Tool support for Service Oriented development from Business Processes

Andrea Delgado[1], Ignacio García-Rodríguez de Guzmán[2], Francisco Ruiz[2], Mario Piattini[2]

[1] Computer Science Institute, Faculty of Engineering, University of the Republica,
Julio Herrera y Reissig 565, 1300, Montevideo, Uruguay
adelgado@fing.edu.uy
[2]Alarcos Research Group, Dep. of Information Technologies & Systems, University of Castilla - La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, España
{ignacio.grodriguez, francisco.ruizg, mario.piattini}@uclm.es

Abstract. The integration of Business Process Management (BPM), Service Oriented Computing (SOC) and Model Driven Development (MDD) paradigms to improve the development of services oriented solutions from business models is nowadays in the spotlight. Organizations wanting to remain competitive despite the constant changes in their business are paying more attention to their business processes and its base lifecycle. Business process modeling is also at the centre of software development efforts, as making those models explicitly constitutes the basis for services definition. Transformations between business process and services models allow the automatic generation of services from business processes in a repeatable and systematic way, easing the development process. In this paper we present MINERVA's tool support for service oriented development from business processes, including QVT transformations from BPMN to SoaML models to automatically generate service models from business process models.

Keywords: Service Oriented Computing (SOC), Model Driven Development (MDD), Business Process Management (BPM), business processes, service oriented methodologies, model transformations, tool support.

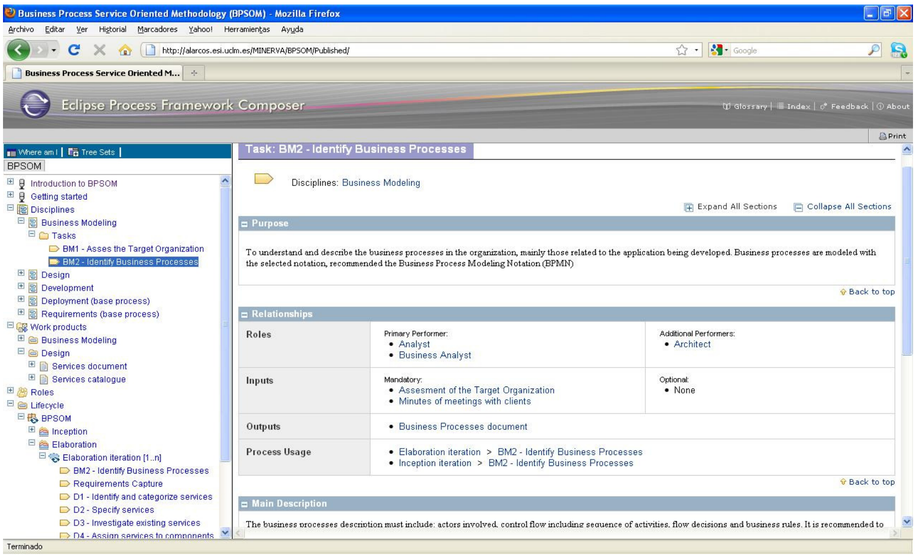
1 Introduction

The development of services oriented solutions to realize business process provides organizations with the needed organizational agility to react to changes, allowing performing changes to each one –business and software – with minimal impact in each other. The integration of Business Process Management (BPM) [1][2], Service Oriented Computing (SOC)[3][4][5] and Model Driven Development (MDD) [6][7][8] paradigms to improve the development of services oriented solutions from business models is nowadays in the spotlight. Organizations wanting to remain competitive despite the constant changes in their business are paying more attention to their business processes and its base lifecycle as defined in [9][10][11]. The implementation of business processes with services also contributes in reducing the gap between the areas of business analysis and Information Technology (IT), easing communication and understanding of business needs. The model driven development supports the definition and maintenance of the relationship between the various models involved, and automates as much as possible the passage from one another by means of transformations. The main objective of the ongoing research work is to provide support to the continuous improvement of business processes based on their lifecycle, applying SOC and MDD paradigms to business process to enable the needed organizational agility. This vision is expressed in MINERVA [12] (Model drIveN and sErvice oRiented framework for the continuous business processes improVement & relAted tools) which is a framework comprising elements in three dimensions: conceptual [13], methodological [14] and tool support, including the Business Process Maturity Model (BPMM) [15] and measures [10][16] for the design and execution of business processes, to guide the improvement effort.

In this article we present MINERVA's tool support for service oriented development from business processes in three main aspects: a method plug-in developed in Eclipse Process Framework (EPF) [17] Composer to support the defined Business Process Service Oriented Methodology (BPSOM) [14]; Query/Views/ Transformations (QVT) [18] transformations from Business Process Modeling Notation (BPMN) [19] models to Service Oriented Architecture Modeling Language (SoaML) [20] models -based on the defined ontology [13]- to generate services automatically, and the technical architecture and selected tools to support MINERVA's lifecycle. The rest of the article is organized as follows: in section 2 the EPF Composer method plug-in to support the methodological approach is described, in section 3 the technical architecture and tools selection of MINERVA are presented, in section 4 the defined QVT transformations are described along with an example, in section 5 related work is presented and finally in section 6 conclusions and future work are discussed.

2 BPSOM method plug-in

The methodological approach defines a Business Process Service Oriented Methodology (BPSOM) [14] as a plug-in to be incorporated in the software base development process used in the organization. The methodology is defined as a method plug-in and published as a web site [21] using the EPF Composer to provide interoperability with other processes defined in the same way. BPSOM defines activities, roles, work products and its templates in three Disciplines defined as key for service oriented development: Business Modeling, Design and Implementation. In Fig. 1 several elements of the methodology are shown, on the left side the defined categories can be seen: Disciplines, Work products, Roles and Lifecycle, along with some of its comprising elements such as activities, tasks, deliverables, roles. On the right side an example of tasks definition is presented for BM2 - Identify Business Processes, showing participating roles, work products defined as inputs and outputs, purpose, description and Discipline that comprises it. Due to space reasons, a brief description of BPSOM is presented here, the complete definition is presented in [14].


Business Process Service Oriented Methodology (BPSOM) - Mozilla Firefox
Archivo
Editar
Ver
Historial
Marcadores
Yahoo!
Herramientas
Ayuda
http://alarcos.esi.uclm.es/MINERVA/BPSOM/Published/
Google
Business Process Service Oriented M...
Eclipse Process Framework Composer
Glossary
Index
Feedback
About
Print
Where am I
Tree Sets
BPSOM
Introduction to BPSOM
Getting started
Disciplines
Business Modeling
Tasks
BM1 - Asses the Target Organization
BM2 - Identify Business Processes
Design
Development
Deployment (base process)
Requirements (base process)
Work products
Business Modeling
Design
Services document
Services catalogue
Roles
Lifecycle
BPSOM
Inception
Elaboration
Elaboration iteration [1..n]
BM2 - Identify Business Processes
Requirements Capture
D1 - Identify and categorize services
D2 - Specify services
D3 - Investigate existing services
D4 - Assign services to components
Terminado
Expand All Sections
Collapse All Sections
Back to top
Back to top

Task: BM2 - Identify Business Processes

Disciplines: Business Modeling

Purpose

To understand and describe the business processes in the organization, mainly those related to the application being developed. Business processes are modeled with the selected notation, recommended the Business Process Modeling Notation (BPMN)

Relationships

Roles	Primary Performer: • Analyst • Business Analyst	Additional Performers: • Architect
Inputs	Mandatory: • Assesment of the Target Organization • Minutes of meetings with clients	Optional: • None
Outputs	• Business Processes document	
Process Usage	• Elaboration iteration > BM2 - Identify Business Processes • Inception iteration > BM2 - Identify Business Processes	

Main Description

The business processes description must include: actors involved, control flow including sequence of activities, flow decisions and business rules. It is recommended to

Fig. 1. Global view of BPSOM web site created using EPF composer

The defined lifecycle is iterative and incremental following the unified process[22] incorporating the four phases: Inception, Elaboration, Construction and Transition. The activities are inserted in the phases of the base software development process, making the corresponding adaptations when needed. The defined lifecycle serves as a guide to indicate the emphasis on the realization of activities at each stage of development. As an example, the activity's worfklow for the defined Elaboration iteration is shown in Fig. 2.

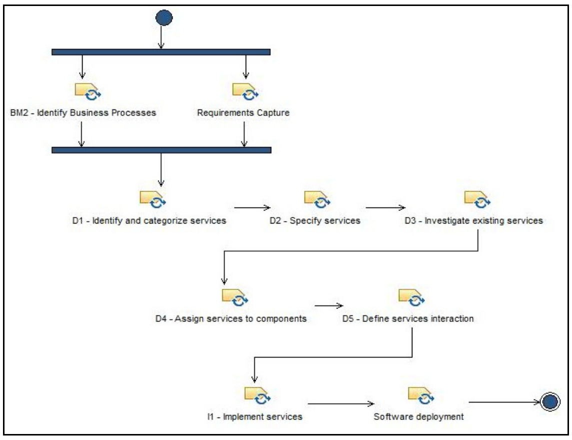

BM2 - Identify Business Processes
Requirements Capture
D1 - Identify and categorize services
D2 - Specify services
D3 - Investigate existing services
D4 - Assign services to components
D5 - Define services interaction
I1 - Implement services
Software deployment

Fig. 2. Activity's workflow for an Elaboration iteration in BPSOM

Disciplines and activities. BPSOM activities are defined in three key Disciplines to guide the service oriented development effort: Business Modeling, Design and Implementation. These activities have to be integrated with other existing activities of the base development process, such as requirements capture, architecture definition, service testing and deployment, and project management.

Business Modeling. The Business Modeling Discipline aims to understand and describe the business processes in the organization, mainly those related to the application being developed. Business processes are modeled with the selected notation, recommended the Business Process Modeling Notation (BPMN). It also promotes the involving of the project team with the organization for which the development is being carried out, in issues such as: the area of business, operation, employees, etc. of the organization. There are two activities defined to reach these goals: BM1 – Asses the target Organization and BM2 – Identify Business Processes.

Design. The Design Discipline adds the following goals to the ones defined for generic software design: to identify and specify the services needed to perform the business processes modeled in the Business Modeling Discipline, classifying them by type of service, to generate and maintain a services catalogue for services reuse in the organization (functionalities, components), and to define the services composition (orchestration, choreography) needed to realize the identified business processes. Five activities are defined to reach these goals: D1 – Identify and categorize services, D2 – Specify services, D3 – Investigate existing services, D4 – Assign services to components and D5 – Define services interaction.

Implementation. The main goal of the Implementation Discipline is to get the identified services implemented and working as defined. To reach this goal the activity I1 – Implement services is defined.

Work products. Work products provide the basis for controlling development progress and reaching the defined goals of the activities, Disciplines and lifecycle iterations. The defined work products in BPSOM are as follows: Assessment of the target organization, Business Processes document, Services document, Services catalogue and Services implemented. Each document has a defined template to guide its realization, defining its content.

Roles. The roles in BPSOM are a selection of base roles that we consider to be the more important ones for service oriented development based on business processes: Analyst, Business Analyst, Architect and Developer. Each role performs several activities as primary role, and is responsible for some work products realization. The Business Analyst is explicitly included now to emphasize the importance of the participation of Business people in the Business Modeling Discipline.

3 Technical architecture and tools

To support service oriented development from business processes following BPSOM methodology, a selection of existing tools is given. The aim of MINERVA framework is to integrate free existing tools facilitating the development using BPSOM; for every activity a tool is recommended for its realization. So far we are working with an integrated selection of tools available in the Eclipse IDE: Eclipse BPMN Modeler [23] for business process modeling, Medini QVT [24] Eclipse plug-in for business process to services QVT transformations, Magic Draw Cameo SOA+ [25] Eclipse plug-in for SoaML modeling and ModelPro [26] Eclipse plug-in for JEE code generation. The technical architecture and the first selection of tools are shown in Fig. 4, to support the complete MINERVA's lifecycle.

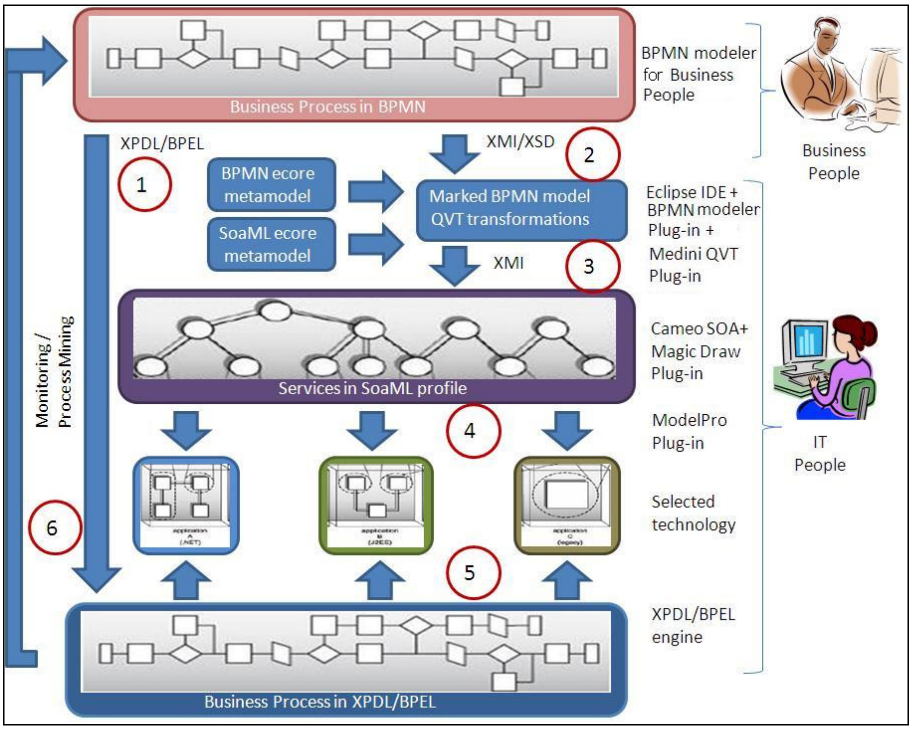

Business Process in BPMN
BPMN modeler for Business People
Business People
XPDL/BPEL
XMI/XSD
2
1
BPMN ecore metamodel
SoaML ecore metamodel
Marked BPMN model QVT transformations
XMI
3
Eclipse IDE + BPMN modeler Plug-in + Medini QVT Plug-in
Monitoring / Process Mining
Services in SoaML profile
Cameo SOA+ Magic Draw Plug-in
4
ModelPro Plug-in
IT People
6
Selected technology
5
XPDL/BPEL engine
Business Process in XPDL/BPEL

Fig. 4. MINERVA tool support for services development from business processes

As it can be seen in Fig. 4 and according to BPSOM, the first activity shown corresponds to the business process modeling that is done by Business People, in a BPMN modeler. After the business process model is defined, in step (1) the business process execution model is obtained expressed in XPDL [27]/BPEL [28], and in step (2) the business process model is exported in XMI/XSD format, so it can be imported for IT People in the Eclipse IDE. Several tools allow these kinds of export files from business process models, and with the release of BPMN 2.0 [29] we expect the format would be homogeneous for all tools. After the BPMN model is imported in Eclipse IDE, activities that will be transformed to services have to be marked as of "Service" type according to the service design activities from BPSOM, defining its input and output messages, operations, implementation, among others. Running the QVT transformations defined in Medini QVT plug-in in step (3), the corresponding elements in SoaML profile are obtained. The inputs for the QVT transformations are the source model in XMI format that is the XMI file associated with the BPMN business process model, and the two corresponding metamodels: BPMN Modeler and SoaML metamodels in ecore format, from which the QVT transformations are defined. Then the target XMI file corresponding to the SoaML model that is generated by the transformation is loaded into Magic Draw Cameo SOA+ plug-in, and in step (4) using ModelPro plug-in the code is generated from the obtained SoaML model. Step (5) represents the invocation of the generated service components from the business process execution engine. Finally, in step (6) monitoring of the business process execution and its evaluation by means of techniques such as Process Mining [30] over log files is done, using tools as Prom [31] and its analysis plug-ins.

Business Process Modeling Tools. For business process modeling we are evaluating several tools that implements BPMN standards, including BizAgi modeler which allows XPDL export, Visual Paradigm Business Process Architect which allows XMI export, as it is desirable that Business People could model the business processes to give them to IT People. For our proof of concept we are using directly the Eclipse BPMN modeler as we want to focus on QVT transformations.

Service Oriented Modeling Tools. There are few implementations of the SoaML specification yet that can be seen in [32]. Although we are analyzing the functionalities provided by each of them along with it ease of use, we have integrated the MagicDraw CAMEO SOA+ and the ModelPro Eclipse plug-ins for the proof of concept, as we are using Eclipse as base platform. SoaML implementations provide the needed stereotypes to specify SoaML models, and to generate code in the desired technology. We are planning our own implementation of SoaML as Eclipse plug-in.

Code generation from services in SoaML. The ModelPro plug-in generates JEE code from SoaML models, providing several examples. For the generation of code for our own example we need to add elements to the generated SoaML model, via new defined transformations or manually, to provide the MDA engine with the required information for doing the generation of code.

4 QVT transformations

BPSOM methodology guides the derivation of services from business processes by means of activities, roles and work products definition. To improve the development process, the automatic generation of services oriented models from business process models is added, which is presented in the following.

4.1 QVT transformations definition

To define QVT transformations between service models and business process models, we previously defined an ontology [13] in the conceptual dimension of MINERVA, in which we identified the relations between these models, in order to understand what elements we want to obtain in service models from elements in business process models. Following the Model Driven Architecture (MDA) [8] standard, we transform BPMN models into SoaML models by defining QVT transformations from a BPMN metamodel as origin metamodel, to the SoaML metamodel as the target metamodel. Then, a BPMN model compliant with the BPMN metamodel will be transformed into a SoaML model compliant with the SoaML metamodel.

There are many elements in each metamodel to be related, so to begin with we selected a sub-set of key concepts and relationships to define the QVT transformations that allow us to obtain the structural view of services, from the business process model. Other views will be also modeled as the dynamic view of services interaction specified by sequence diagrams, as defined in BPSOM. The sub-set of concepts and relationships that we have defined in the ontology and used in the QVT transformations definition are shown in Fig.4; the complete set of concepts and relations is detailed in [13].

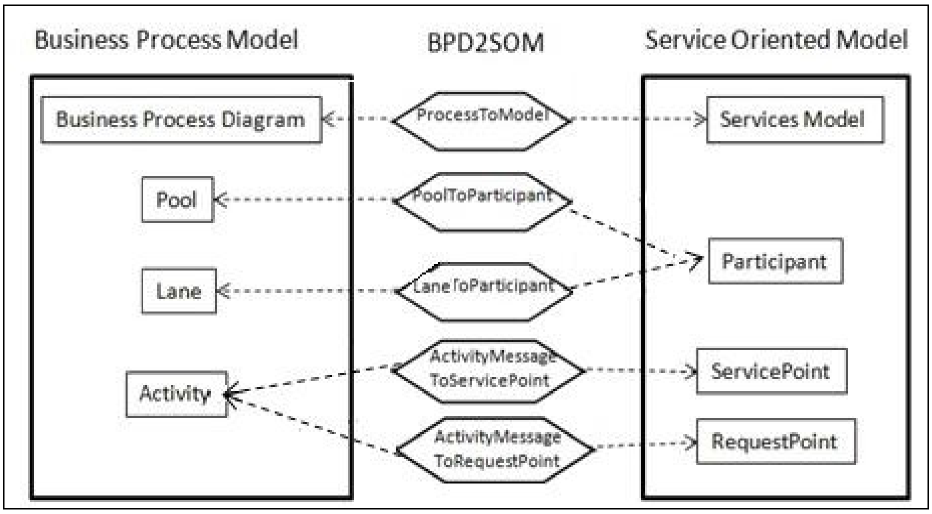


Fig. 4. Key concepts and relationships used in QVT transformations

First of all the services model is obtained from the business process diagram (BPD) as the root element. Each pool in the BPD defines a participant in the services architecture model, and each lane in each pool corresponds to an internal participant. Each participant provides and requires services corresponding to exchanged messages with other participants, so, the generated services will be associated with each participant. Each activity marked as “Service Type” will be transform into a ServicePoint or RequestPoint, depending on the direction of the message: if it is an incoming message then the service is provided in the generated ServicePoint, if it is an outgoing message then the service is request in the generated RequestPoint. We are aware that there are many other alternatives to specify these transformations, which we will explore in our future work. In Table 1 the QVT transformations from BPMN to SoaML comprising the concepts and relationships shown in Fig.4 are presented.

Table 1. Sub-set of the defined QVT transformations from BPMN to SoaML

Sub-set of relation rules defined
top relation ProcessToModel { checkonly domain bpmn bp : bpmn::BpmnDiagram{name = pn}; enforce domain soaml sm : SoaML::Model{name = pn }; }
top relation PoolToParticipant { checkonly domain bpmn p : bpmn::Pool{name = pn}; enforce domain soaml s : SoaML::Participant{name = pn};}

```
top relation LaneToParticipant {
   checkonly domain bpmn p : bpmn::Lane{name = pn };
   enforce domain soaml s : SoaML::Participant{ name = pn };}
top relation ActivityMessageToServicePoint {
   checkonly domain bpmn c : bpmn::Activity{lanes = p : bpmn::Lane{},
         activityType = bpmn::ActivityType::Task,
         incomingMessages = im : bpmn::MessagingEdge{},  name = cn};
  enforce domain soaml t : SoaML::ServicePoint {
         participant = s : SoaML::Participant {},
         isService = true,  name = cn};
         when { p.pool.bpmnDiagram.pools.lanes.activities -> exists
               (x:bpmn::MessageVertex | (x.outgoingMessages.target =
               c.incomingMessages.target) and
               (x.oclAsType(bpmn::Activity).activityType=c.activityType));
               PoolToParticipant (p.pool, s); }}
```

The first rule shown in Table 1 named “ProcessToModel” generates a SoaML “Model” element from a “BpmnDiagram” element corresponding to the BPD of BPMN, relating the top level business process to the services model. The second rule named “PoolToParticipant” generates SoaML “Participant” elements from “Pool” elements of the BP model, one Participant from each Pool. The third rule named “LaneToParticipant” generates SoaML “Participant” elements from “Lane” elements inside the pools of the BP model, which will be used to describe the internal architecture of each participant in the services architecture. The fourth rule named “ActivityMessageTo ServicePoint” generates SoaML “ServicePoint” elements from those activities in the BP model which have incoming messages from another activity, associating it with the participant that provide the service. In this rule it is necessary to look for all the activities in the BPMN model to see which are connected by messages connectors. The OCL expression in the “when” clause of the rule, evaluate all the activities in the model, checking whether they have outgoing messages to the activity being evaluated, that is incoming messages, and comparing the type of activities to be “Activity” instead of “Service” as we defined. This is a restriction of the BPMN Modeler metamodel in which activity elements are defined as “Activity”, “Gateway”, etc. and are differentiated with the ActivityType property, not providing the types “Service”, “Manual”, etc that we defined to use, which is minimal for the purpose of demonstrating the feasibility of QVT transformations from BPMN metamodel to SoaML metamodel.

4.2 QVT transformations example

To illustrate the proposal for SoaML service models generation from BPMN business process models, the Make Appointment business process from a generic hospital is modeled using the Eclipse BPMN Modeler, as shown in Fig. 5.

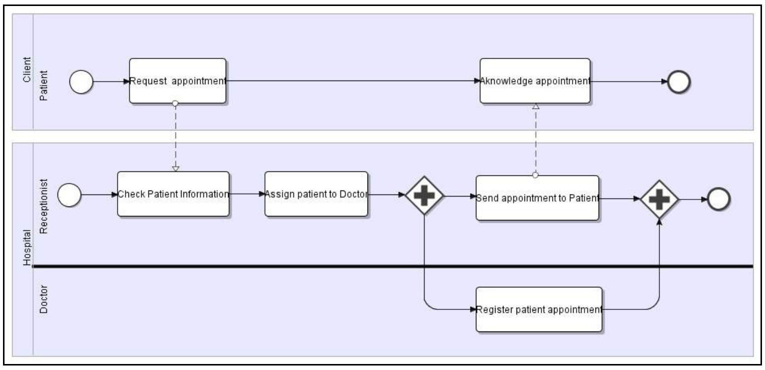
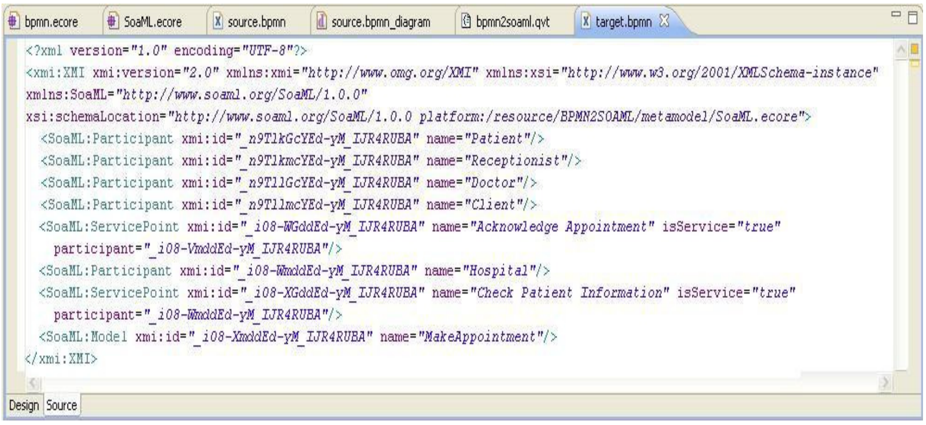


Fig. 5. Make Appointment business process

The business process shown in Fig. 5 starts when a Patient requests an appointment with a Doctor. The Receptionist checks the Patient information assigning a Doctor. After that, two activities are executed in parallel: the Doctor registers the appointment with the Patient and a communication is sent to the Patient with the appointment's information. From this business process we want to obtain two Participants: Client and Hospital, three Participants including in the previous ones: Patient for Client, and Receptionist and Doctor for Hospital. The Client participant offers a service in the activity "Acknowledge Appointment", and requires a service from the Hospital in the "Check Patient Information" activity. The Hospital participant offers a service in the activity "Check Patient Information", and requires a service in the activity "Acknowledge Appointment". After executing the defined QVT transformations the target XMI file is obtained with the SoaML elements generated, as shown in Fig. 6.



```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SoaML="http://www.soaml.org/SoaML/1.0.0"
xsi:schemaLocation="http://www.soaml.org/SoaML/1.0.0 platform:/resource/BPMN2SOAML/metamodel/SoaML.ecore">
  <SoaML:Participant xmi:id="_n9TlkGcYEd-yM_IJR4RUBA" name="Patient"/>
  <SoaML:Participant xmi:id="_n9TlkmcYEd-yM_IJR4RUBA" name="Receptionist"/>
  <SoaML:Participant xmi:id="_n9TllGcYEd-yM_IJR4RUBA" name="Doctor"/>
  <SoaML:Participant xmi:id="_n9TllmcYEd-yM_IJR4RUBA" name="Client"/>
  <SoaML:ServicePoint xmi:id="_i08-WGddEd-yM_IJR4RUBA" name="Acknowledge Appointment" isService="true"
    participant="_i08-VmddEd-yM_IJR4RUBA"/>
  <SoaML:Participant xmi:id="_i08-WmddEd-yM_IJR4RUBA" name="Hospital"/>
  <SoaML:ServicePoint xmi:id="_i08-XGddEd-yM_IJR4RUBA" name="Check Patient Information" isService="true"
    participant="_i08-WmddEd-yM_IJR4RUBA"/>
  <SoaML:Model xmi:id="_i08-XmddEd-yM_IJR4RUBA" name="MakeAppointment"/>
</xmi:XMI>
```

Fig. 6. SoaML target XMI file generated from the BPMN business process

The two participants and the three included participants are generated; for the Client participant the ServicePoint "Acknowledge Appointment" is created, and for the Hospital participant the ServicePoint "Check Patient Information" is obtained. For the generation of the associated RequestPoint we are working on the rules to navigate through the messages to the target participants that provide them. After the SoaML XMI file is generated, it has to be loaded into the Magic Draw Cameo SOA+ plug-in to show the SoaML diagrams and to generate the code by the ModelPro engine.

5 Related work

In the last few years there have been many efforts aiming to apply SOC and MDD paradigms to business processes, from which a selection of works that defines methodological approaches or aims to automate the generation of service and software artifacts from business process are presented in the following. As the best of our knowledge there are no other works relating BPMN models with SoaML models directly as we do, but transformations from BPMN to UML can be seen in [33] where UML artifacts as Activity Diagrams (AD) and Collaboration and Deployment diagrams are generated from BPMN business process models, as a way to travel form business to IT vision, and [34] where BPMN business processes are transformed into AD and from these diagrams use cases and analysis classes are generated, with focus on business process security. Model driven approaches are proposed in [35] including a methodology, models, metamodels and transformations to obtain service oriented Web Information Systems (WIS) expressing the interaction of services to perform business processes, in [36] a business value model is integrated to the work to derive software artifacts from it using ATL[37], in [38] collaborative service oriented architecture is defined and transformations from BPMN models into UML models and BPEL models also using ATL. In [39] models and metamodels for services are defined in the PIM4SOA approach relating them to the defined architecture (brokerless, centralized and decentralized broker). Transformations based on the application of patterns are proposed in [40] starting with the macroflow-microflow pattern which establishes the conceptual basis and the process-based integration architecture pattern that guides the design of an architecture based on sub-layers for the service composition layer. In [41] conceptual transformations are defined based on the successively application of patterns from the top to the bottom layer, using graphs for pattern matching. [42] proposes going from BP models to technical processes matching existing services by applying transformation patterns classified with respect to the quality of the transformation. [43] goes from a business model (CIM) to an analysis model (PIM), identifying serviceAction in tasks, then to a Design model (Architecture specific model, ASM), mapping services to the target architecture. In [44] a methodology for the development of services associated with business processes is defined with focus on Web Services (WS) implementation, and in [45] a survey on existing approaches for identification and analysis of Business and Software Services along with a consolidated proposal is presented.

6 Conclusions and future work

The tool support presented in this article is part of the MINERVA framework we are working on, to support the continuous business process improvement based on the business process lifecycle for BPM activities. The described elements comprise a methodology BPSOM as a basis for the development process, QVT transformations to automate as much as possible the defined steps for services generation, and a set of integrated tools using as a basis the Eclipse open IDE. BPSOM is modeled in the EPF Composer as a method plug-in to be integrated in the existing software development process used in the organization, with this tool. A proof of concept example was developed using the selected tools, to illustrate the steps defined in the proposal to go from a BPMN business process model to a SoaML service oriented model. The defined QVT transformations although simple serve as the basis for the further definition of transformations for the remaining SoaML elements. We believe that MINERVA could be a useful guide to be used in organizations that need rapid and easy integration of methodologies, tools and concepts to adopt the BPM, SOC and MDD paradigms. Our future work is now focused on the definition of the remaining QVT transformations to effectively generate complete SoaML services models from BPMN business processes models, from which to obtain the associated code. We are also working in an implementation of the SoaML profile as an Eclipse plug-in, and we plan to integrate the method plug-in BPSOM into Eclipse to automatically aid in the realization of the methodology activities' flow.

Acknowledgments. This work has been partially funded by the Agencia Nacional de Investigación e Innovación (ANII) from Uruguay; ALTAMIRA project (Junta de Comunidades de Castilla-La Mancha, Spain, Fondo Social Europeo, PII2I09-0106-2463) and PEGASO/MAGO project (Ministerio de Ciencia e Innovacion MICINN, Spain, and Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01).

References

1. Business Process Management Initiative, <http://www.bpmi.org/>
2. Smith,H.,Fingar,P.,Business Process Management: The third wave, Meghan-Kieffer, (2003)
3. Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenge, IEEE Computer Society, (2007)
4. Krafzig, D. Banke, K. Slama, D., Enterprise SOA, Service Oriented Architecture: Best Practices, Prentice Hall, 1st. ed., (2005)
5. Erl, T., SOA: Concepts, Technology, and Design,Prentice Hall, (2005)
6. Mellor, S., Clark, A., Futagami, T., Model Driven Development - Guest editors introduction, IEEE Computer Society, September/October (2003).
7. Stahl, T.; Volter, M. et. al.: Model-Driven Software Development, Technology, Engineering, Management, John Wiley & Sons, Ltd., (2006)
8. Model Driven Architecture (MDA) v. 1.0.1, OMG, http://www.omg.org/mda, (2003)
9. Weske, M., BPM Concepts, Languages, Architectures, Springer, (2007)
10. Mendling, J., Metrics for process models, Springer, (2008)
11. van der Aalst, W.M.P., ter Hofstede, A., Weske, M., Business Process Management: A Survey, In: International Conference on Business Process Management, (2003)
12.Delgado, A., Ruiz F., García - Rodríguez de Guzmán I., Piattini M., MINERVA: Model drIveN and sErvice oRiented framework for the continuous business processes improVement & relAted tools, In: 5th International Workshop on Engineering Service-Oriented Applications (WESOA'09), Stockholm, (2009), Springer (2010) in press.
13. Delgado, A., Ruiz, F., García - Rodríguez de Guzmán, I., Piattini, M.: Towards an ontology for service oriented modeling supporting business processes, In IV International Conference on Research Challenges in Information Science (RCIS'10), Niza, (2010)
14. Delgado, A., Ruiz, F., García - Rodríguez de Guzmán, I., Piattini, M.: Towards a Service-Oriented and Model-Driven framework with business processes as first-class citizens, In: 2nd Int. Conf. on Business Process and Services Computing (BPSC'09), Leipzig, (2009)

15. Business Process Maturity Model (BPMM), OMG, http://www.omg.org/spec/BPMM

16.Sánchez, L., Delgado, A., Ruiz, F., García, F., Piattini, M.: Measurement and Maturity of Business Processes. Eds.: Cardoso, J., van der Aalst, W.,Handbook of Research on Business Process Modeling, Information Science Reference (IGI Global),pp.532-556, (2009)

17. Eclipse Process Framework Composer (EPF Composer), http://www.eclipse.org/epf/

18.Query/Views/Transformations(QVT),v.1.0,OMG,http://www.omg.org/spec/QVT/1.0,(2008)

19. Business Process Modeling Notation (BPMN), OMG, http://www.omg.org/spec/BPMN/

20 Soa Modeling Language (SoaML),OMG, http://www.omg.org/spec/SoaML/, (2009)

21. BP Service Oriented Methodology (BPSOM) http://alarcos.esi.uclm.es/MINERVA/BPSOM/

22.Jacobson, I., Booch, G., Rumbaugh, J. The Unified Software Development Process, Addison-Wesley, (1999)

23. SOA Tools Platform (STP) BPMN Modeler, http://www.eclipse.org/bpmn/

24. Medini QVT, ikv++ technlogies ag, http://projects.ikv.de/qvt/

25. Magic Draw CameoSOA+,http://www.nomagic.com/ text.php?lang=2&item=338&arg=295

26. ModelPro, http://modeldriven.org/

27. XML Process Definition Language (XPDL), v.2.1, WfMC, http://www.wfmc.org/xpdl.html

28.WS BP Execution Language (WS-BPEL),OASIS, http://docs.oasisopen.org/wsbpel/2.0/

29.BPModeling Notation (BPMN), v.2.0, OMG, http://www.omg.org/spec/ BPMN/2.0/, (2009)

30. van der Aalst, W.M.P., Reijers, H. A., Medeiros, A.,Business Process Mining: an Industrial Application, Information Systems Vol.32 Issue 5, 713-732, (2007)

31.ProM, Process Mining Group, Eindhoven University of Technology, Eindhoven, The Netherlands, http://prom.win.tue.nl/research/wiki

32. SoaML implementations, http://www.omgwiki.org/SoaML/ doku.php?id=tool_support

33.Liew,P., Kontogiannis,K. Tong,T., A Framework for Business Model Driven Development,12th Int. Workshop on Se Tech. and Engineering Practice (STEP'04), (2004)

34.Rodríguez,A.; Fernández-Medina, E.; Piattini, M.: Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases. 5th Int. Conf. on BPM (BPM'07)(2007)

35. de Castro, V., Marcos, E., López Sanz, M., A model driven method for service composition modelling: a case study, Int. J. Web Engineering and Technology, Vol. 2, No. 4, (2006)

36.de Castro V., Vara Mesa J. M., Herrmann E., Marcos E., A Model Driven Approach for the Alignment of Business and Information Systems Models, (2008)

37. Jouault, F., Kurtev, I.,Transforming Models with ATL (ATLAS Transformation Language), Satellite Events at the MoDELS Conference, (2005)

38. Touzi J., Benaben F., Pingaud H., Lorré J.P., A model-driven approach for collaborative service-oriented architecture design, Int. Journal of Prod. Economics,Vol.121 Is.1, (2009)

39. Roser,S., Bauer,B., Muller,J., Model- and Architecture-Driven Development in the Context of Cross-Enterprise Business Process Engineering", International Conference on Services Computing (SCC'06), (2006)

40.Zdun, U., Hentrich, C., Dustdar, S., Modeling Process-Driven and SOA Using Patterns and Pattern Primitives, ACM Transactions on the Web, Vol. 1, No. 3, Article 14, (2007)

41. Gacitua-Decar V., Pahl C., Pattern-based business-driven analysis and design of service architectures, 3rd Int. Conf. on Software and Data Technologies SE (ICSOFT'08), (2008)

42. Henkel,M., Zdravkovic,J., Supporting Development and Evolution of Service-based Processes, International Conference on e-Business Engineering (ICEBE'05), (2005)

43. Herold S., Rausch A., Bosl A., Ebell J., Linsmeier C., Peters D., A Seamless Modeling Approach for Service-Oriented Information Systems, 5th International Conference on Information Technology:New Generations(ITNG 08), (2008)

44.Papazoglou, M., van den Heuvel, W., Service-oriented design and development methodology, Int. J. Web Engineering and Technology, Vol. 2, No. 4, pp.412-462, (2006)

45. Kohlborn T., Korthaus A., Chan T., Rosemann M., Identification and Analysis of Business and SE Services- A Consolidated Approach, IEEE Transactions on Services Comp., (2009)