Definition and Execution of Multiple Viewpoints in Workflow Processes

Dario Correal
University of Los Andes / LICEF-TELUQ-UQAM
Department of Systems and Computer Engineering
Cra 1E No 18A-10
Bogota D.C., Colombia / Montreal - Canada
{dcorreal@uniandes.edu.co}

Advisor: Rubby Casallas
University of Los Andes
Department of Systems and Computer Engineering
Cra 1E No 18A-10
Bogota D.C.,Colombia
{rcasalla@uniandes.edu.co}

1. Problem and Overview

Business Process Management aims at helping process-centered organizations in the design, execution, and monitoring of their processes. These tasks are crucial for organizations which would like to support and improve the business processes that identify their value chain by making them visible. Several efforts have been done to better understand, analyze, and organize business processes. Examples of these efforts are the process classification developed by the American Productivity and Quality Center (APQC) classification framework [2] and the Open Process Handbook Initiative (OPHI) [13].

The complexity of defining business processes arises because this task involves the description of activities, resources, products, tools, and control. Typically an organization has a huge amount of interrelated processes each one with a concern in mind hold by some stakeholder.

Nevertheless, the proliferation of process modeling languages and workflow engines capable to enact instances of these process models makes even more difficult the challenging task of modeling business processes. One mechanism to alleviate these problems is to raise the level of abstraction in the process definition and concentrate in the minimum set of elements relevant to the problem. As proposed in [4], the basic objective of a process model is to answer the following questions: what is to be done? Who does it? and what is produced?.

One strategy to face this complexity is to select a particular process, strategic for the company, and used it as a central axis around which different concerns are defined. Process designers need to cope with different difficulties at the moment of modifying processes with new concerns. Some of them are consequence of the absence of a mechanism to express, at a higher level of abstraction, process concerns in the language used by the stakeholders. Others are consequence of the increment of activities, and relationships appended to a target process, making difficult to be aware of the real impact of each modification in the overall process. Finally, the lack of a mechanism to define the interaction between the concerns and the target process leads to inconsistent states and deadlocks found, most of the times, at execution time.

We define a target process (TAP) as a particular business process selected for improvement inside an organization. In addition, we define a viewpoint as the perspective from which a particular process expert defines concerns in a TAP. In this scenario we consider two main actors: stakeholders and process designers. Stakeholders transmit to process designers, the concerns that need to be introduced in a process. Process designers interpret these concerns and introduce modifications in the TAP, keeping in mind the preservation of the coherence and correction of the processes.

The main goal of this research is to propose a mechanism to support process designers with the expression of concerns in workflow processes. We base our strategy on the use of viewpoints, model driven engineering (MDE) and aspect oriented (AOP) technologies [8]. Particularly, we developed a domain specific aspect language, called AspectViewpoint, to support the definition of viewpoints in workflow process models in a modular and non-intrusive way. In this way, process designers can analyze viewpoints to detect inconsistencies and dependencies between them, and then integrate them into the target process model using a model weaving engine. Finally, we transform the resulting woven process model into a particular workflow platform to execute it.

The results of this work can be applied to different domains. In this research, we have been validating our work in the domain of software development processes.

2. Background and Related Work

The Business Process Management Initiative now part of the OMG consortium has proposed the Business Process Modeling Notation (BPMN) [12] as a technology independent mechanism to model business processes

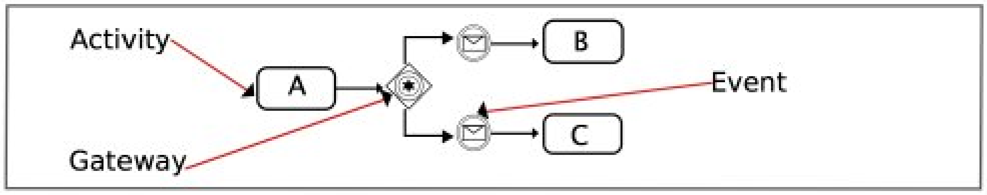


Figure 1. Definition of a Process Model using BPMN.

Figure 1 presents an example of a process model using some of the main elements offered by BPMN. Activities represent pieces of work defined in the process; gateways are used to define conditions in the execution of the process; there are five types of gateways: And, Or, Data Xor, Event Xor, and Complex. Finally, events represent actions occurred during the execution of the process, for example, the reception of a message or a timeout of an operation.

Given the technology independence of the BPMN models, several works have been proposed to transform these models into executable workflow processes such as BPEL[3] or XPDL[16]. We base our proposal on the use of Model Driven Engineering (MDE) domain. MDE proposes a framework to i) clearly define methodologies, ii) develop systems at any level of abstraction, and iii) organize and automate the testing and validation activities with the use always of models as first engineering artifacts [7]. In other words, the objective is to base all the software development life cycle on the models constructed for a particular system and guide the generation of platform specific code. In this sense, we use model transformations to achieve the independence of specific technologies.

Aspect Oriented Software Development aims at providing new ways of modularization, in order to separate crosscutting concerns from traditional units of decomposition, during software development [14]. To cope with the complexity of defining diverse but interrelated processes, we present process viewpoints as an aspect oriented modeling alternative. Aspect Oriented Modeling (AOM) is the intersection point between Aspect Oriented Software Development and Model Driven Engineering, using the separation of concerns with aspect oriented technologies at a modeling level. There are some works trying to use aspect technology in the domain of process. We present below two of the most relevant to our work.

In [6], the authors present AO4BPEL (Aspect Oriented extension for BPEL4WS), an extended process-oriented composition language with an aspect-oriented modularity mechanism, designed to express crosscutting concerns in business process definitions, using XPath as the pointcut language. In our approach, we have adopted BPMN [12] as a more general purpose process modeling language. We use aspects in BPMN process models that can be transformed into BPEL or any other workflow process language.

In [11], the authors propose SP-Aspects to support software processes, by weaving aspects into the development tools. SP-Aspects are platform independent, and they model a software methodology based on a proprietary ontology. As an example of SP-Aspect utilization, the authors present the adaptation of a software process to meet the UnitTest-first practice proposed in extreme programming (XP). In this example, the AspectJ language (http://www.eclipse.org/aspectj/) is used to weave aspects into the eclipse platform. In our work, we have adopted a different strategy. We do not try to weave aspects directly into the development environment to model the software process. Our strategy is based on the use aspects on process models which bring us platform and technology independence.

3. The AspectViewpoint Language

Model driven engineering (MDE) promotes the use of models as first-class entities during the development of software systems. Several benefits are obtained from this practice, specially the use of Domain Specific Languages (DSL) to ameliorate the communication between stakeholders and software developers. We have developed a domain specific aspect language called AspectViewpoint that facilitates us to define crosscutting concerns in process models in a modular and non-intrusive way. AspectViewpoint language offers constructors to define the points in the process affected by the concerns, and the actions to perform in those points to modify the target process model.

To introduce the main elements of the language, we use a simplified software process example consisting of three activities: design, codification, and unit testing. In our example, the quality assurance stakeholder intends to define a concern associated to the verification of the code artifacts. His process establishes that some developer reviews every code artifact, and then the whole team inspects it until no defects are found. This concern involves the modification of the target process by adding two new activities, review and inspection of the code, and also by changing the sequence of execution. Figure 2 presents the target process model and the concern.

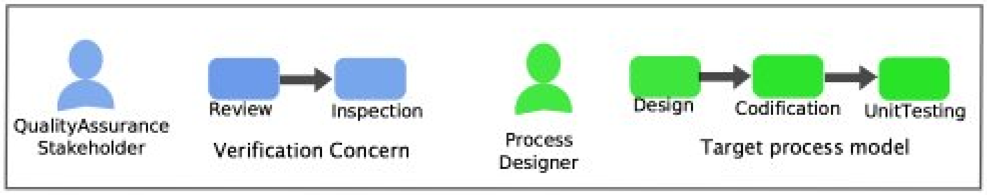


Figure 2. A target process model and the verification concern.

Figure 3 illustrates the expected resulting woven process. The following sections present the constructors of our language to express the points in the process to add the new activities and the way to change the target process.

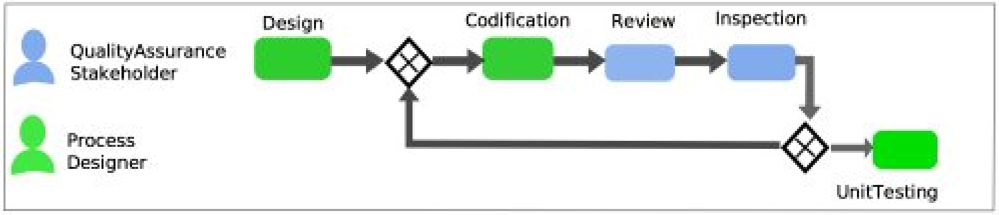


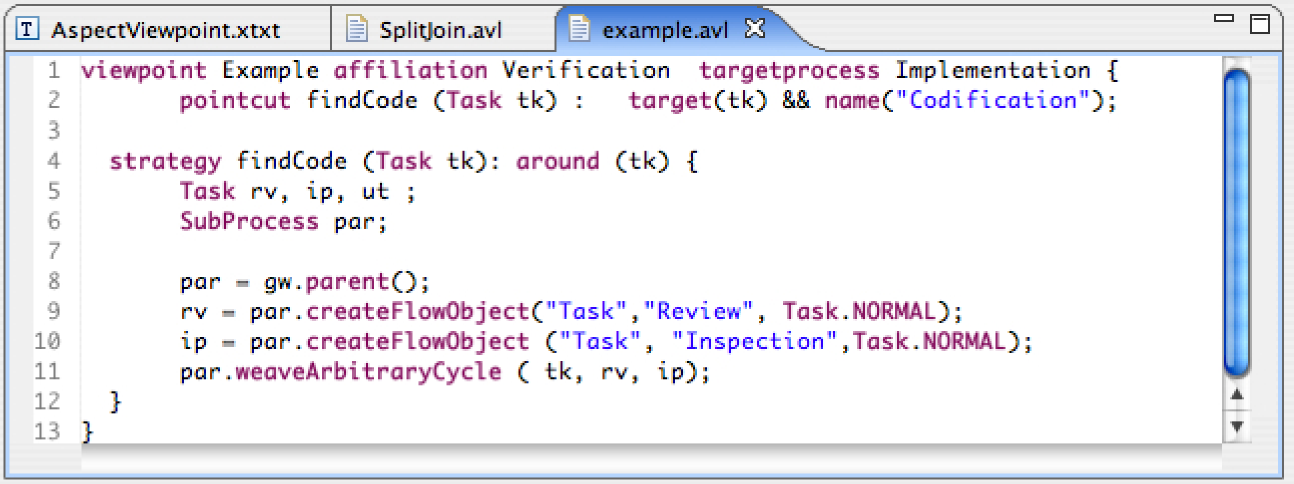
Figure 3. A target process model after the weaving of the verification concern

3. 1 Pointcut Definition

A joinpoint specifies where an aspect might insert adaptations [14]. In our case, we have defined a join point model based on the behavior of three of the BPMN elements: activities, gateways, and events. Pointcuts represent sets of joinpoints, selected using queries on the join point model [14]. In the code presented in listing 1, we define a pointcut (called findCode) in line 2. Its purpose is to select from the process model the codification activity. The pointcut has a parameter (Task tk) that denotes the actual activity during the execution and that can be used during the adaptation.

3.2 Advices and Workflow Control Patterns

Advices define actions to be performed once a pointcut is reached. These actions can be as simple as adding a new activity after or before a pointcut or as complex as changing the control flow of the process using new gateways to link the activities. To facilitate the task of defining advices, the AspectViewpoint language offers as a set of primitives based on the workflow control patterns [1] to express changes in the control flow such as parallel splits, parallel joins, synchronization points, and arbitrary cycles. The association between a pointcut and a set of actions is called in our language a strategy. The use of workflow patterns eases the understanding of the semantics of each strategy during the process execution, using a high level of abstraction and a vocabulary adopted by the workflow community.



```
 1 viewpoint Example affiliation Verification  targetprocess Implementation {
 2        pointcut findCode (Task tk) :   target(tk) && name("Codification");
 3 
 4   strategy findCode (Task tk): around (tk) {
 5        Task rv, ip, ut ;
 6        SubProcess par;
 7 
 8        par = gw.parent();
 9        rv = par.createFlowObject("Task","Review", Task.NORMAL);
10        ip = par.createFlowObject ("Task", "Inspection",Task.NORMAL);
11        par.weaveArbitraryCycle ( tk, rv, ip);
12   }
13 }
```

Listing 1. The Verification viewpoint using the AspectViewpoint Language

Listing 1 presents the AspectViewpoint code associated to the verification viewpoint explained in figure 3. In line 4, we define the strategy associated with the findCode pointcut, the around() method indicates that the workflow control will be modified before and after the selected joinpoint. In lines 9 and 10, we create the activities defined by the verification concern.

In line 11, we define an ArbitraryCycle workflow control pattern [1], using the method weaveArbitraryCycle (). This method introduces one DataXor gateway before the joinpoint and another one after the joinpoint. The first gateway is used to create a DataXor join pattern and the second one to weave a XorSplit pattern (See figure 3).

4. Implementation

We present the implementation of our strategy in three parts: the editor of our language, the weaving of the target process and the viewpoints, and, finally, the transformation from the woven process to a process language executable over a workflow engine.

4.1 Viewpoint Definition

Process designers describe the viewpoints expressed by stakeholders via the AspectViewpoint language. We have implemented a specific editor for our language using the OpenArchitectureware (oAW) framework (http://www.openarchitectureware.org). We used the Xtext language, part of oAW, to define an extended BNF grammar of our language. The oAW framework takes this grammar and generates an eclipse plugin-editor for the language. We enriched the editor with syntactic and semantic extensions developed using other oAW tools to allow the definition of semantic constrains checked when the process designer is creating a viewpoint. Furthermore, using this approach, viewpoints created with the editor are models conform to the AspectViewpoint metamodel produced also for the oAW framework based on the grammar.

Before weaving the target process with the viewpoints, we have defined a transitional transformation to an intermediate model conforms to a Process Weaving Engine (PWE) metamodel. This PWE metamodel enables us to represent at a high level of abstraction the relationships between the viewpoints being woven to detect potential conflicts. For instance, in our example, we used as joinpoint the codification activity to weave a cycle pattern. Now, suppose that other viewpoint selects the same joinpoint, the codification activity, to add a parallel split pattern (And gateway). In this scenario, we have a situation, which could lead into a potential conflict because there are two different gateways at the same point; the process designer has to decide what to do before the weaving takes place.

4.2 Weaving the Target Process and the Viewpoints

The target process model is defined using the BPMN notation and the viewpoints are expressed using our AspectViewpoint language. To perform the weaving, we use the Crosscutting-Specification Aspect Weaving (C-SAW) engine[5], which enables us to weave on a model aspects expressed in a language called Embedded Constraint Language (ECL)[10]. Therefore, to use C-SAW, we defined i) the BPMN metamodel using the Generic Modeling Environment (GME)[9] and ii) from the PWE intermediate model we produce the corresponding ECL code. The resulting weaving process generated by C-SAW is a new BPMN process model.

4.3 Executing the Woven Process

The final step in our strategy is the execution of the resulting woven process model. To accomplish this objective, we are experimenting with BPEL and specifically with Oracle BPEL, and with Cumbia [15], a general purpose workflow engine able to run executable models. We transform the BPMN process generated by the C-SAW engine into either the BPEL process model or the Cumbia process model. These transformations are again done using the GME on which we define the corresponding metamodels and the ECL code.

5. Results and Contribution

In order to evaluate our work we are conducting several case studies all of them in the context of software process. We have been working close with a software development team that has defined several CMMi(http://www.sei.cmu.edu/cmmi/) area processes. We redefined the processes related to verification, validation, quality metrics and generation of new releases using the AspectViewpoint language and then we woven these viewpoints with a target development process.

The phase of using and validating our approach, has been very important because it allowed us to fine tuning our language by testing its expression power and simplicity. So far, we have confirmed that the use of workflow patterns is, for the process designers, an accustomed way to model processes.

Aspect oriented modeling techniques have proved to be a feasible mechanism for the separation of concerns at a higher level of abstraction. In the context of process modeling, we can say that viewpoints help process designers to manage the complexity of the task by achieving this separation of concerns. However, from the beginning of our validation it was manifested the difficulty for the process designers of being aware of possible overlapping between viewpoints because each stakeholder focuses on his concern. For this reason, we developed the intermediate process weaving (PWE) metamodel to help users identify and solve conflicts. Furthermore, during the construction of the intermediate model, we can make explicit relationships, such as scattering, tangling, and crosscutting, among viewpoints that could help users to better understand the whole picture.

The model driven engineering strategy used in our approach has proved to be a supporting tool for process designers facing constant changes in the process definition. MDE has enabled us to be technology independent giving us the possibility of model transformation from notations such as BPMN to BPEL or to other languages.

Finally, we are aware that we need to go further to produce a complete executable business process. Until now, we have focused on the modeling of activities and control, but to execute a process we need to take into account other elements involved such as resources, products, and tools. We are working on this point, testing with some workflow engines and trying to understand the other elements involved to propose extensions to our AspectViewpoint language to deal with. Moreover, we are working on the generation of a traceability model that could be interpreted at execution time to give feedback to the stakeholder responsible of each particular viewpoint.

References

[1] Aalst, W. M. P. V. D.; Hofstede, A. H. M. T.; Kiepuszewski, B. & Barros, A. P. Workflow Patterns Distrib. Parallel Databases, 2003, 14, 5-51
[2]APQC home site. http://www.apqc.org/
[3]BEA Systems, M.C.S.A.S.S. Business Process Execution Language for Web Services Version. Techreport (BEA2003) 1.1 2003
[4] Breton, E. & Bezivin, J. An Overview of Industrial Process Meta-Models In: 13th International Conference Software & System Engineering and their Applications. -, Paris, France., 2000
[5] C-SAW. http://www.gray-area.org/research/c-saw/last visited on 18/05/2006
[6] Charfi, A,Mezini, M. Aspect Oriented Web Service Composition with AO4BPEL. ECOWS, Springer, 2004, 3250, 168-182 [7] Fondement, S. R. Defining Model Driven Engineering Processes In Proceedings of the 3rd Workshop in Software Model Engineering. WiSME, 2004
[8]G.Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming.In M. Aksit and S.Matsuoka, editors, Proceedings European Conference on Object-Oriented Programming, volume 1241, pages 220-242. Springer-Verlag, Berlin, Heidelberg, and New York, 1997.
[9]A.Ledeczi. The generic modeling environment. In WISP'2001, Budapest,Hungary. IEEE, May 2001.
[10] Lin, Y.; Gray, J.; Zhang, J. The Embedded Constraint Language: A Transformation Language for Visual Models,2006
[11] Mishali, O. and Katz, S. 2006. Using aspects to support the software process: XP over Eclipse. In Proceedings of the 5th international Conference on Aspect-Oriented Software Development (Bonn, Germany, March 20 - 24, 2006). AOSD '06. ACM Press, New York, NY, 169-179.
[12]OMG. Business Process Modeling Notation (BPMN) Specification Object Management Group, 2006
[13] Open Process Handbook home site. http://ccs.mit.edu/ophi/main.htm
[14]Schauerhuber Andrea, K. E. R. W. W. M. Towards a Common Reference Architecture for Aspect-Oriented Modeling 8th International Workshop on Aspect-Oriented Modeling, in conjunction with AOSD'06, Bonn,Germany, March 21., 2006
[15] Villalobos, J. Cumbia-XPM: Model Specification Draft Version 0.1 University of Los Andes, 2006
[16]Workflow Management Coalition. Workflow Process Definition Interface -- XML Process Definition Language Version 1.0 Workflow Management Coalition, 2002