From a High Level Business Process Model to Service Model Artifacts
A Model-Driven Approach

Mokhtar Soltani[1] and Sidi Mohammed Benslimane[2]
[1]*Sciences and Technology Department, Ibn Khaldoun University, Tiaret, Algeria*
[2]*Computer Sciences and Technology Department, Djillali Liabes University, Sidi Bel Abbes, Algeria*

Keywords: Business Process Modeling, Model-Driven Development, Service-Oriented Architecture, Service Identification.

Abstract: One of the key activities needed to develop a quality service oriented solution is the specification of service model. The majority of existing methods for service model specification are developed manually because they are based on the competence of the developers. The integration of Business Process Modeling (BPM) and Model-Driven Development (MDD) allows the automation of the SOA (Service-Oriented Architecture) services development. Three steps are used for developing an SOA solution: service identification, service specification and finally service realization. In this paper we propose a method called AMSI (Automatic Model-Driven Service Identification) that automatically identifies the architecturally significant elements from a high level business process model to specifying service model artifacts. The main goal of this work is to support the automation of the development process of service-oriented enterprise information system.

1 INTRODUCTION

Bridging the gap between Enterprise Modeling methods and Semantic Web services is an important yet challenging task. For organizations with business goals, the automation of business processes as Web services is increasingly important, especially with many business transactions taking place within the Web today (Nadarajan et al., 2007). Nowadays, the enterprises are organized in networks, in which various actors can be interacting. The competitiveness of these companies is deeply related to the capacity to structure, share and exchange knowledge with the participants in the collaborative network. This need to exchange knowledge obliges the companies to evolve their information systems and their applications in order to return them interoperable. The interoperability of enterprise applications allows ensuring the exchange of the functionalities and the services in a transparent way. Each functionality, service, or data have a specific model. Several transformations of these models are essential to ensure interoperability between the various heterogeneous entities of the enterprise. So that these model transformations become an effective solution for establishing interoperability in a purely heterogeneous environment; it is necessary that they must be guided by a standard modeling framework. The MDA approach (Model-Driven Architecture) provides the bases to support the model-driven interoperability. The development of an enterprise application to large scale always starts with the highest level abstraction where they are the specification and the representation of the business in the form of business process models. These models must be projected gradually on an adapted architecture to the need for interoperability. Currently, the more adapted paradigm to the realization of the interoperable applications is the service-oriented paradigm. Since the services encapsulate the functionalities of the applications according to enterprise business processes, the comprehension of the business process model is a precondition necessary to the automatic derivation of the Service-Oriented Architecture (SOA). The main idea of the service-oriented approach is to quickly build applications by the assembling of a set of software services. The result of this assembly is called "composite application". The effective achievement of this approach allows facilitating

Soltani M. and Benslimane S..
From a High Level Business Process Model to Service Model Artifacts - A Model-Driven Approach.
DOI: 10.5220/0003989702650268
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 265-268
ISBN: 978-989-8565-12-9

integration and the interoperability of enterprise information systems which are heterogeneous and was not conceived to be interoperable. Therefore, the service-oriented approach proposes a framework to solve these problems by the development of interoperable applications starting from a high level business process.

The remainder of this paper is organised as follows. In section 2 we presented a basic concepts needed to understand our approach including Business Process Modelling, Service-Oriented Architecture, and Model-Driven Architecture. In section 3 we presented our approach called AMSI (Automatic Model-driven Service Identification). Finally, our conclusion and future work is given.

2 BUSINESS PROCESS MODELING AND SERVICE-ORIENTED ARCHITECTURE HANDSHAKE

Service identification is one of the core elements of the BPM and SOA handshake that reinforces the current mantra that “BPM should be service oriented, SOA should be business process focused, and SOA takes over where BPM leaves the enterprise in a path towards agility” (Inaganti et al., 2007).

2.1 Business Process Modeling

The process vision plays a significant role in the theories of the organizations as in the information system field where the process modeling is regarded as a key element of the representation of dynamics. The business process modeling is a prerequisite necessary to design an organizational information system. The business process definition reflects the functional needs implicitly.

However, it is not sufficient to just conceive the business activities connected by control flows of the process. To represent the complete whole of the requirements, a process definition must explicitly indicate all the entities which take part in the process. These requirements should be transformed, without loss of information, in semantic specifications of which different software components can be derived.

2.2 Service-Oriented Architecture (SOA)

Service-Oriented Computing (SOC) is a new paradigm for distributed computing that uses services to support the development of interoperable, evolvable, and distributed applications. Services are autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled by using standard protocols. Service-Oriented Architecture is the main architectural style for SOC. The main idea of Service-Oriented Architecture is the restructuring of enterprise information systems into loosely coupled, independent services. These services should allow the reuse of existing implemented functionality in order to minimize the time between design and implementation when business requirements change. The key challenges in developing the service oriented systems are the mapping of business processes models into service models. Service models play an important role during service-oriented analysis and design phases.

According to the IBM SOMA (Arsanjani, 2004), service-oriented modeling lifecycle has three main phases:

Service Identification. This phase is about identifying the architecturally significant elements of the target solution. The output artifact of this phase is analysis-level service model.

Service Specification. This phase is about describing a service: what it offers, what it requests and how it is exposed. It also describes dependencies with other services, service composition, and service messages. The main model related to this phase is the design-level service model.

Service Realization. This phase is about providing a solution for a particular service. We represent here, how a service is realized. The model related with this phase is the design model. This model has to be traced back to the service model, because it represents its realization.

3 AUTOMATIC MODEL-DRIVEN SERVICE IDENTIFICATION

Service identification is one of the main activities in the modeling of a service-oriented solution, and therefore errors made during identification can flow down through detailed design and implementation activities that may necessitate multiple iterations, especially in building composite applications.

According to (Arsanjani, 2004), the initial activity in the development of a new SOA solution is the service identification. The result of the service identification is a set of candidate services.

The first stage in the service identification process is the modeling of a high level business process that is represented as a CIM model. Metadata are added to the CIM model in the second stage. This operation is based on a whole of generic concepts stored in the process model ontology. The third stage allows the transformation, after the interrogation of the process model ontology, of the input business process model into an executable process model expressed as a PIM model. In the fourth stage, the service identification engine generates a whole of candidate's services for implementing the input process. (**cf. Figure 1**)

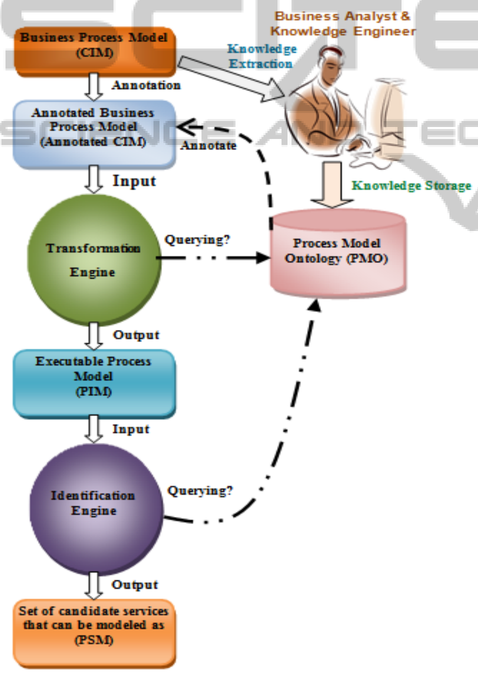

Figure 1: Automatic Model-Driven Service Identifier.

3.1 Business Process Annotation

In this stage, complementary information is added to the business model elements such as the nature of the activities (manual, semi-automatic, automatic), the composition of the activities (decomposable, atomic activity), the goal of activity etc. All knowledge's about the initial business process are extracted and stored in the Process Model Ontology by business analyst and knowledge engineer.

Our automatic service identifier uses a Process Model Ontology (PMO). This ontology allows the annotation of the input business process model and it is regarded as a knowledge base usable to identify the software services.

The Process Model Ontology is a formal framework that provides knowledge description and reasoning techniques. The starting point in defining an ontology is to decide what the basic ontology elements (concepts and roles) represent. The Process Model Ontology is based on two principles: to unify the different existing Business Process Metamodels, and to provide the necessary properties for deriving software services from a height level business processes. The PMO captures generic concepts associated with business processes and the relationships among them. To facilitate the extraction of the multiple views on a process model the PMO allows a business analyst and a knowledge engineer to mark the visibility of activities to different collaboration roles. Thus various views on the business process model can be extracted.

Our proposition is that constructing the Process Model Ontology through a careful analysis of existing reference metamodels, guarantees the representational width of the ontology, i.e. that all existing business process models can be represented and all software services can be extracted from it.

3.2 Business Process Transformation

The first step for identifying SOA services is the business process modelling. However, we cannot directly transform a high level business model into SOA solution because it is independent of any computation specification and it comprises manual, semi-automatic and automatic activities. As well as the high level activities have a great granularity. The same business activity can be transformed into several SOA services. Thus it is necessary firstly to transform the high level business process into an intermediate process called executable process in order to identify the candidate services. As business process models are at a higher abstraction level than executable process models, additional domain knowledge will need to be added during this step. The Process Model Ontology is queried to transform the annotated business process model into an executable process. During this stage, the transformation engine executes the following operations:

- rename a business activity,
- split a business activity into several,
- merge two business activities in only one,
- insert a new activity in the executable process,
- remove an activity from the business process.

3.3 Service Identification

The identification engine queries the ontology and takes the executable process as input in order to generate automatically the candidate services. In this phase, a set of design metrics which satisfy business goals should be extracted from the PMO such as cohesion, coupling, granularity of activities, etc. that are considered as input parameters for classifying the candidate services in a groups (composite services). The transformation between executable process model and Service-Oriented Architecture should be one-to-one. All aspects modeled in the executable process are transferred to a service model artifact. The identification of the services corresponding to the executables activities is possible via their names. The identification engine searches the name of the activity in the concepts taxonomy of the ontology and extracts the properties of the activity and its relations with the other concepts of ontology. After this research phase, the identification engine generates a set of candidate services equivalents (implementing) to the activity in question. Thus their initial descriptions (name of the service, names of the interfaces, etc.). We can formulate our identification algorithm as a multi-objective optimization problem that classifies candidate's services according to optimal values of design metrics. (cf. Figure 2)

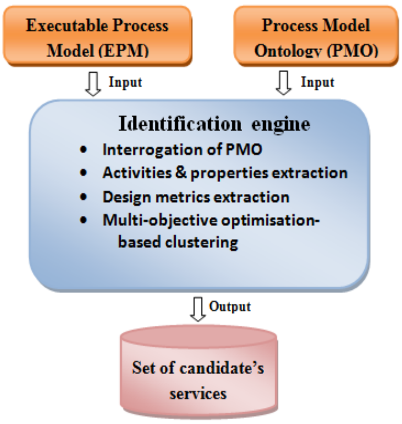

Figure 2: Service Identification Steps.

4 CONCLUSIONS AND OUTLOOK

In this paper we outlined some of our initial work in the development of AMSI, a method for identifying automatically candidate SOA services from a high level business process. The method defines how a high level business process should be transformed into an executable process in order to identifying SOA services. Our automatic service identifier uses a Process Model Ontology (PMO) to annotate the business process model. The annotated business process model is used as input of a transformation engine which transforms it, after the interrogation of ontology, into an executable process. Finally an identification engine querying the ontology and take the executable process as input in order to generate the candidates services automatically.

Future work is mainly related to the description of transformation and identification algorithms, finally implementation and evaluation of our approach.

REFERENCES

Arsanjani, 2004. (SOMA) Service-Oriented Modeling and Architecture: How to identify, Specify, and Realize services for your SOA. *IBM developer Works.*

Fareghzadeh, N., 2008. Service Identification Approach to SOA Development. In: *Proceedings of World Academy of Science, Engineering and Technology, vol. 35, pp. 258–266.*

Inaganti, S., and Behara, G. K., 2007. *Service Identification: BPM and SOA Handshake.*

Jamshidi, P., Sharifi, M., and Mansour, S., 2008. To Establish Enterprise Service Model from Enterprise Business Model. *IEEE International Conference on Services Computing*

Jamshidi P, Mansour S, Sedighiani K, Jamshidi S, Shams F., 2012. *An Automated Service Identification Method. Technical Report*, Department of Electrical and Computer Engineering, Shahid Beheshti University.

Klose, K., Knackstedt, R., Beverungen, D., 2007. Identification of Services - A Stakeholder based Approach to SOA Development and its Application in the Area of Production Planning. In: *ECIS 2007, pp. 1802–1814*

Leonardo Guerreiro Azevedo, Flavia Santoro, Fernanda Baiao, Jairo Souza, Kate Revoredo, Vinicios Pereira, and Isolda Herlain., 2008. *A Method for Service Identification from Business Process Models in a SOA Approach.*

Nadarajan, G., and Chen-Burger, Y.-H., 2007. Translating a Typical Business Process Modelling Language to a Web Services Ontology through Lightweight Mapping. IET Software *In: Formerly IEE Proceedings Software, Vol. 1, Issue 1, p.1-17.*

Papazoglou, M. P., Heouvel, W. J., 2006. Service-oriented design and development methodology. *Int. J. Web Engineering and Technology, Vol 2, No. 4, 2006.*

Yukyong Kim and Kyung-Goo Doh., 2008. *Formal Identification of Right-Grained Services for Service-Oriented Modeling.*