Weaving Business Requirements into Model Transformations

Ying Zou[1], Hua Xiao[2], Brian Chan[1]

Department of Electrical and Computer Engineering[1]
Queen's University
Kingston, Ontario, Canada
ying.zou@queensu.ca, 2byc@qlink.queensu.ca

School of Computing[2]
Queen's University
Kingston, Ontario, Canada
huaxiao@cs.queensu.ca

Abstract. Model driven development (MDD) is regarded as a promising software development technique which can reduce the complexity and cost of developing large software systems. In recent years research in MDD has focused on the technical domain where techniques and tools are developed to assist in automatically transforming design models to implementation models. However, little attention has been paid on automatically transforming knowledge embedded in business requirement models (e.g., business processes) into generic design models and implementation models. In this paper, we use aspect-oriented modeling (AOM) techniques to help us customize the primary model (e.g., design model and implementation model) by weaving different business requirements into it. As a result, we can verify the primary models against the business requirements. Our case study demonstrates the feasibility of our proposed approach.

Keywords: Aspect-oriented modeling, Model transformation, Model annotation, Business process, BPEL

1. Introduction

Model driven development (MDD) is regarded as a promising software development technique which can reduce the complexity and cost of developing large software systems. Developers express the design of a system using design models, such as UML diagrams. These models are then converted to source code (i.e., an implementation model). This approach holds a great deal of promise since models could be analyzed and simulated ahead of time in order to explore design alternatives and uncover limitations. The automatic generation of implementation models offers vast improvements in productivity. In recent years research in MDD has focused on the technical domain where techniques and tools were developed to assist in automatically transforming design models to implementation models. However, little attention has been paid on automatically transforming knowledge embedded in business requirement models (e.g., business process models) into generic design models and implementation models. Whereas design and implementation models are the concern of technical stakeholders in a project, business requirement models are the concern of the business stakeholders. Due to the gap between business and technical domains, it is challenging to effectively integrate business requirements into the technical domain and influence the evolution of the design model and the implementation model.

Thanks to recent product offerings, such as IBM's WebSphere Business Modeller (WBM) [1], business analysts can now create business process models and perform simulations on these models. A business analyst could describe the different tasks and conditions which an organization must follow in order to deliver a particular business service or process. For instance, a "book purchase" business process may record that once a customer picks a book; the inventory must be checked since the book may be out of stock. Also the process may indicate different shipment methods or supplier ordering techniques based on various business rules. Through the business process simulation the analyst may compare the cost and benefits of modifying a particular process. For the "book purchase" example, a company may decide to reduce the "fulfillment time for a book order" by shipping out of stock books directly from the supplier instead of having the supplier ship the books to the company's warehouse. To generate enterprise applications, the business process models are transformed into Business Process Execution Language for Web Services (BPEL4WS) [2] which is an execution language to compose web services into enterprise applications. However, the attributes specified by the business analysts are not carried down to the BPEL during the transformation processes. Those attributes reflect the business requirements from the perspectives of business analysts and are important to validate if the composed web services using BPEL models conform to the business requirements. Therefore, it is important to develop a systematic approach to enhance the BPEL models to properly reflect business requirements specified in the business process models.

In our work, we use aspect-oriented modeling (AOM) techniques to help us customize the primary model (i.e., BPEL models). AOM involves weaving different business requirements into the primary models. Multiple stakeholders (e.g., business analysts and BPEL developers) have their own views or concerns of the models. For example, the business analysts specify business requirements (e.g., cost, time and revenue) and annotate them in the business process models. These requirements are used in the business process simulation to verify if the business process model can achieve the business objectives. BPEL developers are interested in selecting the proper web service that can meet the business requirements. To facilitate the validation of business requirements in the BPEL models, we propose a framework for integrating multiple aspects (i.e., business requirements, such as cost and time) into the primary models using AOM techniques. We express business constraints as aspect models and develop techniques to weave aspect models into the primary models (e.g., BPEL models). We aim to generate the customized applications that meet the business requirements. In particular, we maintain the primary models (i.e., BPEL models) unchanged, and annotate the primary models using separate aspects. As a result, we can verify if the primary model can achieve the business requirements using the annotated model. To examine the feasibility of our approach, we enhance the BPEL model and annotate the business requirements that the original BPEL model cannot describe. This can assist in evaluating the achievement of business requirements in enterprise applications generated from BPEL models.

The rest of the paper is organized as follows: Section 2 provides the overview of business processes and BPEL models. Section 3 presents a framework for integrating multiple aspects into model transformations. Section 4 discusses a case study using business requirements to annotate BPEL models. Section 5 introduces the related work. Finally, Section 6 concludes the paper and discusses the future work.

2. Overview of Business Processes and BPEL

Business processes are designed to describe daily routine for a company or corporation so that interested parties can understand and possibly improve on it. More specifically, business processes are a set of logically related tasks that are performed to achieve business objectives. A process definition specifies tasks, connections, roles and resources. Tasks define the operations for achieving the business objectives. Connections include the control flows and data flows between tasks. A role performs a set of designated tasks. Resources are necessities for executing tasks. An example of a business process could be a "purchase order processing", as shown in Figure 1. The "purchase order processing" contains five tasks, such as "receive purchase order", "select a shipper", and "schedule a production". The data, "purchase order", follow from "receive purchase order" task to "select a shipper" task. A business process can be simulated using a simulation engine. A business process is modeled by a business analyst who annotates simulation attributes into the process definition to evaluate if the business process can achieve the desired business objectives. Simulation attributes (e.g., the processing time to calculate the final price, the cost to schedule production) reflect the criteria for assessing the performance of a business process.

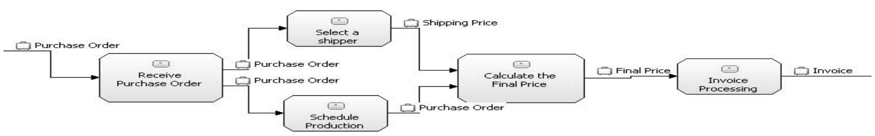

Fig. 1 "Purchase Order Processing" Business Process

BPEL models can be automatically generated from process definitions. A BPEL model usually includes *partner links*, *variables*, and *activities*. A *partner link* represents a conversational relationship between two partner processes. A *variable* provides a means for the process to maintain data and process history based on messages exchanged. An *activity* defines the operations between different partners in the process. *Activities* are composed by basic activities and structured activities. Basic activities are *receive*, *reply*, *invoke*, *assign*, and *wait*. The structured activities are *sequence*, *switch*, *while*, *pick* and *flow*. The "purchase order" data, shown in Figure 1, is a *variable* that flows among tasks, such as "receive purchase order", "schedule a production" and "calculate the final price".

BPEL is used to compose services to implement enterprise applications. The requirements, such as cost and benefit, are interested to the business analysts. However, the BPEL standard does not provide specification for representing requirements. Therefore, BPEL developers do not have reliable criteria to verify business requirements in the composed applications before the deployment.

3. A Framework for Integrating Multiple Aspects into Model Transformations

Conceptually, we model the interests of multiple role players as aspect models that are independent from the primary model (e.g., BPEL models). More specifically, the aspect models describe business requirements that crosscut the primary model. To weave an aspect model into a primary model, a pointcut specifies the locations and contexts that an aspect model is to be woven into the primary model. A weaver produces a composed model by integrating the primary model and aspect models, as shown in Figure 2. A sample application scenario of the approach is illustrated in Figure 3. For example, business analysts may hope to annotate attributes such as cost of a task and key performance indicators (KPIs) to the primary model (i.e., BPEL model). For example, time is an important factor in business process. If a business process requires long time to complete, the organization may lose its customers. Developers may be interested in the attributes, such as, execution time of a task and the frequency of executing a task. Specifically, the execution time of a task reflects the efficiency of the implementation. The frequency of executing a task can help developer to optimize the implementation. We represent each attribute separately as an aspect model. Based on the selection of the aspects from different roles (e.g., business analysts and developers), we can generate multiple composed models for different purposes. For example, the BPEL simulator can evaluate whether the primary model satisfies the requirements specified by the business analysts. In the following sub-sections, we discuss the aspects, point cuts and weavers in more details.

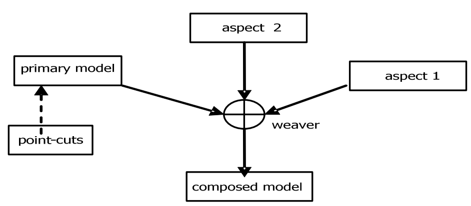

Fig. 2 Main Components of Aspect-Oriented Modeling

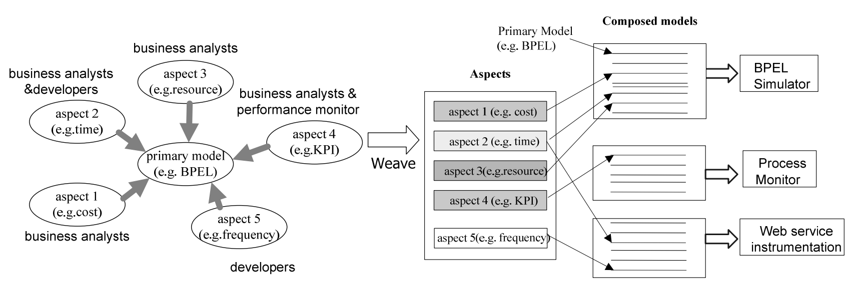

Fig. 3 Enhancing Primary Models with Multiple Aspects by Weaving

3.1 Aspect Models

An aspect model captures common characteristics of the similar attributes. To weave an attribute into a primary model, an aspect model can be instantiated to include the specific value of an attribute in an aspect model. An instantiated aspect model is called a *context-specific aspect model* [3, 4]. An aspect model can be instantiated multiple times to produce multiple context-specific aspect models. For example, if we define a *Cost and Revenue* aspect model for a task, as shown in Figure 4, it contains attributes, including revenue, processing cost, startup cost, wait time cost, and the currency for each cost. We instantiate the *context-specific* aspect by assigning the actual values to the attributes defined in *Cost and Revenue* aspect model.

Cost and Revenue Aspect
revenueCurrency : string revenueValue : double processingCostCurrency : string processingCostValue : double startupCostCurrency : string startupCostValue : double waitTimeCostCurrency : time waitTimeCostValue : time

Fig. 4 An Example of Aspect Model

To provide the specific values to instantiate aspect models, we define a context-value table, which specifies the relations between a context and a *context-specific* aspect model. A context contains the identification information for the join point where the *context-specific* aspect model can be inserted into the primary model.

In the context-value table, we use the context to find the relevant values for the attributes in the aspect models. For the example of the "calculate the final price" task in a business process, we suppose the context is the name of a task. The context-value table can be defined as shown in Table 1.

Table 1. An Example of Context-Value Table

Task Name	Revenue	Processing Cost	Startup Cost	Wait Time cost
Select shipper	USD, 10	USD, 0	USD, 0	USD, 0
Calculate the final Price	USD, 16	USD, 5	USD, 0	USD, 2
Produce schedule	USD, 20	USD, 7	USD, 0	USD, 0
Process invoice	USD 12	USD, 4	USD, 1	USD, 0

Once we obtain the specific values (shown in Table 1) to the attributes in the aspect model, we can instantiate a context-specific aspect model for "*cost and revenue"*. We use an XML schema to represent the *context-specific* aspect model since BPEL is defined using XML schema. It is shown as follows:

```
<revenue currency="USD "  literalValue=" 16" >
<processingCost currency=" USD" literalValue="5" >
<startupCost currency=" USD" literalValue="0" >
<waitTimeCost currency="USD" literalValue="2" >
```

PointcutsPointcut specify join points where the aspects are woven into the primary model. A pointcut can be defined using the regular expression shown in Figure 5.

$$\mathit{Scope}\underbrace{(\ \mathit{name}[\&\ \ \mathit{name}\ldots])}_{\text{One or more}}\underbrace{[|(\ \mathit{name}[\&\ \ \mathit{name}\ \ldots]\ )]}_{\text{Zero or more}}$$

- Scope specifies the location of a point-cut.
- Names specify the required XML attributes names. It can include one or more name with "& (and)" or" | (or)".

Fig. 5 Expression for Pointcuts

For instance as shown below, a "receive" element in a BPEL model represents an activity waiting for a message. The message is described as "variable" attribute associated with "receive" element. And an "invoke" element in a BPEL model allows the BPEL model to invoke an operations. The inputs and outputs of the operation are described using "inputVariable" and "outputVariable" attributes associated with the "invoke" element.

```
   <receive … variable="InputCriterionVariable" …>
   <invoke  inputVariable="InputCriterionVariable" …
outputVariable="TakeBloodSampleOutputCriterionVariable"…>
```

For example, we can define a pointcut by using the following expression:

$$*.(\text{“}\mathit{variable}\text{”})|(\text{“}\mathit{inputVariable}\text{”})|(\text{“}\mathit{outputVariable}\text{”}) \tag{1}$$

In this expression, "*" means the aspect model can be woven in any place of a BPEL model. The expression after "." means that the pointcut should be located in an XML element that has the attributes, namely, "variable", or "inputVariable", or "outputVariable".

Weaver

Composing an aspect model with a primary model involves: (1) identifying pointcuts in the primary model; (2) instantiating the aspect model by using the context-value table to produce a *context-specific* aspect model; and (3) integrating the context-specific aspect model with the primary model. The overall framework is depicted in Figure 6. The outcome of the framework is the composed models. To identify pointcuts in the primary model, the weaver compares the attributes of each element in the primary model with the expressions of the pointcuts . If an element matches the expressions, the weaver inserts a context-specific aspect model into the primary model.

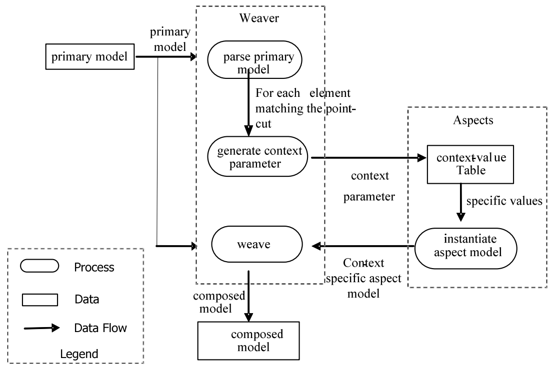

Fig. 6 A Framework of Integrating Multiple Aspect Models into a Primary Model

4. A Case Study

To evaluate the feasibility of our proposed framework, we apply our techniques in an industrial collaborative project. In our project, the simulation attributes are specified by business analysts using a business process modeling toolkit (i.e., IBM WebSphere Business Modeler (WBM)). A business process simulator uses the simulation attributes to validate the business processes in the toolkit. To generate an enterprise application that implements the business processes, the business processes are automatically exported to BPEL models. However, the simulation attributes are lost in the transformations since BPEL cannot represent those attributes. Therefore, the business requirements in an enterprise application can be only evaluated after the application is deployment in the run-time environment. It is very expensive and time-consuming to modify the application, if the business requirements cannot be met in the deployment environment [5]. A lightweight BPEL simulator is required by our industrial collaborators to evaluate the BPEL models before the actual deployment. In our project, we decide to reuse the existing business process simulator in WBM to simulate the exported BPEL processes.

We apply the proposed framework in this project following the steps:

(1) A prototype extractor is developed to extract simulation attributes from business process definitions, and transforms the simulation attributes in business processes to aspect models.
(2) We treat BPEL models as the primary model
(3) A prototype weaver is developed to integrate multiple aspect models into the primary model
(4) A prototype converter is developed to import the composed model to the existing simulation engine.
(5) Simulate BPEL models using the existing simulation engine.

By analyzing the simulation attributes in the process definitions, we identify 14 aspect models for different types of simulation attributes. Each aspect model has its own pointcuts. The weaver annotates the revenue, processing cost, startup cost, wait time cost as the simulation attributes to the corresponding elements in BPEL models using the pointcut expressions. We convert the "purchase order processing" (showed in Figure 1) process definition into the BPEL model. Figure 7 shows the result for simulating the "purchase order processing" BPEL model. The screenshot depicts the revenue, cost and profit for each task, as well as the total cost for this process.

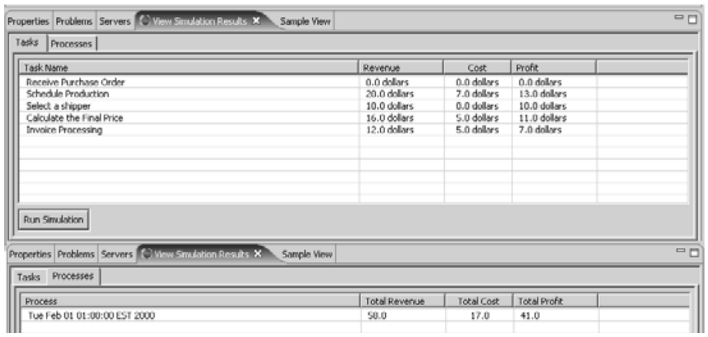

Fig. 7 Screenshot for the BPEL Simulation Result in a BPEL Modeling Environment

Using the BPEL simulator, developers can evaluate the business requirements during the development of the enterprise applications. Furthermore, the simulation results can be used to optimize business processes and assist service composition.

5. Related Work

Research in Early Aspects focuses on managing aspects in the requirements engineering and the architecture design [6]. To detect cross-cutting concerns in requirements, Duan and Cleland-Huang [7] provide a probabilistic model to compute the similarities between different requirements and use a hierarchical algorithm to cluster similar requirements. Siy *et al.* [8] introduce an aspect-oriented requirements specification system. They describe non-functional concerns as advices for transforming parameterized requirements to product-specific requirements. Marta *et al.* [9] give an approach for tracing semantics to identify the tracing links between the source and target elements. However, most of the research in Early Aspects does not transform the requirements to the implementation models. Model weaving is a technique to handle the relations between different models. The primary objective of the model weaving is to associate the fine-grained relationships among the elements of distinct models, and establish the linkages between models. Czarnecki and Helsen [10] compare different model transformation approaches and categorize a number of existing approaches. Del Fabro *et al.* [11] present an approach for handling mappings between models. Del Fabro and Jouault [12] use model weaving to improve efficiency in the creation and maintenance of model transformation programs. However, these approaches do not consider the information losses models during model transformations or weaving. Our work aims to capture the business requirements defined in a former model and ensures that the business requirements can be carried through the model transformations.

6. Conclusion and Future Work

In the process of MDD, it is critical to transform business requirements from the business processes into design models and implementation models. Due to the limitation of representing design models and implementation models, the business requirements are often lost during the model transformations. In this paper, we present a framework to enhance models with multiple aspects from business requirements. This framework keeps the primary model unchanged, represents the concerns separately, and generates composed models. In this context, we can integrate multiple aspects of different requirements into model transformations. As a result, multiple composed models can be generated with respect to a particular interest from a stakeholder. A case study in an industrial setting demonstrates the feasibility of our framework. In the future, we plan to enhance the expressiveness of the pointcuts and examine complex BPEL models.

Acknowledgement

This research is sponsored by IBM Canada, Centers for Advanced Studies (CAS), and National Sciences and Engineering Council (NSERC). We would like to thank Mr. Jay Benayon, Mr. Bill O'Farell, Ms. Jen Hawkins and Ms. Elena Litani at the IBM Canada Toronto Laboratory for their suggestions on this work.

References

1. IBM WebSphere Business Modeler, http://www-306.ibm.com/software/ integration/wbimodeler/.
2. Business Process Execution Language for Web Services version 1.1, http://www.ibm.com/developerworks/library/specification/ws-bpel/.
3. Straw, G., Georg, G., Song, E., Ghosh, S., France, R.B., Bieman, J.M.: Model Composition Directives. In: Proceedings of the 7th International Conference Unified Modelling Language: Modelling Languages and Applications, pp.84-97. Lisbon, Portugal.
4. France, R., Ray, I., Georg, G., Ghosh, S.: Aspect-Oriented Approach to Early Design Modelling. In: IEEE Proceedings-Software, 2004. 151(4): pp. 173--185.
5. Bennett Jonathan, D., Hemani Malik, S., O'Farrell William, G.: Business Process Execution Language Program Simulation, United States Patent. 200601190926, http://www.freepatentsonline.com/20060190926.html.
6. Resende, A. M. P. de, Silveira, F. F., and Cunha, A. M. da: Early Aspects: Some Analysis, Trends and perspectives. In: Workshop on Early Aspects, OOPSLA 2005, October 16, San Diego, California. USA.
7. Duan, C., Cleland-Huang, J.: A Clustering Technique for Early Detection of Dominant and Recessive Cross-Cutting Concerns. In: Workshops in Aspcet-Oriented Requirements Engineering and Architecture Design ICSE 2007, May 20-26, 2007, Minneapolis, MN, USA.

8. Siy, H., Prasanna, A., Victor, Winter, Mansour, Zand: Aspectual Support for Specifying Requirements in Software Product Lines. . In: Workshops in Aspcet-Oriented Requirements Engineering and Architecture Design ICSE 2007, May 20-26, 2007, Minneapolis, MN, USA.
9. Marta, S. T., Ana, M., Raquel, A., Fernando, A., Joao A.: A Traceability Method for Crosscutting Concerns with Transformation Rules. In: Workshops in Aspcet-Oriented Requirements Engineering and Architecture Design ICSE 2007, May 20-26, 2007, Minneapolis, MN, USA.
10. Czarnecki, K., Helsen, S.: Classification of model transformation approaches. In: OOPSLA2003 Workshop on Generative Techniques in the Context of MDA, Anaheim, CA, USA, 2003.
11. Didonet Del Fabro, M., Bezivin, J., Jouault, F., Breton, E., Gueltas, G.: AMW: A Generic Model Weaver. In: Premieres Journees sur I'Intgenierie Dirigee par les Modeles, 2005.
12. Didonet Del Fabro, M., Jouault, F.: Model Transformation and Weaving in the AMMA Platform. In: Pre-proceedings of the Generative and Transformational Techniques in Software Engineering (GTTSE 05), Workshop. Centro de Ciências e Tecnologias de Computação, Departemento de Informatica, Universidade do Minho, Braga, Portugal, pages 71--77.