the next sections we present a SOSA modelling proposal which meets all these features.

3. PIM-level SOSA Model

3.1 MDA Framework

The definition of the service architecture model is part of the MIDAS methodology, a framework for the development of information systems based on the MDA principles in which the architecture guides the whole development process.

The MIDAS model architecture, as shown in figure 1, can be seen from 3 orthogonal dimensions:

- **Vertical Axis (Y):** reflects the three abstraction levels of the original MDA proposal.
- **Horizontal Axis (X):** Models are separated in the different concerns involved in the information system development.

- **Traversal Axis (Z):** Models in this axis are referred to aspects that have influence on other models of the cross-cutting axis.

The architecture is considered to be the driving aspect of the development process as it allows specifying which models, elements inside models or relations within models should be created during the entire software development process.

With an architectural view of the system at PIM level, we ensure that there are no technology or implementation constraints in the model. In addition, it facilitates the establishment of different PSM-level models according to the specific target platform and derived from one unique PIM model.

3.2 Service Concepts and Metamodel

![Figure 2. PIM-level SOSA Metamodel.](image)

- **Service providers**

SOSA is widely used as a way to integrate enterprise applications and so the existence of these providers is justified by the necessity to project the business organizations into the architecture model. In the presented architecture model these entities are identified as UML 2.0 packages. Service providers can be classified into two main groups: inner service providers (innerProvider in the metamodel), which are internal to the system designed, and outer service providers (outerProvider), which are identified as external entities containing services which perform a specific task of
value for the system but which are not under its control.

The relationship between two service providers is modelled as a dependency understood as a "business contract" (businessContract).

3.2.2. Active components

In a SOSA solution, the main elements are services; however, as user responses are needed in many cases (for example when a process composition takes place and user is required to select among several execution paths), a first-class element which retrieve the user interaction must be represented in the architecture model, namely SystemFront-End. By definition, services lack of the ability to interact or, in particular, to modify its behaviour according to user interaction. SystemFront-End components will be identified as system boundaries.

The main role of services inside SOSA is to support the functionalities offered by the system and derived from the requirements stated by the user and the information gathered in the upper CIM models. Our vision of the service concept (at PIM level) is aligned with that of the OASIS reference model for services – A service is a mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description [12]–.

Services can be classified according to its atomicity in simple services (SimpleService in the metamodel) or composite services (CompositeService). Composite services have a property (named policy) which can be set to two possible values: Orchestration or Choreography. In case of taking the Orchestration value, it is required that at least one of the compound services assume the coordinator role (OrchestratorService).

Simple services, in turn, can be divided according to the functionality role given into basic services (BasicService) or supporting services (SupportingService). Basic services represent those services with distinctive functional meaning. Supporting or specialized services perform actions clearly identified and not necessarily with direct relation to the modelled system functionality.

These supporting services can be again classified in two: orchestration services (OrchestratorService) or core services (CoreService). Core services perform tasks needed for other services to work correctly. Among these we find, for example, location, registry or security management services.

Despite the type of service modelled, all of them are defined by a set of operations (serviceOperations) and data stored (serviceData). We identify service operations as atomic functionalities that collaborate to outline the service features. Data is modelled as an element inside the architecture model because, from the point of view of our development methodology, these data represent a link point between the parts of the methodology in charge of modelling the data aspect (Content aspect in MIDAS) and the architecture model guiding the development process.

3.2.3. Service Interactions and Messages

Services relate, communicate and interact with each other through contracts. The contracts in a service oriented architecture model act as connectors between the different architecture components. These contracts will have different features depending on the components connected. So, we refer to ServiceContract when talking about service-to-service connections (asymmetric interactions based on request/reply message exchanges) and InteractionContract when talking about the communication between a SystemFront-End and a service.

The conditions under which the contract takes place refer, usually, to non-functional requirements such as quality of service, security, interaction protocol or message order, type of communication (synchronous or asynchronous), etc.

It is possible that for two services to communicate they fulfil some pre- or post- conditions. In our architecture model this clauses are represented as restrictions associated to the serviceContract element and using OCL as description language.

The communication between the application front-end and the services may be as simple as a parameterized invocation of the service from the front-end component, or may be more complex, for example, based on a negotiation protocol. In the metamodel this concept is represented by the InteractionContract element.

Although the semantics associated to the message appear in the architecture model, the message format does not. The format of a message is an issue that depends mostly on the implementation technology and so it should be modelled in the correspondent PSM-level models.

4. Conclusions and Future Works

Service orientation is gaining importance in the software development field mainly because of the broadening of fields in which it is being used and accepted as guiding paradigm. However, several problems, such as an increase of the complexity development process, are derived from this evolution. This motivates the necessity to design complete and accurate development methods to assist the development process of service-oriented systems.

The model-driven approach in general and the MDA proposal in particular have turned out to be a great advance in software development. The separation in
abstraction levels (CIM, PIM and PSM) and the definition of transformation rules between the models defined within these levels and among levels are the foundations of the MDA proposal. The application of the MDA principles to the SOC paradigm favours the development of solutions based on services. On the other hand, the design of the architecture is a critical aspect in SOSA developments since one of its main goals is to achieve flexible implementations of interacting software entities.

In this article we have explained all the concepts and elements that should be identified in the definition of Service-Oriented Software Architectures at the PIM level of MDA. The results of the research work presented have been included in the MIDAS framework, an ACMDD methodology for the development of information systems.

Regarding other proposals for the service-oriented system development, we have seen that many of them consider Web Service principles as basis, keeping them apart from generic designs. The ones which are enough generic to avoid fixing to a specific technology usually can not be applied to MDA-based developments or do not use a high-level standard notation to help the development process. Among the proposals which define UML profiles for SOSA, practically none of them have a complete development method associated and only focus on the ways of representing the SOSA principles and not in the entire development process. Finally, almost none of them allow the representation, as first-class architectural element, of a component in charge of interacting with the user.

We are currently studying the influence of the designed architectural model in some other parts of the MIDAS model architecture as well as the transformation rules from one model to another.

5. Acknowledgements
This research is partially granted by project GOLD (TIN2005-00010) financed by the Ministry of Science and Technology of Spain, the FoMDAs project (URJC-CM-2006-CET-0387) co-financed by the Rey Juan Carlos university and the Regional Government of Madrid, and the Spanish project "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010).

6. References