Modeling and Generating Mobile Business Processes

Lasse Pajunen
Nokia Research Center
Nokia Corporation
P.O.Box 407, FI-00045 Nokia Group, Finland
lasse.pajunen@nokia.com

Anna Ruokonen
Institute of Software Systems
Tampere University of Technology
P.O. Box 553, FI-33101 Tampere, Finland
anna.ruokonen@tut.fi

Abstract

A mobile business process is a special case of a business process where most of the human interaction is done using mobile devices. In this paper, we propose UML-based support for developing such mobile business processes. Here, the business process is first modeled using UML. Then the process model is translated into a BPEL description, which can be run in mobile and/or network-based workflow engines. We propose rules to guide modeling of mobile business processes, to import existing WSDL decriptions into UML models, and to generate executable BPEL descriptions with appropriate WSDL definitions. In addition, we introduce our implementation of the approach. The practical applicability is demonstrated by designing a group messaging process. It provides a customizable mobile device based communication service offering a business case for mobile operators.

1 Introduction

A business process is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal [21]. A mobile business process is one kind of a business process where most of the human interaction is done using mobile devices.

Business processes are automated by workflow management systems. *A workflow management system* is a system that defines, creates and manages the execution of business processes through the use of software that runs on one or more workflow engines. A workflow management system is able to interpret the process definition, interact with workflow participants and where required, invoke the use of IT tools and applications [21]. *A mobile workflow* management system is an extension to traditional workflow management systems. Here, some parts of the processes are executed in mobile devices. Especially, human interaction part of the business process is executed in mobile devices.

Different mobile devices support different set of communication protocols and applications interacting with users. Firstly, short message services (SMS) are developing very rapidly and they are basically available in all type of mobile phones. Secondly, modern mobile devices can have support for Web services and they are also becoming an integral part of service and workflow systems. Web services allow more data intensive communication than SMS between mobile devices and network based services. In addition, service-based approach allows using specialized software in the device to increase usability. Thirdly, some mobile devices have proper Web browsers. Here, the interaction model is similar to traditional Web browsing.

Creating services for mobile devices require flexible development and deployment technologies to support all of these different scenarios. Workflows are a good solution here. They allow describing services in an abstraction level that can be easily automatically adapted to used scenario. Workflow execution can be distributed between mobile and network-based devices based on device capabilities. Depending on mobile devices, concrete protocols and used applications can be automatically selected. So, workfows provide flexible and distributable way to interact with the users using various mobile devices. As an example of a concrete service, mobile data services can be used to assign a new job assignment to an employee on the road with interaction and feedback on job execution.

A process definition is a representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by workflow management system [21]. Business processes, as discussed in this paper, interact with external entities implemented as Web services. Discovering and communication of Web services is based on WSDL descriptions [20]. WSDL description defines the service interface including provided operations and used data types. In addition to interface definition, it defines service name and location. WSDL descriptions can be used as a basis for constructing Web service systems, but the language itself does not provide any mechanisms for compos-

2007 IEEE International Conference on Web Services (ICWS 2007)
0-7695-2924-0/07 $25.00 © 2007 IEEE

ing such a processes.

Several languages have been developed for service coordination and orchestration. BPEL [6] and XPDL [22], for example, are developed for composing Web services into business processes. Currently, BPEL is the most established language used for Web service composition, especially service orchestration. BPEL can be used to define executable business processes, which can be run in specific workflow engines, for example ActiveBPEL engine [1] or a mobile workflow engine.

When an organization wants to automate its business processes, a designer needs to create aforementioned process definitions for execution in workflow management systems. The manual creation of BPEL, WSDL, and other XML-based descriptions is cumbersome. It would be better if some graphical notation could be used. In addition, standardized notations would enable development of appropriate tools and support from training organizations. There are basically two graphical notations to express the BPEL language: BPNM and UML Activity diagrams. OMG, which is the development organization of UML, promotes a model-driven development (MDD) [12] approach, which uses models as primary development artifacts in software engineering lifecycle.

We propose UML2-based rules for designing mobile processes with mapping to executable BPEL descriptions. An important part of the mobile process model is WSDL definitions, which define the services implementing the mobile process. In this paper, we also report a MDD fashion tool, which supports our approach and implements transformations from UML2 to BPEL and WSDL. Our implementation is an extension to an existing IBM's UML2 CASE-tool. An overview of our approach is presented in Figure 1. The first step is to locate existing WSDLs and to translate them into UML model. Based on operations defined in the imported WSDLs, a selection of process actions are generated. Finally, the mobile process model is translated into executable BPEL code. If needed, the WSDL descriptions, including BPEL specific extensions, can be exported as well.

The rest of the paper is organized as follows. In Section 2, we will describe modeling based scenarios for developing mobile processes. In Section 3, we will propose UML2-based rules for modeling the processes. In Section 4, we will describe our experiences in implementing and applying this approach. In Section 5, we will analyze currently existing solutions on designing workflows. Here, we will analyze different related approaches and tools, including usage of different business process description notations. And finally, Section 6 will conclude the paper.

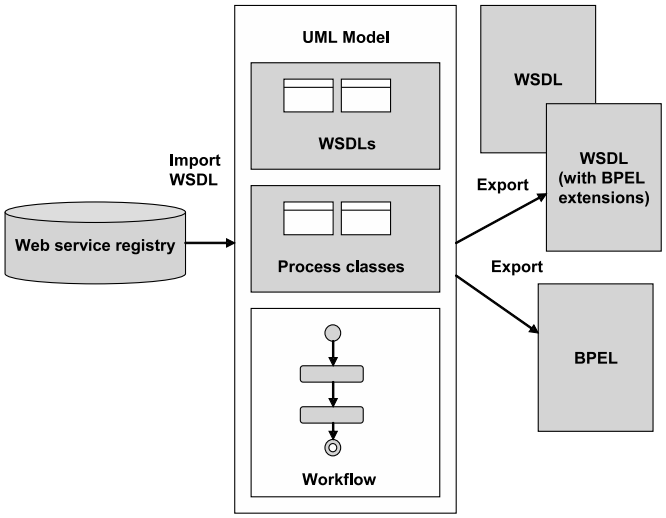

Figure 1. Process development

2 Model-based mobile process development

An overview of the approach was already presented in the Figure 1. From the approach, the following development phases can be identified:

- Import existing WSDL descriptions,
- Create template for the mobile process model,
- Define the mobile process model,
- Export WSDL descriptions, and
- Export BPEL decription.

Different mobile process development scenarios supported by our approach are presented more detail in the following sections. To illustrate the development scenarios, we use a concrete example: development of mobile business process, called *Group Messaging Process*. The idea is that a mobile phone company wants to provide a customizable process, which can be included in mobile phone operators' service portfolios. The operators can tailor the process for their clients, namely, mobile phone users.

2.1 Group messaging process

Group messaging (GM) process provides messaging service with ability to manage communication groups, and to send and receive messages among a user group. The service can be deployed to support various device types and various communication protocols. Figure 2 presents all components participating GM Process: GM Data Service, GM Core process, GM SMS UI process, GM Advanced UI process, GM

SMS gateway, HTTP gateway, Web server, Phone SMS UI, Phone Application UI, and Browser.

GM Data Service provides a database access implemented as a Web service. Messaging groups, user details, and messaging history are stored in the database. Data manipulation and query operations are provided through WSDL described interface. *GM Core Process* provides an abstraction between users and the database. It provides a customized group messaging service. *GM SMS UI Process*, *GM Advanced UI Process*, and *Web server* generated Web pages enable connection to user interfaces. *SMS gateway* and *HTTP gateway* are infrastructure services to enable contacting mobile devices from the network.

GM Core, *GM SMS UI* and *GM Advanced UI* processes form a complete mobile business process. In this scenario, the mobile business process is deployed on these three subprocesses. *GM Core* and *GM SMS UI* processes are executed in the network and *GM Advanced UI Process* is executed in the mobile device.

The initialize method creates a new process instance and creates a new user group in the database. After the process is created, new users can be added in the group and messaging service is ready to use. The actual 'send message' requests are delegated to another process, *GM Core Process*, which is responsible of message handling. It creates and sends the messages. The process can be used with *Browser UI*, basic *Phone SMS UI*, or advanced *Phone Application UI*. A user interface provides the user a message inbox and outbox, and support for sending messages to individual users or all users belonging the same messaging group. *GM Core Process* is responsible for updating UI processes according to the database.

2.2 Starting from WSDLs

Let us assume that we already have the database and SMS gateway services defined as service descriptions *SMSService.wsdl* and *GroupCommunicationDatabase.wsdl*. The aim is to design a group messaging process, which reuses these predefined services. The process development is started by specifying the locations of the existing WSDLs and importing them into the process model as UML class model representation.

WSDL definitions are important parts of the process model. WSDL documents are translated into UML models defining for example port types, operations, messages and data types. In addition, a process class with variables (corresponding in and out message in WSDL) is created. The process class includes attributes for defining target and business namespaces Each WSDL file is translated into one UML package (stereotyped as *wsdl*) containing all the WSDL elements.

WSDL models are assumed to contain BPEL extensions

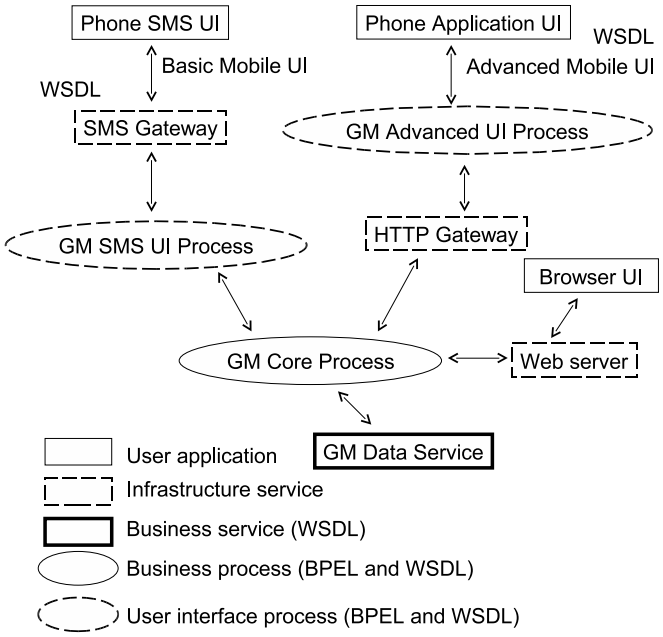

Figure 2. Group Messaging (GM) Scenario

(i.e. in our case partner link type definitions). If the original imported WSDL does not contain BPEL extensions, missing WSDL elements are generated and placed in the UML model. Process-related classes are placed in a package stereotyped as *process*. From partner link types, partner links and partner definitions are also generated into process package.

One activity diagram for the process flow is created as well. For each operation, one *call behavior action* is generated. The generated actions can be (tracked and dropped) edited and used as base elements when modeling the actual workflow. For example, we can use actions to invoke database operations inside GM process and we can send SMS messages by invoking operations provided by message service. Figure 3 presents a part of the generated WSDL model (on top) and an automatically generated invoke action (on the bottom). Operation name 'createUser' is used as action name, the *input pin* type is operation's input message 'createUserRequest' and the *output pin* type is operation's output message 'createUserResponse'.

2.3 Starting from an empty model

If there are no existing services to be imported, the model can be of course created manually. To produce complete process model manually is inconvenient. Therefore, the user can create automatically an exemplification template of the process model. It includes templates for required BPEL elements, WSDL with BPEL extensions, and an activity model for the process flow. UML model can be modified

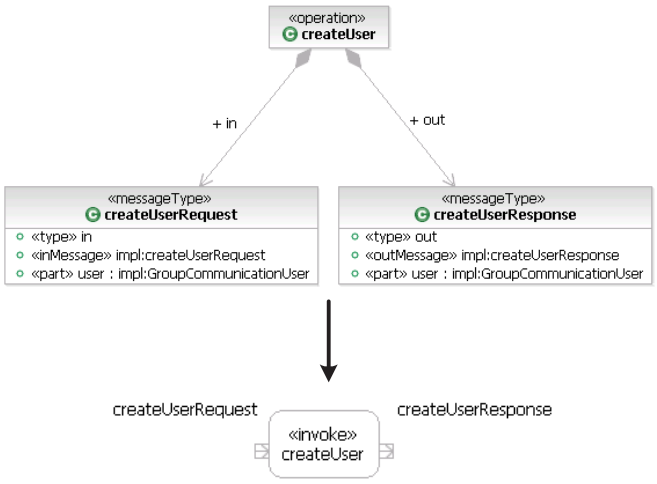

Figure 3. Generating invoke actions from WSDL operations

and extended to process model.

In case of *Group Messaging Process*, let us assume that there are no existing services, which are required to implement the process. In order to define configuration similar to one in Figure 2, we need to define the process definitions with required WSDL definitions. Furthermore, an complete implementation of the service must be defined as a process flow.

3 UML-based process model

UML profiles can be used for customizing UML to be used in a specific context. A UML profile defines domain specific concepts, stereotypes, and constraints. IBM's Draft UML 1.4 Profile for Automated Business Processes with a mapping to BPEL 1.0 [3] specifies how to create UML models that describe executable business processes and which can be mapped to BPEL4WS [4]. A profile defines a subset model elements to be used in the profile, and stereotypes, and constraints to be used with specific model elements. The constraints are defined informally as textual descriptions. In this paper we show, how UML Profile for Automated Business Processes [3] can be adopted to UML2 [13, 14] and BPEL-WS 1.1 [6], and implemented extending an existing UML2 case tool, Rational Software Modeller [17]. Clarity and simplicity of the UML models is taken as a leading factor, when defining the rules discussed in this paper. Simplicity of the models also affects the usability of the developed tools. Some chosen limitations, e.g. always generating default Remote Procedure Call (RPC) style SOAP over HTTP bindings, can be considered sufficient since they make the process modeling an easier task for the developer.

3.1 Concepts

The process definition is divided into three parts: a WSDL definitions class model (*«wsdl»*), a process definition class model (*«process»*), and a workflow activity model (*«activity»*). Example model hierarchy, a actual model instance, can be seen in Figure 4. Stereotypes in the model instances correspond to elements in the conceptual models. Conceptual models of each of these three parts are presented in the following sections.

Figure 4. Hierarchy of the mobile process model

3.1.1 Process definition

Process related data is placed in UML package, stereotyped as *process*. It includes *a process class*, *partner link*, and *partner* classes. Conceptual model of the process definition is presented in Figure 5.

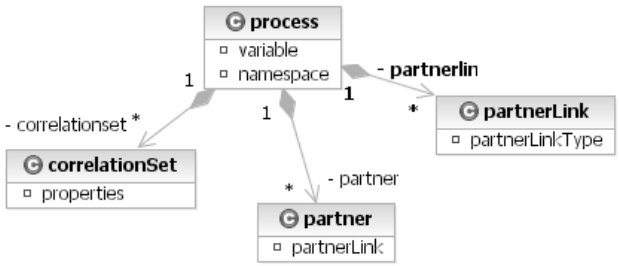

Figure 5. Process definition

A process class defines the process name, namespaces and variables. An example model, constructed according to the process rules, is shown in Figure 6.

3.1.2 WSDL definitions

A WSDL package includes port types their operations, messages and message types and parts. It defines also service

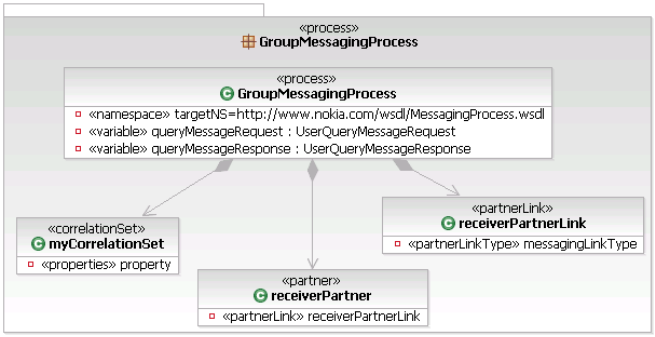

Figure 6. Process definition model

class and partner link types, target and SOAP namespaces. *Service class* defines the service name, location, binding and port name. A conceptual model of the WSDL definition is presented in Figure 7. As shown in the figure, WSDL model also includes BPEL specific extension of partner link types. A detailed structure of WSDL documents are not disscussed here. We aim to keep the WSDL definition as simple as possible and only to model the necessary elements to be able to generate WSDL and BPEL documents. Currently, we only support RPC-style binding mechanism and the SOAP protocol in use is HTTP. Since, the binding are not specified in the WSDL model and the default SOAP bindigs are always generated. A complete UML-based profile and validation support for WSDL descriptions have been proposed, e.g., in [11].

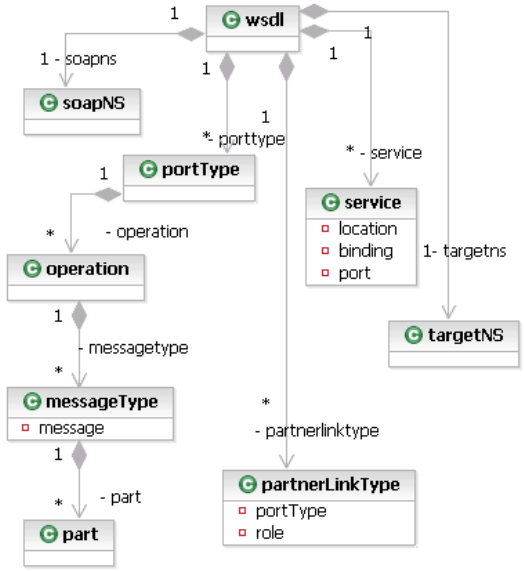

Figure 7. WSDL definition

Table 1. UML to BPEL mappings

Stereotype	UML element	Description and constraints	BPEL construct
invoke	Call behavior action	Input pin for parameters and output pin for return value.	Invoke action
while	Structured activity node	Attached UML constraint defines the while condition.	While action
pick	Structured activity node	UML note defines value for 'createInstance' attribute. Default value is 'yes'.	Pick action
onMessage	Call behavior action	Inside pick structured activity node. Input pin defines input variable.	onMessage action
	Decision node	Outgoing control flow name defines the case condition	Switch case structure
receive	Call behavior action	inputPin defines input variable value. UML note defines value for 'createInstance' attribute. Default value is 'yes'.	Receive action
	Initial node	Followed by receive or onMessage (except inside while loop)	Start of sequence
	Final activity node	End of while loop	End of while loop
reply	Call behavior action	Follows synchronous invoke action. Input pin defines action name. Output pin defines variable.	Reply action
	Join node	Combines two parallel paths. Unambiguous name must be given	Ends switch case
assign	Call behavior action	inputPin=from variable, outputPin=to variable syntax: 'variable:part:query'	Assign copy
	Final flow node	End of process flow.	Ends the process
	Control flow	Only one incoming and outgoing control flow for each action is allowed	Sequence
	Fork node	Concurrently performed branches	Flow

3.1.3 Workflow definitions

A mobile process flow is defined as a UML activity model. The model is stereotyped as *activity*. Table 1 defines a subset of UML2 activity elements with mappings to BPEL constructs, which is used in the approach. First column defines a stereotype of a UML element, if required. A short description as well as related constraints is also presented in the table.

In addition to constraints presented in Table 1, only one outgoing edge is allowed to leave from each basic and structured activity. Instead, flow branching is supported on through a *fork* node.

3.2 CASE-tool limitations and main conventions

We use Rational Software Modeller (RSM) [17] as a UML CASE-tool. It is based of draft version of UML2

metamodel and its native support for activity diagrams is rather limited. Due to the limitations, some alternative notations must be omitted. Main restrictions and conventions are explained in this chapter. A used metamodel fragment is presented in Figure 8.

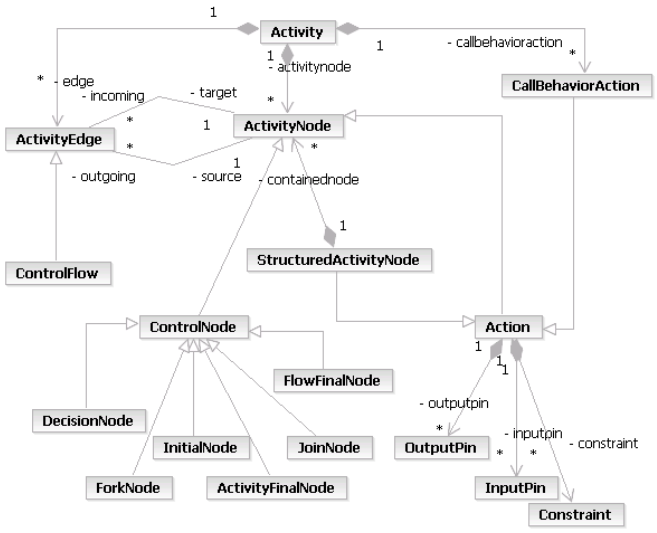

Figure 8. Process flow metamodel

Activity partitions are usually used to separate distinct ports. In RSM a structured activity cannot be involved with several partitions. Thus, partitions cannot be used. Operations are mapped to certain port by their name and port definition found from WSDL.

Only a limited number of *action* node types is supported. *Call behavior action* supports *input* and *output pins* and it is used to model all basic activities. *Input pin* is used to define input message and *output pin* is used to define out message. The stereotype defines the activity type. Supported basic activities are *invoke*, *receive*, *onMessage*, *reply*, and *assign*.

By default, a *control flow* between two nodes represents a *sequence*, a set of actions occurring in a certain order. *Switch case* control structure is represented by *decision* node, as shown in Figure 9. Each outgoing control flow defines a case condition. For example, when process receives a request from UI, if a *commandName* field of *UIProcessRequest* is 'invoke', *queryInbox* operation is invoked; if the command name value is 'outbox', *queryOutbox* is invoked. *A join* node ends the control structure and starts a new sequence.

RSM supports only one type of structured activity node. A stereotyped structured activity is used to model *while* and *pick* activities. *Pick* presents a nondeterministic choice and it may include several alternative branches. Each branch starts with *onMessage* activity. Example pick activity is presented in Figure 10. A UML note is used to define if a new instance of the process should be created.

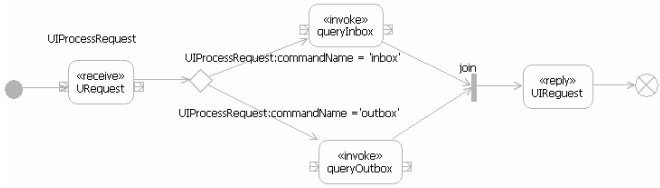

Figure 9. Decision construct

While activity starts from *initial* node and ends with *activity final* node. *While* activity is presented in Figure 10. The attached UML constraint defines the loop condition. While the condition evaluates true, the process loops. *Flow final* node (Figure 9) terminates the whole process.

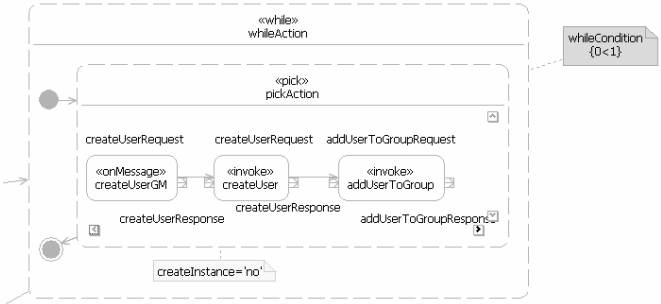

Figure 10. While activity

In BPEL specification, concurrency and synchronization between activities is provided by a *flow* construct. In these process defition rules, the *flow* is implemented as a *fork* activity.

4 Implementation and evaluation

Rational Software Modeller (RSM) [17] is IBM's UML2 case tool supporting software development cycle. RSM is based on Eclipse platform [7], which can be extended by plug-ins. We have extended RSM with MobileProcess plug-in to implement our approach (also compatible with Rational Software Architect [16]). Naturally, any other extensible UML2 editor could be used as well or a specific editor supporting the proposed rules and development steps could be build.

There are some limitations in the current prototype implementation. Error and exception handling is not supported by the current prototype. Error handlers can be implemented as attached stereotyped *call behavior actions*. Currently, there is no support for several interacting processes. Subprocesses could be defined as a type of self-contained structured activity. BPEL *links* are not currently supported. Support for the *flow* construct is also limitted.

It is due to missing *link* construct, which defines synchronization dependencies of sequences inside a *flow* construct. To support schemas and importing of namespaces, schema files should be into UML models and imported in the process model as well. Our future work including removing these limitation.

Future improvements could also include support for generating valid process models. It is possible to validate models against the defined UML profiles [11,18]. Similar UML-based validation mechanisms could be applied to BPEL models as well. Furthermore, to provide more powerful design support the correctness of the model should be ensured online. Also, a generative model development technique could be used to provide the user online guidance. E.g., the user can be provided updating task-list and customized wizards to create the structural composition of UML elements. A tool can instruct the user, which elements should be created, in which order, and to propose a repairing task if a conflict is found [8].

Our implementation is independent of any specific workflow engine. In order to execute generated BPEL code, some process engine specific code and build files are usually required.

5 Related work

In [19], Skogan *et al.* an approach to using UML 1.x to model Web service compositions is proposed. In the approach, UML models can be translated into executable specifications descriped in composition language, for example BPEL. This approach focuses purely on composing existing Web services and discusses a limited set of all possible workflow activities. The proposed UML profile uses heavily tagged values and requires quite amount of manual work. In our approach, a template model with default elements and activities can be generated automatically as well as BPEL specific WSDL extensions.

UML sequence diagrams have also been applied for modeling Web services choreography. A MDA fashion approach with automatic transformations to a Web Service composition language have been proposed by Bauer and Müller in [5]. The paper concludes that in addition to sequence diagrams, additional information concerning Web services is needed. The information can be WSDL definition specified with UML class diagrams. Importing WSDLs into UML models is an essential phase of our approach.

Business Process Modeling Notation (BPMN) [15] is OMG's standard for modeling business processes and can be used as a bridge to process implementation. BPMN is also targeted to providing graphical notation for XML-based process description languages, such as BPEL. We have taken typical development scenarios as a starting point and developed a prototype tool for supporting our purposes enabling designing of executable mobile processes. As most of software developers are familiar with Unified Modeling Language (UML) [14] and UML case tools, our process modeling tool is based on UML and an existing UML case tool.

IBM's ETTK toolkit [9] also generates BPEL and WSDL documents from workflow models given in UML. ETTK allows developers to design, develop and execute emerging Web services and autonomic technologies. In addition, ETTK toolkit enables developers to create and deploy Web services. Deployment of Web services and business processes varies among different containers and engines. In addition, it requires a server-specific configuration. Our goal is to generate executable BPEL code. Further, the proposed approach is independent of a workflow engine to be used. Mobile processes must provide mechanisms for interacting with mobile users, whereas autonomic computing aims at self-managed computing systems requiring minimal human interaction.

Related work on UML2 profile for WS-BPEL and CASE-tool consequent conventions are discussed in Ambühler's Master of Science thesis [2]. Required changes for updating UML 1.4 Profile for Automated Business Processes [3] to UML2 is proposed. No actual tool or approach is described. Rather limitations and proposed conventions, which must be omitted when implementing the UML2 profile in RSM [17].

6 Conclusions

Mobile messaging, especially SMS, is currently developing into the most important revenue for mobile phone operators. The mobile device will become the preferred method of accessing personalized services. While people are no longer tied to a fixed environment for communication with presence information, mobile business processes add a new value to original business processes.

In this paper, we have identified common development scenarios for mobile business processes. Especially, the process construction often involves composing existing Web services into composite service. This can be supported by importing existing service descriptions into UML model. In addition, invoke actions and process variables can be automatically generated from WSDL operations and messages. If process development is started without reusing existing WSDLs, templates for service WSDLs as well as workflow activity are provided. Thus, our approach also supports defining of service components, which implement the business process.

We have identified mappings from UML2 to BPEL and WSDL descriptions. We have also introduced our prototype implementations, which can support previously defined development scenarios and transformations from UML mod-

els to BPEL and WSDL descriptions. The approach is demonstrated by developing a group messaging process. It is a configurable mobile business process, which provides adaptive way of communication among a group of users with different devices. Our implementation is an extension for an existing integrated development environment with UML support. So, there is no need for separate process modeling tools and for learning a different notation for process modeling.

Our future and on-going research concentrates on better supporting of user interactions and user interfaces in different types of end-user devices, especially mobile devices. A support for dynamic user interface design is required as the changes in the process status is reflected in the UI [10].

Acknowledgments

This research has been carried at Nokia Research Center. The authors also want to thank Tarja Systä for her valuable comments and Suresh Chande for his support in the project.

References

[1] ActiveBPEL, http://www.activebpel.org/. *ActiveBPEL Engine*, 2006.

[2] T. Ambühler. *UML 2.0 Profile for WS-BPEL with Mapping to WS-BPEL*. Universität Stuttgart, 2005.

[3] J. Amsden, T. Gardner, C. Griffin, and S. Iyengar. Draft UML 1.4 profile for automated business processes with a mapping to BPEL 1.0. version 1.1, April 2004. http://www128.ibm.com/developerworks/rational/library/.

[4] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business process execution language for web services version 1.0, May 2003. http://www.ibm.com/developerworks/library/ws-bpel/.

[5] B. Bauer and J. P. Müller. MDA applied: From sequence diagrams to web service choreography. In *Lecture Notes in Computer Science: Web Engineering*, pages 132–136. Springer Berlin / Heidelberg, 2004.

[6] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. Business process execution language for web services version 1.1, July 2002. ftp://www6.software.ibm.com/software/developer/library/ws-bpel1.pdf.

[7] Eclipse, http://www.eclipse.org. *Eclipse WWW site*, 2005.

[8] I. Hammouda. A tool infrastructure for model-driven development using aspectual patterns. *Model-driven Software Development - Volume II of Research and Practice in Software Engineering*, pages 139–178, 2005.

[9] IBM, alphaWorks, http://www.alphaworks.ibm.com/. *Emerging Technologies Toolkit (ETTK)*, 2006.

[10] Y. Jaojin and L. Pajunen. Towards supporting user interface agility in developing heterogeneous device enabled business processes. In *Proc. of The 22nd Annual ACM Symposium on Applied Computing (SAC 2007)*. ACM, 2007.

[11] J. Jiang and T. Systa. UML-based modeling and validity checking of web service descriptions. In *Proc. of ICWS 2005*, July 2005.

[12] Object Management Group, Inc. *MDA guide*, 2003. Version 1.0.1.

[13] Object Management Group, Inc. *Unified Modeling Language: Superstructure version 2.0 Final Adopted Specification ptc/03-08-02*, August 2003. Version 2.0.

[14] Object Management Group, Inc. *Unified Modeling Language Specification*, May 2005. Version 2.0.

[15] Object Management Group, Inc., http://www.bpmn.org/Documents/OMG-02-01.pdf. *Business Process Modeling Notation (BPMN) Specification Final Adopted Specification 06-02-01.*, 2006.

[16] Rational Software, http://www-306.ibm.com/software/rational. *Rational Software Architect*, 2006.

[17] Rational Software, http://www-306.ibm.com/software/rational. *Rational Software Modeller*, 2006.

[18] P. Selonen and J. Xu. Validating UML models against architectural profiles. In *ESEC/FSE-11: Proc. of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 58–67, New York, NY, USA, 2003. ACM Press.

[19] D. Skogan, R. Grønmo, and I. Solheim. Service composition in UML. In *Proc. of Enterprise Distributed Object Computing (EDOC)*, pages 47–57. IEEE, 2004.

[20] W3C, http://www.w3.org/TR/wsdl. *Web Services Description Language (WSDL) 1.1*, 2001.

[21] Workflow Management Coalition, http://www.wfmc.org/standards/docs/. *Terminology & Glossary*, 1999.

[22] Workflow Management Coalition (WfMC), http://www.wfmc.org/standards/docs.htm. *XML Processing Description Language (XPDL)*, 2005.