Web Application Development Focused on BP Specifications*

Victoria Torres
Dept. De Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46022 Valencia
vtorres@dsic.upv.es

Pau Giner
Dept. De Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46022 Valencia
pginer@dsic.upv.es

Vicente Pelechano
Dept. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46022 Valencia
pele@dsic.upv.es

Abstract

Business Process specification can be used during the software development process for different purposes. In this paper we present a Web Engineering approach that has been extended to allow the construction of Process Driven Web Applications. In this approach, BP specifications are defined at the PIM level and are used to produce (1) the proper navigational models that support the execution of these processes and (2) the executable definition that will allow automating the original Business Process.

1. Introduction

Business Process (BP) specifications play a very important role within the software development process. This kind of specifications allows defining organizational goals by means of the composition of different tasks, which can be performed by different roles within an organization and by external partners. By means of this notation it is possible to define at the modelling level the interaction between different systems. This approach is very useful since software systems are no longer conceived as isolated systems. Instead, their construction is based on the assembly of functionality that is provided by different partners. In this context, the Web service technology appears as an excellent technological candidate to make real this scenario.

Another aspect that plays up the importance of BP specifications is that these are better understood by domain experts. This fact ensures that system requirements can be unambiguously defined in a common notation that is shared between technical and domain experts.

Regarding the software development process, the Model Driven Development (MDD) approach has been proven satisfactory to overcome this goal. Proofs of this success are the MDA approach adopted by the OMG to promote the usage of models in software development and the standards (MOF QVT), model transformation languages (ATL, MOFScript, etc.) and tools (Eclipse Modeling Project, ATL Development Tools, oAW) developed to turn this approach into reality.

Within the Web Engineering area, the MDD approach is been applied successfully for the construction of Web applications. The set of different proposals developed in this area ([3][5][6][8][15][19]) endorses not only the benefits of applying such approach but also stress the importance of BP during the development process.

BP specifications are very important for organizations not just because they gather the organizational knowledge but also because they can be executed by process engines what improves the productivity of the organization. Business Process Management Systems (BPMS) are software systems built for the design, execution and monitoring of organizational BP. Moreover, these systems make possible improving BP since allow managers to analyze and change processes in response to data. However, these tools are more oriented to be used by managers. In some cases, we may want users to handle BP by means of the own corporate application, as if BPs were integrated into the application. It is in this case where our proposal makes sense.

We have joined research advances from two different areas, which are the Web Engineering area on the one hand and the Business Process area on the other. We have extended the OOWS

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

[12] Web Engineering method with enough expressivity to build, in a systematic way, Web applications that provide support to the execution of BP (considering BP as both short and long running processes).

The remainder of the paper is structured as follows. Section 2 introduces the possibilities that bring BP specifications within the software development process. Section 3 places BP specification within a Model Driven Development process. Section 4 presents a Web Engineering method that includes BP specifications during the development process and it follows a MDA approach. Finally, section 5 presents some conclusions.

2. Taking Advantage of BP Specifications

BP specifications define both organizational goals as well as the way to achieve them. These specifications represent a very valuable documentation within an organization. However, these specifications get more valuable when these change from passive to active. The way to make this change is by the use of a software system (usually an engine) that could give life to them.

In the OOWS method we have extended the architecture of the generated Web applications by the introduction of a process engine. This engine is in charge of making progress the launched process instances.

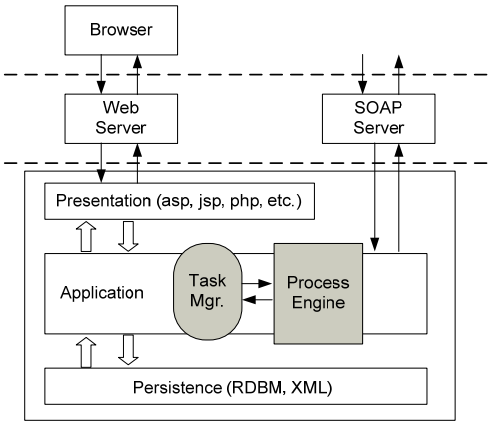

Figure 1. OOWS Web Application Architecture

As Figure 1 shows, the Application layer within the application architecture includes two grey-coloured elements, which are the *Process Engine* and the *Task Manager*. The former, as we have mentioned previously, is the one that invokes the tasks that conform the process definition in order to accomplish certain goals. The latter it was necessary to include for handling the asynchrony that introduces human tasks (tasks in which humans take part). It behaves as an intermediary element between the engine and the application and it has been implemented as a Web service.

Since we rely on Web Services as the mechanism to access functionality provided by external partners, we make use of a process engine compliance with WS-BPEL 2.0 [20] (a language for specifying business process behaviour based on Web Services).

3. Model Driven Development

Software Engineering methods propose to tackle the software development process from the domain space point of view. The main advantage of this approach is that it allows developers to concentrate on the particularities of the domain being developed. During this stage of the development process peculiarities regarding technological issues are not considered. In the same direction, the MDD approach promotes the use of models to achieve the software development process. In this approach models and model transformations become essential elements during the development stages. As a result of applying the MDD approach we get higher quality software solutions since (1) it allows developers concentrate on the domain being solved and (2) it allows performing the code generation step automatically, what avoids the existence of implementation errors.

There are different proposals to tackle the MDD approach. Among them we find MDA [9], which represents the initiative promoted by the OMG. This initiative proposes the construction of systems at different levels of abstraction. In particular it defines three levels which correspond to CIM (Computational Independent Model), PIM (Platform Independent Model) and PSM (Platform Specific Model).

Depending on the grade of detail in which BP specifications are defined we could place them at any of these three levels. For instance BP specifications can be defined during initial phases

of the software development process in order to state domain requirements (i.e. abstract processes modelled in the BPMN [4] notation). Then, these specifications can be refined specifying the system (i.e. private processes defined in BPMN). However, no details about target platforms are defined at this level. Finally, when BPs are defined in terms of the target platform (i.e. in an executable language such as WS-BPEL) we place BP specifications at the PSM level.

In the following subsections we present the way we have put into practice the MDD approach for the construction of BP driven Web Applications. This approach has been carried out in two steps. The former step consists in identifying the abstraction primitives that form the method metamodels. The latter consists in defining the transformation rules that move original models into other models or final code. Moreover, in these subsections we will see that there are already development tools that allow to put into practice the MDD approach.

3.1. Model Definition

Metamodels are used to gather the conceptual primitives that allow carrying out the modelling process for a specific class of problems. In order to achieve the separation of concerns promoted by the MDA, it is very important to distinguish the different aspects that characterize systems (its behaviour, structure, interaction with users, etc.). This distinction promotes the definition of different models that are in charge of representing a specific aspect of the system being defined.

BP languages such as UML Activity Diagrams, UML EDOC Business Processes, IDEF, ebXML BPSS or BPMN among others gather a set of primitives that allow representing requirements following a process-like way. Although BPMN was initially developed by the Business Process Management Initiative (BPMI) it is now being maintained by the Object Management Group. This fact promoted the standardization of this notation in contrast to other promising notations such as UML. In fact, UML defines a set of diagrams (which include the UML Activity diagram and Use Case diagram) that are used in UML for modelling business processes. However, in UML, objects are considered as first order citizens within the modelling process. As a result, this language is more suitable for software engineering than domain experts. Another aspect to consider is that UML lacks mathematical foundation. Then, to obtain executable UML business process models it is necessary to define the mappings from UML business process meta-models to an executable metamodel. On the contrary, BPMN provides a mapping to an execution language such as BPEL4WS, which utilizes the principles of formal mathematical models such as pi-calculus.

Another important aspect that determines the success of a specific language is the availability of tools that support them. It is very important to have tools that allow handling metamodels and models during the development process. For the modeling of BPs using the BPMN notation we find available different editors some of them developed using the Eclipse framework. Among these initiatives we find Borland Together [2] and the STP project [16]. The main benefit of the tools based on Eclipse is that models are defined in EMF, what facilitates manipulating them. Moreover, these tools are empowered with validation and BPEL generation support from BP diagrams.

3.2. Model Transformations

In order to get the most of BP specifications we propose to use them to derive not only the equivalent executable definition but also the *Navigational Model* that provides support for the execution of these BPs. There are different possibilities to achieve this task. Among them we find graph grammars [7], MOF 2.0 QVT [11] or XSL transformations [22]. From these alternatives we have chosen the QVT specification because it has been adopted by the OMG as the MDA standard to achieve model-to-model transformations. Although there are not many implementations of this standard and the existing ones are partially QVT-compliant (Borland Together [2], MTF [10], smartQVT [14], ATL [1]) we have found all the necessary expressivity to define and implement the necessary transformation rules for achieving the planned transformations. In a previous work [17] we have defined transformations using the Operational Mappings imperative language. This language allows us to define unidirectional transformations

between models, which are instances of MOF (Meta Object Facility) metamodels. We have used the Borland Together Architect 2006 for Eclipse that allows us to define transformations in this language and execute them over models defined by means of Ecore (a MOF Core implementation).

4. OOWS. A Web Engineering Method with support for BP

The OOWS (Object Oriented Web Solution) method [12] is the extension of the object-oriented software production method OO-Method [13] and it introduces the required expressivity to capture the navigational and presentational requirements of web applications.

In order to generate BP driven Web Applications, the OOWS method has been extended at the modelling level with the introduction of the Business Process Model (BPM) [19]. Figure 2 provides a graphical overview that includes the models involved in the proposal as well as the relationships defined between them. The purpose of the BPM 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. These activities invoke operations that have been modelled either in the *Structural Model* or in the *Services Model*. The existing relationship between the BPM and the *Structural* and *Services models* is depicted graphically by means of an arrow stereotyped with the <<uses>> keyword. The set of operations defined in the *Structural Model* 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 *Services Model*. This model was introduced in a previous work [18] in order to define the set of services (and operations provided by these services) that are supplied by external partners. The major benefit of having external functionality at the modeling level is twofold, (1) it allows us to handle external functionality as if it was part of our system and (2) it facilitates the integration between external functionality and other models defined in our method.

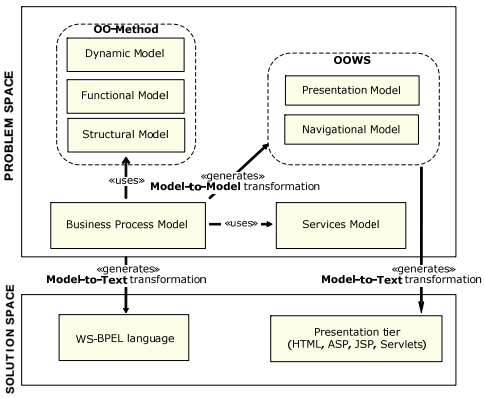

Figure 2. Method Overview

Some activities defined within the process definition require the existence of a user interface to be performed. These user interfaces are defined in the *Navigational Model* and allow the user to interact with the process by introducing some data, starting some activities or taking decisions over the system data. The relationship between the BPM and the *Navigational Model* is depicted in Figure 2 as an arrow stereotyped with the "<<generates>> Model-to-Model transformation" keyword. This means that from a BP definition we are going to obtain, after the application of a set of transformation rules, an initial version of the *Navigational Model* that will give support to the process execution. Then, once the *Navigational Model* is completely defined, again, by means of model-to-text transformations we can generate the equivalent graphical interfaces represented in a specific technology.

Finally, to automate BP definitions we have to translate them into an executable language. Then, this executable definition can be run by a process engine. This transformation is represented in Figure 2 with an arrow stereotyped as "<<generates>> Model-to-Text transformation". Once we have the equivalent executable description we can execute the process in any engine capable of executing process definitions created in WS-BPEL.

4.1. BP Definition

The BPM uses the BPMN notation to define process specifications. We opted for this notation because it provides mappings between the

graphics of the notation to the underlying constructs of an execution language such as WS-BPEL [21]. Although the BPMN notation is designed to cover a wide range of diagram types, we have used it for the design of detailed private business processes with interactions to one or more external entities. This decision allows that BP specifications could be transformed automatically into an executable process definition. Therefore, it is important to make this clear in order to obtain, after the application of the transformation rules the equivalent executable definition. In fact, as we have mentioned before, from a BPMN definition we want to obtain two different kinds of assets, one is the graphical interface that will allow the user to interact with the process, and the other one is the executable definition of the process.

When using the BPMN notation to model BP specifications we model each partner involved in the process in a different pool. As example Figure 3 depicts two different pools, one that defines the private process (left hand pool) and another that represents an external partner (right hand pool).

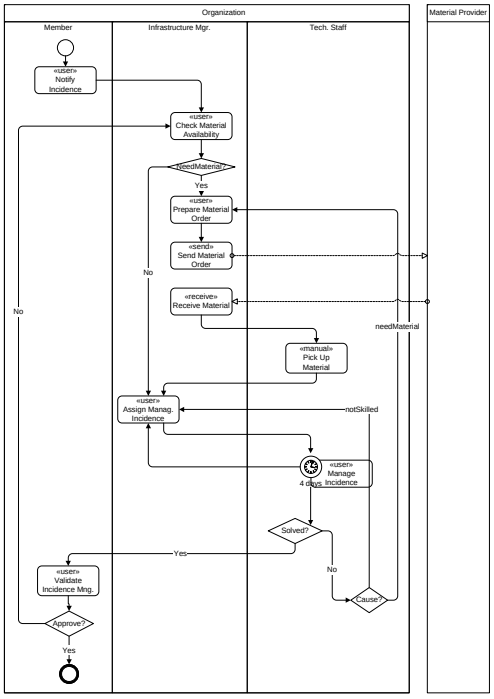

Figure 3. Business Process Specification Example

The private process is detailed defining the activities that form the process as well as the roles that are in charge of these activities. The BPMN notation allows classifying tasks in eight different types, which are *Service*, *Receive*, *Send*, *User*, *Script*, *Manual*, *Reference* and *None*). We make use of this classification to derive the *Navigational model* associated to the process. For instance, those activities that are defined either as *User* or *Manual* will produce new elements in the *Navigational Model*. These new elements will define the required navigation to support the execution of these activities.

The interaction that occurs during the execution of the process between different partners is defined by means of messages (dotted arrows depicted in Figure 3).

4.2. Functionality Definition

As it is depicted in Figure 2, the BPM uses functionality that is defined both in the *Structural Model* and the *Services Model*. On the one hand, the functionality defined in the *Structural Model* includes all the functionality that is provided within the boundaries of the system. On the other hand, the *Services Model* brings up to the modelling level the functionality that is provided by external partners.

Regarding the *Structural Model*, at the modelling stage, functionality is specified by means of its declaration and behaviour, which are defined in different models:

- The *declaration* of the functionality (operation signature) is defined in the Structural Model. In addition to declaring functionality, this model allows also specifying constrains and preconditions associated to each operation.
- The *behaviour* of the functionality is specified both in the Dynamic and Functional Models. The Dynamic Model allows defining the different valid object-life sequence for each class using State Transition Diagrams. It specifies the functionality (class operations) that allows moving class objects from one state to another. On the other hand, the Functional Model allows defining the effect that functionality (class operations) has over object properties.

As we rely on BPEL-compliance process engines, it is necessary that all the functionality that is invoked by the process was available as Web services. With the functionality that is gathered in the Services Model there is no problem. This functionality was original provided as Web services and then abstracted to the modelling level. However, the functionality declared in the Structural model is necessary to provide them as Web services. Therefore, during the transformation step where technological artifacts are generated we also produce Web services with this functionality.

5. Conclusions

Business Processes specifications play a very important role during the software development process. Considering the MDA approach, a BP specification can be used either at CIM, PIM or PSM levels depending on the detailed level in which these are specified. At CIM level BP specifications can be used to define system requirements. At PIM level, a more concrete but still platform independent representation can be used. Finally, at PSM level, BP specifications can be expressed in terms of an executable language.

Moreover, BP specification can be used along the development process for the generation of other artefacts. For example, BP specifications defined at PIM level can be used to produce the minimum required Navigational Model that allows the execution of the process.

The availability of tools let us put into practice the MDD approach for any kind of domain. In this case we have demonstrated that BP specifications can be used during the development process not only as a mechanism to integrate different systems but also to assist the generation of the Navigational Model.

Similarly to the appearance of Data Base Management Systems (DBMS) in the sixties for the management of data bases, now, there is a growing trend towards Business Process Management Systems (BPMS). These systems have emerged as key systems within enterprises to ensure their success in terms of productivity and costs. The Software Engineering community can take advantage of these systems in order to improve their solutions towards software systems based on any kind of process.

References

[1] Atlas Transformation Language (ATL). http://www.sciences.univ-nantes.fr/lina/atl/

[2] Borland Together. Retrieved from http://www.borland.com/us/products/together/index.html

[3] Brambilla, M., Ceri, S., Fraternali, P., & Manolescu, I. (2006). Process Modeling in Web Applications. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, issue 4. October 2006.

[4] Business Process Modeling Notation (BPMN) Specification. Retrieved from http://www.bpmn.org/

[5] de Castro, V., López, M., Marcos, E.: "Business Process Development based on Web Services: a Web Information System for Medical Image Management and Processing," *ICWS*, pp. 807-814, IEEE International Conference on Web Services (ICWS'06), 2006

[6] De Troyer, O., & Casteleyn, S. (2003). Modeling Complex Processes for Web Applications using WSDM. Paper presented at the Third International Workshop on Web-Oriented Software Technologies, Oviedo, Asturias.

[7] Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. 1999. Handbook of Graph Grammars and Computing by Graph Transformation. Vol 1. World Scientific

[8] Koch, N., Kraus, A., Cachero, C., Meliá, S.: Integration of Business Processes in Web Application Models. Journal of Web Engineering, vol. 3, no.1 pp. 022-049, May 2004.

[9] MDA Guide Version 1.0.1. Retrieved from http://www.omg.org/mda/specs.htm

[10] Model Transformation Framework (http://www.alphaworks.ibm.com/tech/mtf)

[11] Object Facility (MOF) 2.0 Query/View/Transformation Specification. Final Adopted Specification ptc/05-11-01.

[12] 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

[13] Pastor, O., Gomez, J., Insfran, E., Pelechano, V.: "The OO-Method Approach for

Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming". Information Systems, vol. 26, issue 7, pp 507–534, November 2001.

[14] SmartQVT. Retrieved from http://smartqvt.elibel.tm.fr/

[15] Schmid, H. A., Rossi, G.: Modeling and Designing Processes in E-Commerce Applications. IEEE Internet Computing, vol. 8, no. 1, pp. 19-27, January/February, 2004.

[16] STP BPMN (SOA Tools Platform Subproject). Retrieved from http://www.eclipse.org/stp/bpmn/index.php

[17] Torres, V, Pelechano, V., Giner, P.: "Generación de Aplicaciones Web basadas en Procesos de Negocio mediante Transformación de Modelos". In: Riquelme, J.C., Botella, P. (eds.): Ingeniería del Software y Bases de Datos. JISBD 2006. 443-452.

[18] 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. Sydney, Australia.

[19] Torres, V., Pelechano, V.: "Building Business Process Driven Web Applications". In S. Dustdar, J.L. Fiadeiro, A. Sheth (ed.). *Proceedings of the 4th International Conference on Business Process Management.* Lecture Notes in Computer Science, Springer. vol. 4102 (pp. 322-337). Vienna, Austria.

[20] Web Services Business Process Execution Language Version 2.0. Retrieved June, 2007, from http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

[21] White, S.A.: "Using BPMN to Model a BPEL Process". IBM. February 2005.

[22] XSL Transformations. Retrieved from http://www.w3.org/TR/xslt