Modeling Ubiquitous Business Process Driven Applications*

Victoria Torres, Pau Giner, Vicente Pelechano

Department of Information Systems and Computation
Technical University of Valencia
Camí de Vera s/n 46022
Valencia, Spain
{vtorres,pginer,pele}@dsic.upv.es

Abstract. This paper presents a model driven approach for the generation of business process driven web applications within the context of ubiquitous environments. For this reason, we have taken into account that the generated Web applications can be accessed via different devices (such as PDAs), which introduce a set of constraints that are better faced at the modeling process. This proposal relies on both model-to-model and model-to-text transformation techniques to obtain the final software artifact. As a result, we have implemented an Eclipse based prototype that generates from a business process definition the adequate GUI (in terms of the device) to execute the process as well as the equivalent definition into an executable language such as WS-BPEL.

1. Introduction

Business Process specifications play an important role in enterprise application development. These specifications allow defining the interaction between different participants in order to achieve certain agreed goals. However, enterprise applications cannot be limited to be run in the computers that are located in our offices (such as desktop environments). In fact, there exist scenarios where application users are constantly moving to different locations, needing applications that (1) help them performing the tasks that they are responsible of and (2) fit into their day life working devices. As a result, the development of this sort of applications could overcome spatial and technological obstacles that exist today. Then, we should take into account how our applications are going to be displayed in different devices which usually introduce both new characteristics and limitations.

Therefore, in this work we propose a methodological approach that allows us to obtain from a business process definition (represented in the BPMN notation [2]) the graphical user interfaces (GUIs) that provide support to the execution of the process for a wide range of output devices. In this proposal, device characteristics/limitations are handled at the modeling level but always following the separation of concerns promoted by the MDA approach, where the domain system and the technological

* This work has been developed with the support of MEC under the project DESTINO TIN2004-03534 and cofinanced by FEDER.

issues are specified by models at different levels of abstraction. Moreover, BPMN descriptions are also used to obtain the equivalent executable definition in terms of the WS-BPEL language [1], which will orchestrate during the execution of the process the set of services that made up the process.

The remainder of the paper is structured as follows. Section 2 presents briefly the proposed MDA approach and how it is completed to overcome output device limitations at the modeling level. Section 3 provides a general overview of the generation process by means of an implemented tool, where both model-to-model and model-to-text transformations are applied to automate the generation process. Finally, section 4 closes the paper with some conclusions and outlines further work.

2. A MDA Approach for Building Ubiquitous Business Process Applications

Roughly, the MDA approach proposes defining the software building process based on a set of models. Depending on the level of abstraction, these models are dependent or independent of technological issues. Following this approach, the OOWS approach [3] proposes to overcome the software generation process in two steps. In the first place, a technological independent description of the system is performed. In this step, the software designer concentrates on the domain problem (problem space) and how it should be represented using the models defined by the method. In a second stage, this technological independent representation is transformed in terms of a specific technology. In a previous work [4] we presented an extension to the OOWS approach for developing business process driven web applications. This extension endowed the approach at the modeling level by the introduction of the Business Process Model (BPM). This model is defined in terms of the functionality modeled both in the Structural Model (represented by UML Class Diagrams (CD)) and the Services Model (SM) and its purpose is to describe by means of a graphical notation a set of activities performed by different agents and sequenced by means of a control flow. The set of operations defined in the CD include the functionality that is provided within the boundaries of our system. On the other hand, the functionality that is "lent" from external partners is defined in the SM [5]. Moreover, in [4] we extended the navigational model with a set of new primitives that allowed the correct execution of the business processes defined in the system. The main idea of the proposal was to obtain automatically from a business process definition the corresponding graphical user interfaces (GUI) as well as the equivalent definition of the process in terms of an executable language. However, in this previous work we focused on generating these abstract GUIs for a specific target device (a desktop computer). We did not consider the possibility of having multiple devices to access the application.

Navigational and Presentation models (NM and PM hereafter) were introduced by Web Engineering methods in order to handle interaction aspects during the software modeling process. However, these two models are built independently of any technological aspects, and therefore they cannot be used to introduce specific aspects concerned about target devices [6].

For this reason, we have defined, at the PSM level, a mechanism that allows us to characterize those aspects introduced by output devices (such as space limitation) and

that need to be considered during the modeling process. This mechanism has been designed taking into account that some characteristics and limitations introduced by output devices are not always exclusive of one device and can be shared by several devices. In addition, these characteristics are not always orthogonal to each other, and some of them can be used to solve the limitations introduced by others.

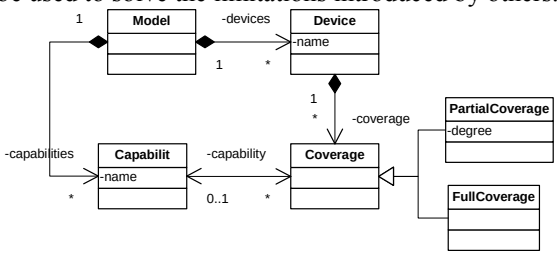

Fig. 1 PSM Metamodel for handling devices and their characteristics

Fig. 1 shows the metamodel that has been defined to handle different output devices at the PSM level. It defines (1) the set of output devices that are going to be supported by the Web application, (2) the set of capabilities/limitations introduced by these devices (space, voice, etc.), (3) which capabilities apply to which devices (because some devices can share some of them) and finally (4) the coverage that each capability is going to have over the contents defined in the Navigational model, which can be defined as *PartialCoverage* (it processes only part of the contents defined in the navigational model) or as *FullCoverage* (it processes all the contents defined in the navigational model).

3. An Eclipse Based Tool for the Generation of Ubiquitous Business Process Driven Web Applications

In order to materialize the presented approach, we have developed an Eclipse based tool as a proof of concept of these ideas. In particular, this tool allows us to obtain from a business process definition (1) the corresponding GUIs (specified by means of a navigational model) that are necessary to execute the process in any of the supported devices and (2) the process equivalent definition in terms of an executable language (WS-BPEL). Moreover, the Navigational Model previously generated is also transformed into the code that represents these GUI.

The definition and implementation of a set of transformations rules have empowered us to fully automate the generation process. The prototype tool has been developed within the context of the Eclipse environment. First of all, we have defined the metamodels involved in the proposal (CD, SM and NM) in EMF. The reason of having models defined in this format are that (1) we can manipulate them easily with the model instances editor provided by the framework and (2) it allows us to navigate models in order to transform them into either other models or directly into code. On the other hand, the set of transformations implemented in the prototype have been defined using two different languages. In the first place, to implement model-to-model transformations we have used the MOF 2.0 QVT Operational Mappings implementation provided by the Borland Together Architect 2006 for Eclipse. In the second place, model-to-text transformations have been implemented using the MOFScript

tool, which provides an implementation of the MOFScript language, an OMG candidate proposal for MOF2Text transformation.

4. Conclusions and Further Work

Ubiquitous Computing capabilities can improve considerably the way Business Processes are performed. Moreover, these improvements can be applied to a wide range of scenarios, from critical aircraft maintenance operations to supply chain management. However, little research has been done in taking advantage of conceptual modeling techniques for enabling an automatic software development of Business Processes defined within a ubiquitous environment.

In this work we have presented a MDA approach for the development of ubiquitous business process driven applications. This approach allows us to obtain automatically from a business process definition both the GUIs that represent the required user interaction to support the process execution for different kind of output devices and the executable definition of the process in the WS-BPEL language. Moreover, this proposal has been reinforced with the implementation of a prototype, which allows automating the generation process by the use of metamodels and the application of model transformation technologies.

As a result of this work, we have learned that model transformations techniques allows us to generate in a systematic way technological independent system specifications in terms of an specific technology. The real challenge arises in defining and handling properly the primitives that represent the system (its domain and its execution context) at the modeling level.

As further work we have planned to extend the proposal to cover the new challenges that introduce ambient intelligence and eAccesibility and eInclusion systems within a model driven approach and how these issues are translated into our tool.

5. References

1. Business Process Execution Language for Web Services version 1.1. Feb 2005.
2. Business Process Modeling Notation (BPMN) Version 1.0 - OMG Final Adopted Specification, February 6, 2006.
3. Pastor, O., Fons, J., Abrahao, S., Pelechano, V.:Conceptual Modelling of Web Applications: the OOWS approach. Web Engineering. In: Mendes, E., Mosley, N. (eds), Springer 2006, pp. 277-302
4. Torres, V., Pelechano, V.: Building Business Process Driven Web Applications. In: Dustdar, S., Fiadeiro, J.L., Sheth, A. (eds.): Business Process Management. 4th International Conference, BPM 2006. Lecture Notes in Computer Science, Vol. 4102, Springer Berlin / Heidelberg (2006) 322-337
5. Torres, V., Pelechano, V., Ruiz, M., Valderas, P.: A Model Driven Approach for the Integration of External Functionality in Web Applications. The Travel Agency System. MDWE 2005.
6. Vanderdonckt, J., Florins, M., Oger, F., Macq, B.: Model-Based Design of Mobile User Interfaces. In: Proceedings of Mobile HCI'2001: Third International Workshop on Human Computer Interaction with Mobile Devices, M.D. Dunlop and S.A. Brewster (eds.), Lille, 10 September 2001, pp. 135-140.