A Novel Design Framework for Business Process Modelling in Automotive Industry

Zwikamu Dubani[1], Ben Soh[1] and Chris Seeling[2]
1. Department of Computer Science and Computer Engineering
La Trobe University, Australia 3083
2. Centre for Computational Prototyping, VPAC Ltd, Australia 3053

ABSTRACT

.

The Business Process Modeling Notation (BPMN) was developed by an international body called the Business Process Management Initiative (BPMI). The strengths of BPMN include that it is easily readable and understandable, and could be transformed into the current widely used business process implementation language, the Business Process Execution Language (BPEL). However, there are some fundamental incompatibilities between BPMN and BPEL, which make it difficult to generate BPEL code for implementation from BPMN models.

The aim of this paper is to investigate the synergic uses of BPMN and BPEL to model and implement executable business processes. To that end, we propose a Business Process Modelling framework. Then we apply it to a real life business process scenario involving a typical manufacturing process in the automotive industry. From the case study, we are able to conclude that BPMN is suitable for modelling business processes and then implementing the model in BPEL in spite of the incompatibilities between the former and the latter.

1. INTRODUCTION

Business process management has received considerable attention recently by both business administration and computer science communities. Over the last decades, a number of different technologies have been proposed in order to cope with challenges that businesses face in today's business environment. Today's environments require flexible and adaptable business applications that can cope with and meet frequent changes in business conditions and policies. *Business process management* includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes[1]. The underlying principles of business process management that organizations need to understand in order to manage processes better can be expressed as follows. A *business process* consists of partially ordered activities that correspond to the operations of their defined business in order to achieve their common goal. Figure 1 shows an example flow chart for a business process.

2. MOTIVATION

2.1 Problem Statement

In the engineering discipline, particularly the automobile industry, organizations are actively exploring ways to improve the *Vehicle Development Process (VDP)* used by Motor Vehicle producers. This process is a complex socio-technical problem-solving activity and automotive manufacturers continually adapt it in order to maintain a competitive edge. Their main interest is in understanding how such a large process can be agile. The field of *Business Process Management* has a number of potential solutions for making the VDP more agile and responsive.

2.2 Aim and Scope

The aim of this research is to investigate the use of the Business Process Modeling Notation (BPMN), the Business Process Execution Language (BPEL) and the ActiveBPEL Community Edition Engine to model and implement executable business process workflows in the automobile industry.

3. PROPOSED BUSINESS PROCESS MODELLING (BPM) FRAMEWORK

In order to test the hypothesis mentioned in the previous section, we propose a simple framework for Business Process Modelling. This new BPM framework consists of six main steps (see Figure 2): (1) Identifying business processes, (2) Modelling the business processes in a graphical notation, (3) Implementing the process in BPEL, (4) Deploying the prototype in a BPEL server, (5) Analysis of the process models and, (6) Acceptance as a standard.

3.1 Step 1: Identifying Business Processes

Our BPM framework starts by identifying the business processes in which the organisation is involved. This step allows for selection of processes that need to be modelled and are suitable for automation and optimisation. The benefits of this step are to know which processes can be optimized to: increase productivity; gain competitive advantage by delivering products quicker with reduced costs and higher quality; improve cost effectiveness by reducing cycle time and eliminating unnecessary steps; and improving the processes by measuring them and seeing the results each and every time; one can control only what one can measure.

978-0-7695-3978-2/10 $26.00 © 2010 IEEE
DOI 10.1109/DELTA.2010.48

3.2 Step 2: Designing and Modelling the Business Process

In order for a business process to be adequately explained in a way that is understandable to all stakeholders, it has to be graphically represented as a model. A process model is useful for communication within and across organizations.

In the proposed *modelling phase*, a problem description (workflow scenario) of a business process in the automobile industry given in natural language will be converted into a *Business Process Diagram (BPD)* using any graphical business process diagramming notation. In our case studies, we will use the *Business Process Modeling Notation (BPMN)*.

These elements support the requirement of a simple notation and will adequately model most business processes. Currently, there are many business process modelling languages, tools and methodologies, but we propose to use the BPMN because it is a more standardized notation that helps to unify basic business process concepts as well as advanced modelling concepts. A BPMN diagram can serve the dual purpose of providing a business-level view of a business process and allow the generation of a process executable code through BPEL [2]. For this purpose, BPMN is more appealing than the other notations for the following reasons: (1) BPMN is a notation that is easily readable and understandable for all users who, design, implement and monitor business processes, (2) BPMN can be easily transformed into the Business Process Execution Language (BPEL) for executable business processes and, (3) BPMN allows us to create a Business Process Diagram which represents the activities of the business process and the flow controls that define the order in which they are performed.

3.3 Step 3: Implementing the Business Process Model

To show that the process model from phase 1 (see previous section) is valid, and in order to execute the process, the BPD will be mapped to the Business Process Execution Language for Web Services (BPEL4WS, or BPEL). BPEL was chosen for the implementation task because it is the most frequently used and widely accepted industry business process execution language, and its strengths lie in the good support for workflow patterns and its powerful data structures support.

Unlike other workflow languages, BPEL is a specialized language focused on the definition of business processes and it is less complex than other traditional programming languages. Some features of BPEL which make it more suitable for this purpose are [3]: (1)It can be used to describe business processes through composition of services, to make large processes out of smaller processes and services, (2) BPEL can handle synchronous and asynchronous operation invocation on services, and manage call-backs that occur at a later time, (3) It can invoke service operations in sequence and in parallel, (4) BPEL is capable of correlating requests within and across business processes, (5) Business processes can be structured into different scopes using BPEL and, (6) It can handle both message related and time related events.

3.4 Step 4: Deploying the Business Process Prototype into a BPEL Sever

To execute the BPEL code from the previous section and also to validate our BPD from the *modelling* phase, it will be deployed in the ActiveBPEL Community Edition Engine BPEL server. BPEL servers provide the run-time environment for executing business processes implemented in BPEL. There are a number of BPEL servers that support the execution of BPEL business processes. Below is an overview of some that were considered suitable for this task.

Oracle BPEL Process Manager supports BPEL version 1.1 and provides a complete run-time environment for orchestration of web services and with support for long running transactions[4].

Microsoft Biz Talk is an integration server service with support for integrated business processes and web services[4].

IBM Business Process Execution Language for Web Services Java Run Time (IBM BPWS4J) has been developed by alpha works and provides support for BPEL version 1.1. It has been developed in Java[4].

ActiveBPEL Engine is an open-source BPEL implementation written in Java[5].

3.5 Step 5: Analyzing Resulting Business Process Graphs

The aim of step 5 of our proposed BPM framework is to answer the following questions: (1) Is the process correct?, (2) Are the two graphs related?, (3) What are the differences and similarities between the modeling notation, in this case BPMN and BPEL? And, (4) What can be done about it?

These questions are important because they will help us draw our conclusions and also propose notational elements to adapt the modelling notation (BPMN or any other graphical notation) to BPEL. We need to adapt the modelling notation(s) to reduces the differences between it and BPEL and this can enable reverse engineering, i.e. generating executable BPEL code from a business process diagram, and then generating the original business process diagram from the BPEL code.

3.6 Step 6: Acceptance as a standard

In the last step of our proposed BPM framework, the output of the framework is accepted. This means that an organisation whose business processes are being modelled decides to adopt the tools that were being investigated/tested using our framework for future business process modelling and execution.

4. CASE STUDY

4.1 Scenario: OEM-Supplier Interaction

This section corresponds to the first step of our BPM framework (identifying business processes). We identified the following case study, which is a basic version of an OEM-Supplier interaction process. This example will illustrate a few situations that occur within BPMN diagrams and how they map to BPEL, such as parallel flow, and loops.

In this business process, the OEM has a design that it would like a designated supplier to build. We will not include the commercial arrangement but we will focus on the design change process. A design has a version number. The OEM wants to make sure that the supplier always works to the latest agreed design variation. Say we start with design 1, the supplier needs to acknowledge that the he has received the new design and is working to it. Receiving and working to a design is a two stage process - first the supplier receives the data and acknowledges this. Secondly, the supplier needs to make this new design the working copy.

The OEM also wants to be able to verify when this second step has occurred. Any number of design changes (revisions) can occur during the process so the cycle will be repeated. Also, the OEM will have a number of suppliers it will deal with at any one time.

From the above scenario, we identify a number of issues that the OEM and the supplier have to deal with to satisfy their business goals. The most obvious problems are: (1) acknowledgement of receipt of work by supplier, (2) keeping track of which version of design the supplier is working on (version management) and, (3) change management between both OEM and supplier.

4.2 Issues identified in the case study

From the scenario, clearly there is no standard representation of the interaction process between the supplier and the OEM. This representation is necessary for both the supplier and OEM to know their responsibilities within the context of the business process. It is very important for both OEM and the suppliers to know some of the following: (1) Which steps in the business process are really necessary and who is responsible for them? (2) Who does what and how should it be done? (3) What capabilities are needed? (4) What results are expected?

If the process is defined / modelled in a notation that both the OEM and the Supplier understand, then the diagram will provide both parties with the answers to the above and more questions that may arise.

Due to the evolutionary nature of the engineering design process, it is necessary to provide support for version management. There are lots of reasons why the OEM's initial design could be changed, some of which are: (1) OEM may have new requirements, (2) The designers may want to improve functionality, (3) Production implements continuous change, (4) OEM's installation staff give input on what is really important in the field and, (5) OEM could revise their budget and want to reduce costs.

At any time during the design process, both the OEM and the supplier need to know which version of the design is being worked on. If the working copy on the supplier's side is not the same as the current copy from the OEM's perspective, then the designs would not be what the OEM wanted. OEM may face the following problems: (1) They receive the wrong product or a wrong release of it, (2) They get the wrong replacement part, (3) Deliveries are late, especially at product launch and repair, (4) Budgets are exceeded, (5) The project is not on schedule and, (6) There is lack of visibility of changes and innovation and the new product design suffers.

The above scenario requires the suppliers to take into consideration the need for changes. Communication between OEM and the suppliers is vital for changes to be successful. In many cases, changes will need to be closely monitored so as to minimize the impact of problems on production.

By modelling the above process in BPMN, we will provide a single definition of the OEM-Supplier interaction, from which different views can be rendered. This *unified process representation*, means that different people with different skills (OEM and Supplier) can each view and manipulate the same process via a representation suitable for them and derived from the same source.

4.3 Solution implementation using the proposed BPM framework

To demonstrate the usefulness of our proposed framework, we will implement this business process within the framework with the following goals in mind: (1) To standardize the BPMN so that both the OEM and the supplier can look at the business Process Diagram and the process graph from the ActiveBPEL Community edition engine and understand them and their responsibilities in the business process. There should also be minimal differences between the modelling notation and BPEL in order to support reverse engineering, (2) Define the process such that the supplier cannot work on a design before sending an acknowledgement of receipt to OEM and, (3) Define the process such that OEM can verify which design is a working copy and the supplier always works on the most recent design revision.

4.3.1 Modelling the process in BPMN

Figure 3 shows a Business Process Diagram (BPD) for the OEM-Supplier interaction process, resulting from the first stage of our proposed BPM framework. The process begins with the receipt of a design to build on by the supplier. The supplier then sends an acknowledgement message to the OEM. After the OEM replies that the acknowledgement has been received, the supplier then proceeds to the next step of making the design a working copy. Since the OEM needs to verify that this step has occurred, the supplier sends a message that the step has been done. After the OEM receives the verification, then the supplier starts building the product.

If at any point, the OEM reviews the design for any reason, then the process begins again with the new design version. The process is repeated until the OEM is satisfied with the design. The process ends when there are no new design variations and the supplier completes building the product.

Note that the activities in this process are partitioned using pools (see Figure 3). Each pool represents a participant in the collaborative process. The first pool represents activities performed by the OEM and the second pool represents the Supplier. Message flow is therefore used to handle all interaction between the two participants. Also, note that the dotted lines (Message flow) always occur between two separate pools and do not connect two objects within the same pool. This is because message flow is used only in collaborations.

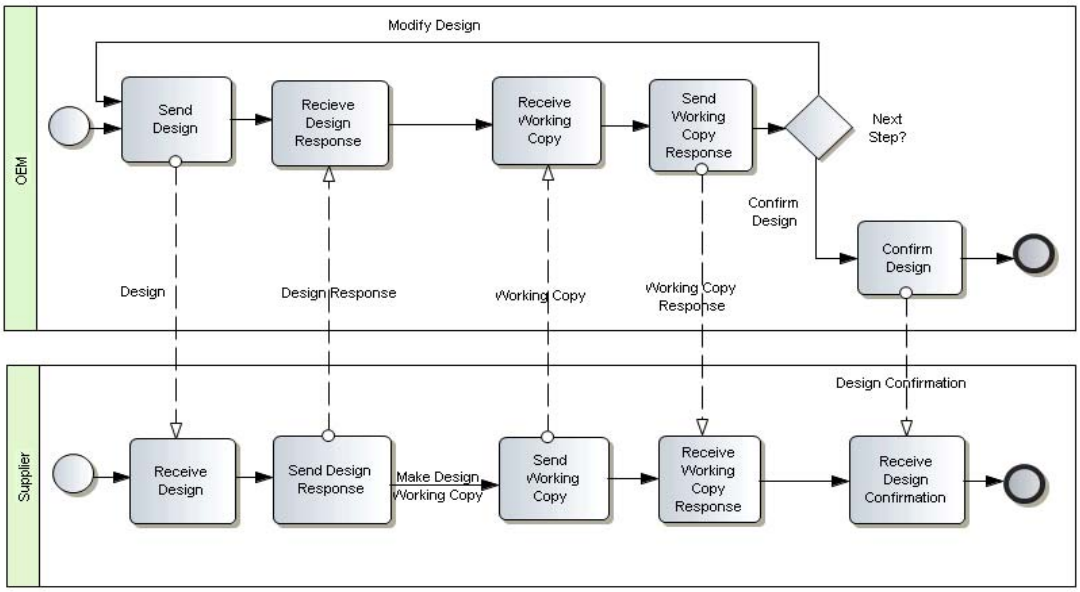


Figure 3: OEM-Supplier interaction process with PMN

5. RESULTS AND DISCUSSION

5.1 Proposed BPM Framework

Using our proposed BPM framework we were able to: (1) identify our business process, i.e. OEM-Supplier Interaction (see section 6.1); (2) design the process model in BPMN (see section 6.3.1); (3) implement the process model in BPEL (section 6.3.2); and (4) deploy the executable process in the ActiveBPEL community edition engine and obtain the process graphs.

The results of all these steps in the proposed framework allow us to answer some of the questions we had before and also to deal with the issues that we identified in the case study. The next sections discuss how our BPM framework helps to deal with these problems in business process modelling.

5.1.1 Identifying our business process

Section 6.1 identified the OEM-Supplier interaction as the business process that needs to be optimized. This is an important step because, as the rest of the case study shows, after identifying a process we were able to: (1) increase productivity by eliminating unnecessary steps; (2) improve cost effectiveness by reducing cycle time and eliminating unnecessary steps; and (3) improve the processes by measuring them and seeing the results each and every time, since one can control only what one can measure.

5.1.2 Design the process model in BPMN

The process model answers the following questions: (1) Which steps in the business process are really necessary and who is responsible for them? (2) Who does what and how should it be done? (3) What capabilities are needed? And (4) What results are expected?

By following our framework, both the OEM and the supplier will be better aware of their responsibilities. The designed model also provides a procedure for information exchange and change management.

5.1.3 Implement the process model in BPEL

Implementing the process in BPEL allowed us to deploy it into the ActiveBPEL Community Edition Engine.

5.1.4 Deploying the executable process in the ActiveBPEL community edition engine

After deploying the process (proposed BPM framework step 5) we are able to obtain a process graph which helps us to continue to the next step in the framework which involves answering the following questions: (1) Is the process correct? (2) Are the two graphs related? (3) What are the differences and similarities between the modeling notation, in this case BPMN and BPEL? And (4) what will be done to address any differences?

5.2 Analysis of the ActiveBPEL Engine's Output Process Graphs

In this section we evaluate the BPEL process implementation model resulting from the deployment of BPEL code into the ActiveBPEL Community Edition Engine. This model is analysed with respect to the BPMN design model from step 5 of our BPM framework.

5.2.1 Sequence of Activities

The sequence of activities from the BPEL code corresponds directly to the BPMN model from which the code was mapped. The message exchanges are implemented in the BPEL code.

5.2.2 Repeating Activities / Loops

It is apparent that repeating activities modelled in BPMN are different from the model obtained from the ActiveBPEL Community Edition Engine. The most obvious differences are: (1) In BPMN, they are represented by an arrow (basic flow chart notation) and a decision box to show where the loop begins and ends. In the BPEL server they are just enclosed / grouped in a box, and (2) The BPMN model shows the loop condition, whereas the server just shows that there is a loop, i.e. in our case study it is when the design needs to be changed .

Also, note that both the while and sequence elements are used in the server's model. Unlike BPMN, if there are many loops in a model and each loop consists of a number of sequences, the model from the ActiveBPEL engine can be complex and difficult to understand.

The last activity that the OEM performs in this process is to confirm the design. This happens after the OEM decides to make no further changes to the design. There is a message from this process to the supplier but it is not shown in the ActiveBPEL engine's process graph view.

6. CONCLUSION

Despite the differences between the Business Process Modeling Notation (BPMN) and the Business Process Execution Language (BPEL), BPMN is suitable for modelling business processes for implementation in BPEL. The ActiveBPEL Community Edition Engine can be used as a standard tool for implementation and execution of automobile industry workflows. Our BPM framework can be applied to evaluate Business Process Modelling Languages (BPMLs). The proposed BPM framework is very useful in the field of business process management. It is a generic framework and can be used by any organization that wants to optimize its processes. Our BPM framework can be applied to evaluate any BPML.

Acknowledgement

This original research was proudly supported by the Commonwealth of Australia, through the Cooperative Research Centre for Advanced Automotive Technology.

REFERENCES

[1] M. Weske, *Business Process Management Concepts, Languages, Architectures*, 1 ed.: Springer-Verlag Berlin Heidelberg, 2007.

[2] Stephen A. White, "Using BPMN to Model a BPEL Process," 2005.

[3] 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 1.1," 2003.

[4] M. B. Juric, B. Mathew, and P. Sarang, *Business Process Execution Language for Web Services, An architect and developer's guide for orchestrating web services using BPEL4WS*, 2nd ed.: PACKT Birmingham-Mumbai, 2004.

[5] "The ActiveBPEL Community Edition Engine," 2008.

[6] S. A. White and D. Miers, *BPMN Modelling and Reference Guide*, 1st ed.: Future Strategies Inc. Lighthouse Point, Florida, USA, 2008.

Figure 1: Flow chart for a business process[6]

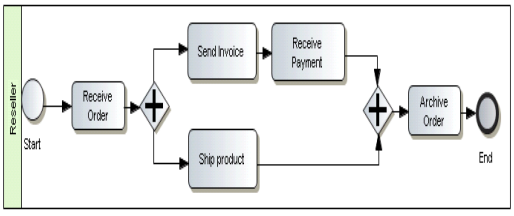
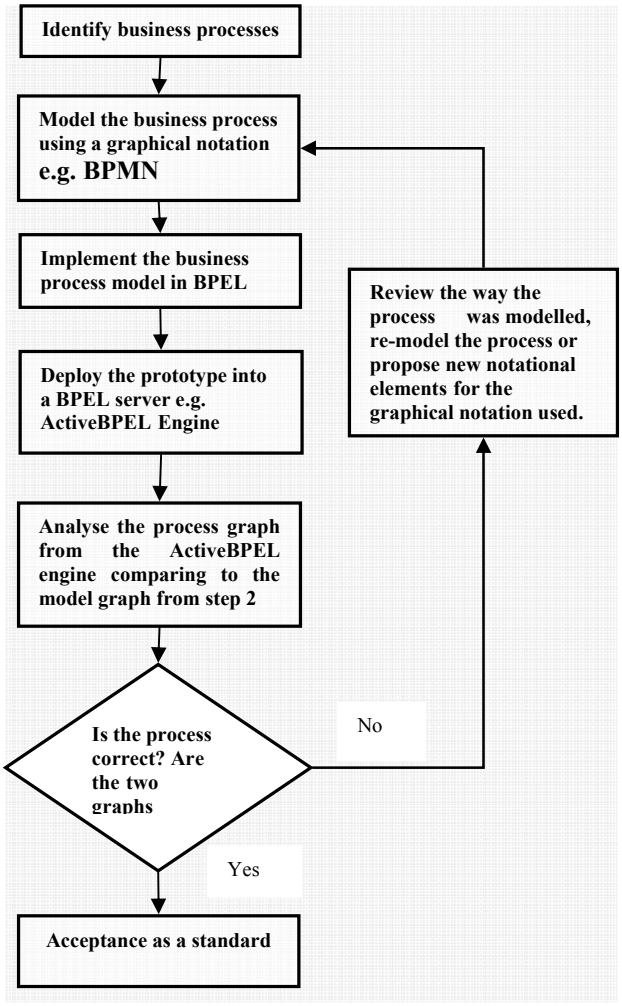


Figure 2: Proposed BPM framework


Identify business processes
Model the business process using a graphical notation e.g. BPMN
Implement the business process model in BPEL
Deploy the prototype into a BPEL server e.g. ActiveBPEL Engine
Analyse the process graph from the ActiveBPEL engine comparing to the model graph from step 2
Is the process correct? Are the two graphs
No
Yes
Review the way the process was modelled, re-model the process or propose new notational elements for the graphical notation used.
Acceptance as a standard