COMPONENT-ORIENTED MODEL-BASED WEB SERVICE ORCHESTRATION

Suela Berisha, Jacques Hamalian and Béatrice Rumpler
Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205 F-6962, Villeurbanne, France

Keywords: Service Oriented Architecture (SOA), Web Services Orchestration (WSO), Service Modelling, UML Modelling, Model-driven Architecture (MDA), Service Component Architecture (SCA).

Abstract: The Web service orchestration enables cross departmental coordination of business processes in a heterogeneous environment. We have designed a component-oriented web service orchestration based on service functionality in an Enterprise Service Bus (ESB). Our orchestration model is centralized-based, where a single service is the central coordinator containing the composition logic of the other services involved. These services are modelled using Service Component Architecture (SCA) specification, defined in our UML profile described in this paper. SCA conforms to SOA (Service Oriented Architecture) principles using CBSE (Component-Based Software Engineering) techniques. Finally, our specified orchestration model is implemented using Model-Driven Architecture Approach (MDA).

1 INTRODUCTION CONTEXT

The Information System (IS) of the SNCF (http://sncf.com/en_EN/flash/#/CH0001/BR0959/) is based on a complex data exchange system with, heterogeneous, distinct and very large data sources. So, it becomes very difficult to get the "good" information or data according to a user need, rapidly. The introduction of Service-Oriented Architecture (SOA) (Papazoglou & al. 2007), (Pautasso & al. 2008), whose main component is the Web Service (WS), promises a better organization on the global system and enhances more the visibility of the business process, according to a standard-based service orchestration. However, this type of architecture presents new problems because services constitute a set of consequent "moving parts" not knowing each other. Moreover, in the case of dysfunction, the task of identifying the problem becomes very difficult, and searching the error costs a lot. Therefore, it is necessary to implement a general politic for a system in order to manage and supervise services and their life-cycle, integrated in a global Web Service Orchestration (WSO). Currently, there exist some implementation works regarding this type of infrastructure. However, at the moment, these works don't enough convince managers to be proposed at the company level.

Our work places this problem area in an empirical case study about the "human resource domain". First, we have proposed the architectural model of a platform based on ESB infrastructure (Chappel 2004), (Schmidt & al. 2005). This model contains a meta-model layer and a model layer based on a service orchestration engine, a service interface and three types of repositories: process repository, event repository and service repository (Hamalian 2009). This paper presents the following steps: (i) the proposition of a conceptual orchestration model and (ii) the procedure for an implementation and (iii) the results obtained from the conceptual model.

2 FUNDAMENTAL CONCEPTS AND MODELLING APPROCHES

WSO implies three fundamental concepts: (i) the orchestration model (Joffroy & al. 2007), (Ortiz & al. DC 2006), (ii) the business processes and (iii) the implementation languages (Panagiotis 2008).

An *orchestration model* is an interaction model that takes place during the construction of composed web services. This orchestration is based on some requirements (Ivanova 2006) and criteria (Dustdar & al. 2005), (Erradi & al. 2005) concerning web service composition, like: modularity (Chappel 2004), capacity of communication and coordination (Dustdar 2005), incremental approach to service

Berisha S., Hamalian J. and Rumpler B. (2010).
COMPONENT-ORIENTED MODEL-BASED WEB SERVICE ORCHESTRATION.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 479-482
DOI: 10.5220/0002908204790482

composition and modelling (Gronmo & al. 2004). We have taken account of these criteria during our model designing.

Business processes are integration programs that tie together various entities within (and between) organizations, to achieve business goals. They are ideally suited to be implemented in a Service Oriented Architecture (SOA), where functionality is provided by services with well-defined interfaces (Papazoglou & al. 2007). The service orchestration, make profit of the POM (Process-Oriented Modelling) and aims to describe the sequence of activities involved in a business process. These sequences are designed graphically thanks to modelling tools (by business actors) or specified in a formal language (by programmers).

Currently, some formal *orchestration languages* (Ivanova 2006) are based on programming languages, for example: JSR207 and jPD. But they are limited in terms of interoperability and reusability. Thus, the most available are derivation of XML which try to model business processes interpreting WSDL Files (Web Service's Interface). Indeed, each service represents a business specific function and a set of services form a business process. For example BPEL (or BPEL4WS) design composite service applications that need some process logic. It is a system-to-system process well adapted for well known process structures. We have chosen BPEL4WS for his interoperability and reusability capacities, but it can't deal with processes only formed at run-time, without modelling patterns. It requires from the designer to be aware of the sequence logic between the different activities involved prior to designing the orchestration model. So, we are interested in modelling approaches that might occur patterns.

The first, the SCA (Ortiz & al. 2006) is a CBSE (Bernd 2008) specification providing the composition and the aggregation logic of services according to the system functionalities, i.e. business process logic. It provides methods and concepts to create components and to describe how they can work together. SCA has sharply gaining acceptance in the services-oriented industries because it offers some advantages like understandability, maintainability, reusability, composability, capability in reconfiguration. But SCA doesn't integrate the remaining development stages (Ortiz & al. DC 2006), neither the sequence activities logic, that can be defined separately, using UML sequence diagrams for each possible business activity. For the complete automatic development system from and to end, we have chosen the meta-model of MDA approach (Nasser 2005) (Ortiz & al. 2006), (Ivanova 2006), (Pham & al. 2007). This model leads us to distinguish a Platform-Independent Model (PIM) where we find the business logic and the functionalities, and a Platform-Specific Model (PSM) which represents a specific model for an implementation technology or an execution platform.

3 CONCEPTUAL MODEL

According to the architectural model (Hamalian 2009) and MDA recommendation we have designed a Conceptual model (Figure 1) showing coordination between layers, repositories and execution platform.

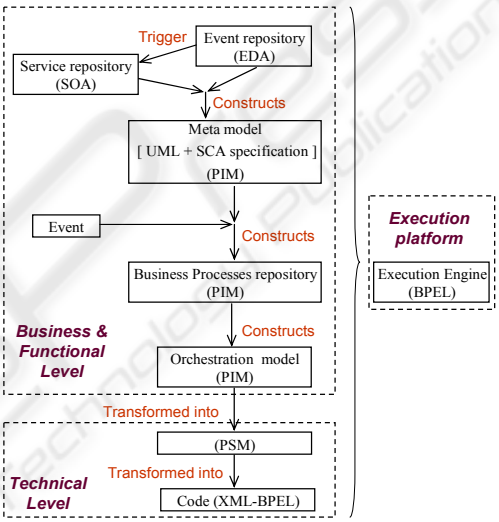

Figure 1: Our conceptual model according to MDA.

Services could be activated by events which are defined in the event repository according to Event-Driven Architecture (EDA) (Papazoglou & al. 2007) in an SOA. We have combined SCA with MDA in a PIM model: the business and functional layer. UML became the foundation modelling language in OMG's MDA and then, we represent SCA specification using UML 2 component diagram. The next step transforms the PIM model based on MOF into an XML document (PSM model) via XMI. Moreover, our PSM model contains the SCA specification defined in XML. Since we have a model conformed to MOF, we can use any MOF-based frameworks or tools. This permits us to ensure the interoperability between the systems by the help of an exchange file which contains the mapping rules between MOF-XMI. Next, the PSM model is transformed into BPEL code that describes our orchestration model which contains the SCA specification.

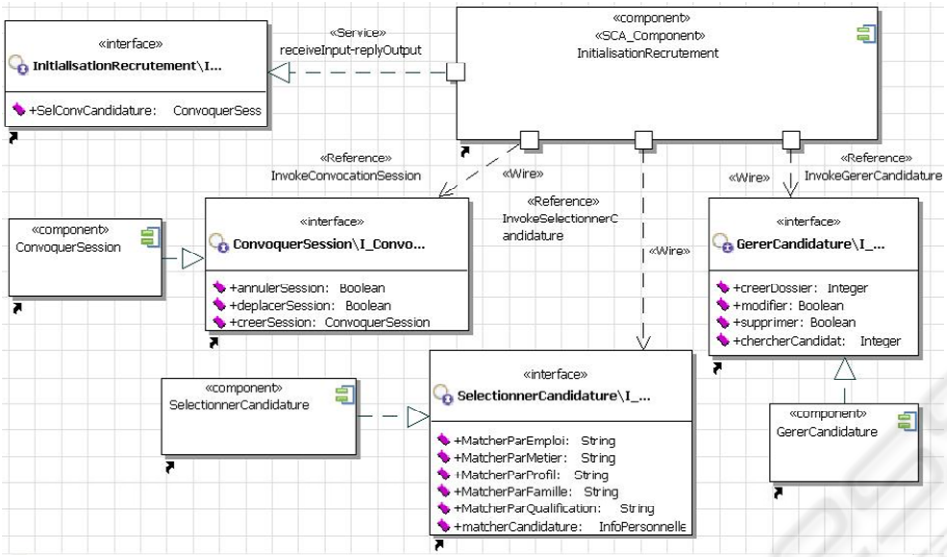

Figure 2: Our orchestration PIM model combined with the UML profile for SCA.

4 THE IMPLEMENTATION

We propose to build an orchestration model based on where the service implementation is encapsulated in the form of a black box and its functionalities are exposed via input/output interfaces using the SCA specification (Ortiz & al. 2006). (Masmoudi & al. 2006) had represented a correspondence between UML 2 component models and its equivalent in BPEL, in order to model aggregate software components using the UML. However, the difference, with respect to our work is that, he didn't envisaged the "mapping" between SCA and BPEL which permits us to obtain a better modularity and adaptability for our system. These mappings are represented in the Table 1 and the Table 2.

After having defined these correspondences, we have built our orchestration PIM model combined with the UML profile for SCA (Figure 3).

So, the service component is now stereotyped as <<SCA_Component>>. It supplies its interface "I_InitialisationRecrutement" through the port "P_IR" stereotyped as <<Service>> which could be used by other services. Moreover, it provides a reference for other services through the interface "I_Resultat" and through the port "P_IR_Resultat" stereotyped as <<Reference>>. Similarly, other service components are stereotyped as <<SCA_Component>> providing their services through interfaces by the help of their corresponding ports stereotyped as <<Service>>. The connection between two components is specified through a dependency connector which is stereotyped as

Table 1: "Mapping" proposition between SCA and BPEL.

SCA	BPEL4WS
Component	PartnerLink(<partnerLinks>)
Composite	PartnerLink [1]
System Assembly	Process [2] (<process/>)
Components Invocation	Sequence of activities (<sequence/>)
Reference	Reply
Service	Invoke, receive
Wire	Service components interaction logic
Property	Variable
Implementation	Web Service's Implementation corresponding to a partnerLink
[1] BPEL Process (exposed as a WSDL Service Type) [2] BPEL Process (corresponding to the final system)	

Table 2: Our "mapping" proposition between SCA and UML 2 component diagram.

SCA	UML 2 component diagram
Component	Component
Composite	Composite component
System Assembly	Artefact
Components Invocation	Sequence diagram between the involved components
Reference	Port (Required Service)
Service	Port (Supplied Service)
Wire	Dependency
Property	Property
Implementation	Service Implementation of a component

<<Wire>> following the SCA specification. For simplification, the property stereotype is not shown in Figure 2. In order to implement the new UML Profile for SCA, we have defined a sequence of operations using a UML sequence diagram. (Hamalian J 2009). The implementation of our

orchestration model is formalized by the way of a XML file containing the BPEL source code (Hamalian S 2009). This BPEL process is interfaced as a WSDL file that could be consumed as web services in other functional contexts.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a web service orchestration model following the SCA specification which takes into account elementary service aggregation and service design regarding the business perimeter. We consider that a specific actor can define an elementary service adapted to a functional task or an activity of its business. This service can be combined with other services (elementary or composite) to be adapted to an activity granularity or a business process. In order to implement this type of aggregation, we have built the UML service profile model which extends the UML actual metamodel for taking into account the characteristics of an SCA component.

At the moment, we work at the implementation of a system able to reuse the BPEL file source code in a larger scale of framework, compared to this prototype. In terms of further perspective we propose to work on the integration of the Aspect-Oriented Modelling (AOM) approach in the current models for taking into account technical transversal tasks shared by many business processes (identification, authentication, service class management, etc.).

REFERENCES

Belwood, T., Clement, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y. L., Januszewski, K., LEE, S., Mckee, B.,Munter, J., Von Riegen, C. Universal Description, Discovery and Integration. In *Technical Committee Specification 2004*. http://uddi.xml.org/.

Bernd, J. K., Component meets service: what does the mongrel look like? In *Innovations in Systems and Software Engineering 2008*. Vol. 4, pp. 385-394. Springer, London.

Chappel, D.: Enterprise Service Bus. O'Reilly Media, Inc., Sebastopol (2004).

Dustdar, S. Schreiner, W., 2005. A survey on web services composition. In *J.Web Grid Serv*. Vol. 1, PP. 1-30.

Erradi, A., Maheshwari, P.: AdaptiveBPEL. 2005. A policy-driven middleware for flexible web services composition. In: *Proc. of the EDOC MWS 2005 Workshop*, Netherlands.

Grønmo, R., Solheim, I. 2004 Towards Modeling Web Service Composition in UML. In *WSMAI 2004. PP.72-86*.

Hamalian, J, june 2009. Orchestration des Services d'un ESB fondée sur les modèles, Report Master's degree, INSA of Lyon, France

Hamalian, J., september 2009. Orchetration de services dans un bus de services d'entreprise, Report Engineer's degree, INSA of Lyon, France

Houspanossian, A., 2006. Enhancing a bpel4ws engine: Supporting the execution of flexible ws-flows according to the refflow model.BSc. Thesis, UNICEN, Argentina.

Ivanova, E., 2006. Technologies for Web Services Orchestration. In: *XLI International Scientific Conference on Information, Communication and Energy Systems and Technologies ICEST. pp. 240-243*. Sofia, Bulgaria.

Joffroy, C., Mosser, S., Blay-Fornarino, M., Nemo, C., 2007 Des Orchestrations de Services Web aux Aspects. In: *3ème Journée Francophone sur le Développement de Logiciels Par Aspects JFDLPA 2007*. France.

Masmoudi, A., Paquette, G., Champagne R., 2006. Implémentation à l'aide de BPEL de trois processus d'agrégation de composants, dirigée par les modèles. In *Proceedings of the Objects, components and models in information systems engineering Workshop (INFORSID 2006). pp. 231-246*. Hammamet, Tunisia.

Nassar, M., 2005. Analyse/conception par points de vue : le profil VUML. Theses INPT, Toulouse.

Ortiz, G. Hernández, J., 2006 Aspect-Oriented Techniques for Web Services Model Driven Approach. In: *Leymann, F.; Reisig, W.; Thatte, S. R. & van der Aalst, W. M. P. (ed.) IBFI 2006*. Internationales Begegnungs- und Forschungszentrum fuer Informatik, Schloss Dagstuhl, Germany.

Ortiz, G., Hernandez, J., 2006. Toward UML Profiles for Web Services and their Extra-Functional Properties. *In: In Proceedings of the IEEE international Conference on Web Services ICWS 2006 PP.889-892. IEEE Computer Society*, Washington, DC.

Panagiotis, L., 2008. Orchestrating Web Services with BPEL. In *IEEE Software, vol. 25, pp. 85-87*.

Papazoglou, M. P., Heuvel, W., 2007. Service oriented architectures: approaches, technologies and research issues. In *The VLDB Journal, vol. 16, pp. 389--415*.

Pautasso, C., Zimmermann, O., Leymann, F., 2008. Restful web services vs. "big"' web services: making the right architectural decision. In *Proceeding of the 17th international Conference on World Wide Web* WWW 2008. pp. 805--814. Beijing, China. ACM, New York, NY.

Pham, H. N., Mahmoud, Q. H., Ferworn, A., and Sadeghian, A., 2007. Applying Model-Driven Development to Pervasive System Engineering. In *SEPCASE 2007*.

Schmidt, M., Hutchison, B., Lambros, P., and Phippen, R., 2005 The enterprise service bus: making service-oriented architecture real. In *IBM Syst. J. 44, 781--797*.

Senthil, R., Kushwaha, D. S., and Misra, A. K., 2007. An improved component model for component based software engineering. In *SIGSOFT Software Engineering 2007, vol. 32, pp. 9.*