An Approach for Reducing the Gap between BPMN Models and Implementation Artifacts

Bruno Leonardo Barros Silva[1], Bruno Leonardo Barros Silva[11], Robson Wagner Albuquerque de Medeiros[1], Julio Cesar Damasceno[1], Fernando Antonio Aires Lins[1], Nelson Souto Rosa[1], Paulo Romero Martins Maciel[1]

Bryan Stephenson[2], Hamid Reza Motahari Nezhad[2], Jun Li[2], Caio Northfleet[3]

Abstract: The need for using high-level modeling tools (e.g. BPMN) is increasing considerably. The proliferation of the service oriented architectures (SOA) is also apparent. In this context, there is a gap between the developed model and its execution. This work introduces the MoSC Translator which translates models produced in BPMN into executable WS-BPEL processes.

1 Introduction and Related Work

The need for tools which enable high-level specification of business processes and service compositions is apparent. Despite this fact, the existing gap between the modeling tools and the execution resources imposes barriers to the adoption of these strategies. The lack of tools that translate high-level business process models to an executable artifact (e.g. web service composition) contributes significantly to this issue.

In the business processing modeling area, the Business Process Modeling Notation (BPMN) is a standard for specifying business processes using a graphical, high-level and flow-based notation. BPMN could be used to model the *composition* logic of a set of Web services. In such a case, there is a need for explicit representation and visualization of the associated process execution logic. Often, the WS-BPEL standard is used to represent this *execution* logic. Today, business users specify process models in BPMN and send them to a

[1] Federal University of Pernambuco, Centre of Informatics
{blbs,arss,rwam,jcd,faal2,nsr,prmm@cin.ufpe.br}
[2] HP Labs Palo Alto
{bryan.stephenson,hamid.motahari,jun.li@hp.com}
[3] HP Brazil
{caio.northfleet@hp.com}

WS-BPEL programmer, who manually writes the executable code for the associated business process.

There exists semiautomatic translation approaches from BPMN to WS-BPEL (e.g. BPMN2BPEL [2]). However, they do not generate a fully executable BPEL for BPMN models The user must edit the generated model in order to enable the execution of the process (e.g. adding information about the participant services). In particular, the BPMN2BPEL project translates specific BPMN models (with a subset of BPMN elements) to non-executable WS-BPEL service compositions. Furthermore, it does not generate configuration code and needed orchestration engine files that are required to execute the service composition by orchestration engines.

2 The BPMN to WS-BPEL Approach

To fill the gap described in the previous section, we present the MoSC Translator, which translates BPMN models into specific WS-BPEL service compositions. This tool has two main components: the BPMN editor (named Sec-MoSC Editor), that encapsulates all graphical notations commonly used in BPMN models, and the translator itself (MoSC Translator), that is responsible for mapping the BPMN graphical elements (and possible annotations) into an executable WS-BPEL service composition.

In particular, we identify issues that must be addressed in order to enable the automatic translation to executable WS-BPEL processes. We have incorporated the attribute *Tasktype* for tasks, present in the BPMN specification, in the MoSC Editor. A task may have one of the following types: service (provides some type of service), receive (waits for a message from an external participant), send (sends a message to an external participant), user (activity executed by a human with the assistance of software), script (script interpretation by an orchestration engine), manual (performed without the aid of any application), reference (refers to another previously defined task) or none (no type). The screenshot in Figure 1 shows these options.

Current BPMN to BPEL translators do not consider the type of the task (or consider the task type as none). This assumption leads to an abstract BPEL file that cannot be executed directly in any orchestration engine. The proposed approach uses this information (that can be specified in the BPMN diagram) to guide the generation of an executable version of the WS-BPEL process.

One important issue that must be addressed is that technical service information is also needed in order to generate an executable WS-BPEL service composition. As shown in Figure 1, the user may provide information about the service to be associated to the task, such as URI, service name, operation name, service description, business type and so on. From the collected information, an XML file is generated to represent the service information. The following XML code describes the task “Request Flight Availability” that can be found in the BPMN diagram on Figure 1.

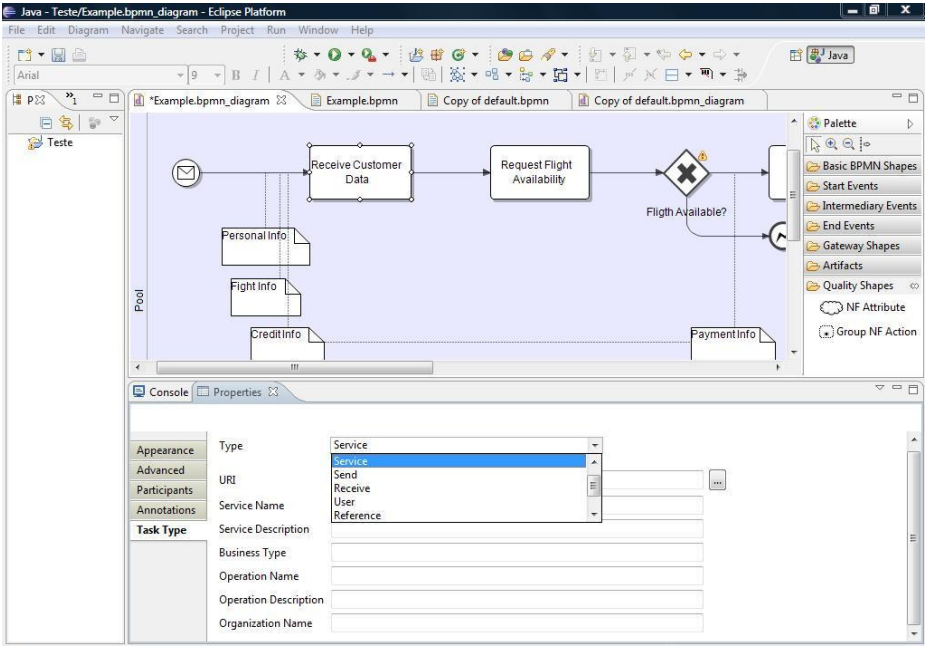

Figure 1 - Sec-MoSC Editor

```
(1)  <?xml version=1.0 encoding=”UTF-8”?>
(2)  ...
(3)  <si:service-information
(4)    xmlns:si=”http://cin.ufpe.br/sec-mosc/service-information”
(5)    xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
(6)    xsi:schemaLocation=”http://www.cin.ufpe.br/sec-mosc/service.xsd”>
(7)  ...
(8)  <si:service-desc>
(9)    <si:service-desc name=”FlightAvailabilityService” taskID=”RFA”>
(10)     <si:businessType> AirLineCompany </si:businessType>
(11)     <si:wsdlNamespace mnemonic=”sac”
(12)      URI=”http://www.cin.ufpe.br/sec-mosc/AirCompany.wsdl”/>
(13)     <si:operationName> requestFAvailability </si:operationName>
(14)     <si:organizationName>Delta AirLine </si:organizationName>
(15) </si:service-desc>
(16) ...
(17) </si:service-information>
```

An XSD schema was created to validate the XML service file. All relevant service information was inserted in the presented code (service name in line 9, taskID in line 9, business type in line 10, operation name in line 13 and organization name in line 14). This information can be retrieved from a service repository made available in the approach. In this repository, all WSDL files associated to the services are stored. The service name, operation

name, input/output/error parameters and message types can be retrieved directly. The information in the repository needs to be provided by human.

Finally, the executable WS-BPEL files are generated using information from the repository and the high-level BPMN editor (BPMN annotated with the task type and associated service information). When the attribute tasktype is stated as "service", the algorithm uses the associated service information to generate needed WS-BPEL information (such as partnerLinks).

3 Demonstration Scenario

To illustrate the application of the MoSC Translator, we use an example service composition called Virtual Travel Agency – VTA (i.e. a travel agency that serves as an interface between many different travel business companies). A diverse number of scenarios were identified, such as: buy a national/international airline ticket, rent a car, book a hotel, buy a tourism packet and check the VISA.

The demonstration starts with the BPMN models of the VTA. We show how a business user can associate a service to a task (using the attribute tasktype). We also present the generation of WS-BPEL code along with the needed configuration files for the execution of the service composition by the orchestration engine.

4 Conclusions and Future Work

In this work, we presented an approach for enabling the generation of executable WS-BPEL code from high-level specifications in BPMN. This approach reduces the gap between the modeling of a service composition and its implementation, facilitating the development of service-oriented applications by high-level users. As future work, we intend to continue the implementation of the MoSC Translator in order to incorporate other task types. More detailed information about the MoSC project can be found in [1].

Acknowledgement. This research is supported by Hewlett-Packard Brasil Ltda. using incentives of Brazilian Informatics Law (Law nº 8.2.48 of 1991).

5 References

[1] Souza, A.R.S. *et al*. "Incorporating Security Requirements into Service Composition: From Modeling to Execution". In 7th International Joint Conference on Service Oriented Computing (ICSOC 2009). Stockholm, Sweden.

[2] White, S. A. (2005), Mapping BPMN to BPEL Example, Technical report, IBM Corporation. Available at www.bpmn.org (last visit: 31st june 2009).