A proposal for a method to translate BPMN model into UML activity diagram

Nguyễn Quốc Bảo

Vietnamese-German University - BIS 2010

contact@nqbao.com

Contents

1. Abstract

BPMN is one of the most popular Business Process modeling language nowadays. But to implementation a BPMN can be a challenged for everyone involved, from Business modeler to Software engineer. UML is stable standard for software implementation and widely adopted in the industry. UML Activity diagram has the same purpose with BPMN and can be considered to be exchangeable. The idea of a method to translate from BPMN model into UML Activity diagram would bring many benefits to the modeler and the business. This report will discuss about a method to translated BPMN model to UML activity diagram without losing semantic meaning and context.

The structure of this report is as following: Section 2 gives an overview of BPMN and UML Activity diagram, Section 3 discuss about the need and benefits for an automated translation method from BPMN to UML Activity diagram. Section 4 discusses about the transformation method, Section 5 extends the topic with BPMN extension to increase the feasibility of the method. Section 6 concludes the report and provides further information about this approach.

2. Overview of BPMN and UML Activity diagram

2.1 BPMN Overview

Business Process Model and Notation (BPMN), also known as Business Process Modeling Notation, is a standard graphical notation used to specify and model a business process. BPMN was originally developed by Business Process Management Initiative (BPMI) and the first version was released in 2004. In 2005, BPMI is merged with Object Management Group (OMG) and BPMN is currently maintained by OMG.

The primary goal of BPMN is to provide a standard notation that is readily understandable by all business stakeholders[1]. BPMN uses a process-oriented approach to business process modeling which focus more on business perspective of the process. BPMN model consists of small set of graphical elements, classified into four categories: Flow Objects, Connecting Objects, Swim Lanes and Artifacts. These elements look pretty much the same to Data Flowchart Diagram help business stakeholders to understand the model easier. There are three levels of abstraction in BPMN: Private (Internal), Abstract (Public) and Collaboration (Global). The three levels are different in term of complexity of the process and depends on the requirements and context of the business business modeler should choose suitable abstraction level.

BPMN is a non-executable Business Process modeling language but the specification of BPMN includes a mapping from BPMN to BPEL (Business Process Execution Language), an executable language allow to implement the process defined in BPMN. Many tools, such as Eclipse STP or Visual Paradigm, allow modeler to generate BPEL model during designing the BPMN model.

The latest version of BPMN at the time when this report is written is BPMN 2.0, released on March 2011.

[1] Source: Wikipedia

2.2 UML and UML Activity diagram Overview

Unified Modeling Language (UML) is a standardized general purpose modeling language in the field of object-oriented software engineering[1]. UML is maintained and developed by OMG. The first version of UML was released in 1995. The current latest version is at the moment when this report is written is UML 2.3, released in 2010. UML 2.4 is being developed and the first beta version was released in March 2011.

The primary purpose of UML is for object-oriented software modeling but the notation also allows modeler to use UML to model non-software system. UML provides various methods to extend UML such as stereotype or UML profile. UML is widely adopted and considered as de facto standard for software modeling and design.

UML Activity diagram is an UML diagram represents the dynamic (or behavioral) views of a system model. UML Activity diagram is used to describe activities, choices, interactions and concurrency of a workflow or a process.

3. The need for a method of automation translation from BPMN model to UML Activity diagram

BPMN and UML Activity diagram have the same purpose and share many same characteristics. Both modeling languages are used to model workflows or business processes and the notation for each language can be translated into the other language under some condition. However, the decision for which modeling language to use is quite difficult for business modeler. There are limitations for both BPMN and UML Activity diagram.

UML has been developed for more than ten years and it is considered as an implementation standard for the industry. There are already methods and tools to help modeler to generate source code from UML models or vice versa. Modeler can take advantages of these methods to reduce time and resource to implement an UML models. But UML has many limitations in business process modeling, there are many studies shown that UML activity diagram is not suitable for business process modeling due to the limitation in presentation power[2]. UML focus on system design and implementation perspective, this approach makes difficulties for modeler to model a business process which usually needs to be abstract and general.

BPMN is more suitable for business process modeling with higher presentation power but lacks of execution semantic[3] which will take more effort to implement a BPMN model.

Therefore, a method of automation translation from BPMN models to UML Activity diagram would be useful for the modeler. A transformation rules can be established to translate BPMN elements to UML Activity diagram element remaining the semantic meaning in original model. This method would allow modeler to be able to take advantages of both presentation power in BPMN and implementation power in UML, business modelers will only have to focus on his expertise in process modeling and software

[1] Source: Wikipedia
[2] Source: Workflow Pattern
[3] BPMN Execution semantic is being developed in BPMN 2.0.

engineers only have to focus on implementation tasks. The communication time between modeler and software engineer is also reduced. Besides that, automation of translation from BPMN to UML Activity diagram would allow to synchronize business process with implementation seamlessly.

The rest of this report will present about the proposal for a transformation rules to translate from BPMN model to UML activity diagram, the possibilities, limitation and further works for this proposal. Implementation of this method is considered further work on this report.

4. BPMN to UML Activity automated translation method

4.1 Transformation rules

BPMN and UML Activity diagram have some elements share the same semantic meaning. These elements are transformed directly without considering about the element context or neighborhood elements. Table 1 describes one-to-one transformation rule from BPMN to UML Activity diagram.

Table 1: BPMN direct transformation rules

BPMN	UML Activity diagram
Pool	Partition
Swim lane	Partition
Task	Activity Node
Sub Process	Structured Activity Node
Sequence Flow	Control Flow
Start Event	Initial Node
End Event	Final Node
Timer Event	Accept Time Action
Throw-Message Event	Send Event Node
Catch-Message Event	Receive Event Node

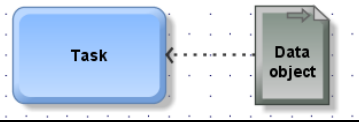
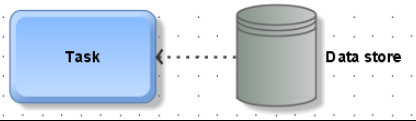
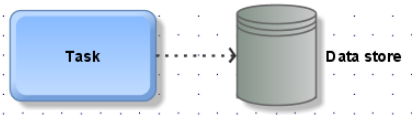
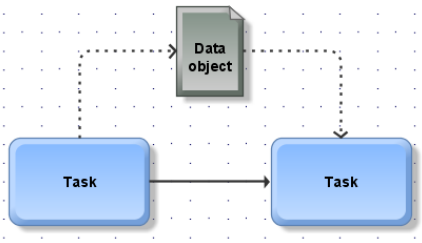
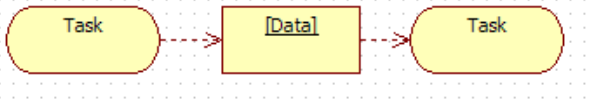
Some BPMN element types cannot be transferred directly to UML Activity diagram elements. To be able to remain the same semantic meaning, these types will be translated to two or more elements in UML Activity diagram. Table 2 describes complex transformation rules from BPMN to UML.

Table 2: BPMN complex transformation rules

BPMN	UML Activity diagram
Send Task	Activity Node + Send Event Node
Receive Task	Accept Event Node + Activity Node
Start Message Event	Initial Node + Receive Event Node
End Message Event	Send Event Node + Final Node
Throw Link Event – Catch Link Event	Control Flow (each pair of link events will result one control flow element)
Conditional Event	Control Flow with Guard condition
Message Flow	Send Event Node + Control Flow

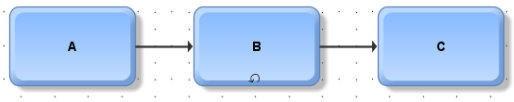
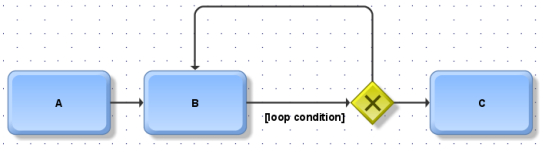
Data object in BPMN will be translated into another activity depends on the direction of the object is input or output. Data store will be translated into Read data store or Write data store activity. Data object with no input or output semantic will be ignored.

Table 3: Translating Data object

BPMN	UML Activity diagram
	
	
	
	
	

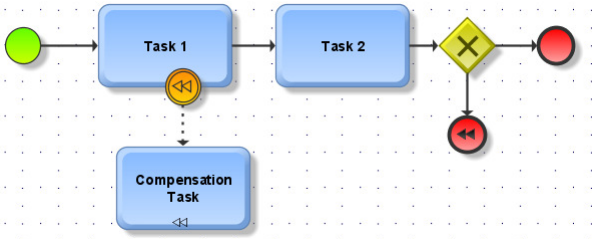
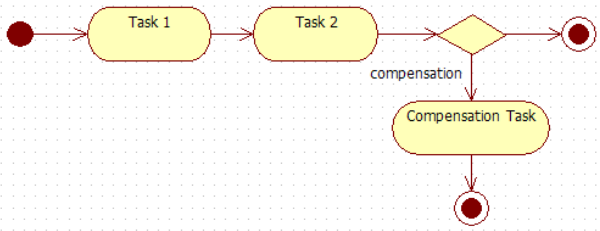
BPMN Loop will be translated back into normal BPMN elements then translated into Activity diagram element.

Table 4: Translating BPMN Loop

BPMN Loop	BPMN Translated Loop
	

Compensation events and compensation activity in BPMN, all compensations task will be processed in parallel at the end of UML process if compensation condition is true. An example of compensation transformation is visualized in the next table.

Table 5: Compensation transformation rule

BPMN	UML Activity diagram
	

Corresponding with BPMN gateway is Fork node, Merge node, Join Node and Decision node in UML AD. To determine which element will be mapped, the context of the element must be examined and depends on the incoming and outgoing flows of BPMN gateway, corresponding element will be selected. The following table describes the mapping rules for BPMN gateway to UML Activity diagram element.

Table 6: BPMN gateway transformation rules

Gateway	Incoming	Outgoing	UML Activity diagram
OR	= 1	> 1	Fork Node
OR	> 1	= 1	Merge Node
OR	> 1	> 1	Fork Node
AND	= 1	> 1	Fork Node
AND	> 1	= 1	Join Node
AND	> 1	> 1	Fork Node
XOR	= 1	> 1	Decision Node
XOR	> 1	= 1	Merge Node
XOR	> 1	> 1	Decision Node

Modeler can also develop a set of UML Stereotype to remain the semantic meaning of BPMN element when translated into UML. UML Stereotype one of different methods to extend UML language. Depends on the context and situation of the modeler, proper stereotype set should be used. The following table describes a set of stereotype used to enhance the semantic of translated element.

Table 7: BPMN stereotype transformation rules

BPMN	UML Activity diagram
Service Task	<< Service >> Activity
User Task	<<User>> Activity
Manual Task	<<Manual>> Activity
Business Rule Task	<<Business Rule>> Activity
Script Task	<<Script>> Activity
Multi-Instance Activity	<<Multi-Instance>> Activity
Multi-Instance Sub Process	<<Multi-Instance>> Structured Activity

Complex elements such as Exception, Cancel and Escalation are not yet considered in this proposal because of the differences of two languages. These special cases need to be transformed manually by modeler.

4.2 Possibilities and limitation of the proposal

The transformation rules can work very well for simple and non-collaboration BPMN model. In reality, the result of the translation need validation from business modeler, the modeler can use the transformation rules as a semi-automation translation method to reduce the time for him when translating the models manually.

The translation method can be also used as an implementation verification method to check for potential issues in the model where transformation rules cannot be applied. Another case when we use BPMN to model exist system with established UML models, addition translation rules can be implemented to translate BPMN model to UML entities (class / method call).

There are many limitations in this approach because the transformation rules do not guarantee that all elements in BPMN will be translated into UML Activity diagram without losing information or semantic meaning. There are some elements cannot be translated because the context is too abstract. The suggested transformation rules for complex elements such as Compensation and Data object may change the meaning of the translated model. Stereotype will require addition work from implementation and may cause difficult to use the translated diagram with other system because of incompatible stereotype.

Besides that, UML Activity diagram contains too many differences to BPMN, which makes it is difficult to develop a complete translation rules from BPMN to UML Activity diagram. In UML Activity diagram, there can be only one controller in one Activity diagram and data cannot be passed from outside of the model. These restrictions make it difficult to model collaboration process in UML Activity diagram. Besides that, Synchronized Merge pattern (below figure) and Interleaved Parallel pattern routing are not supported in UML[1].

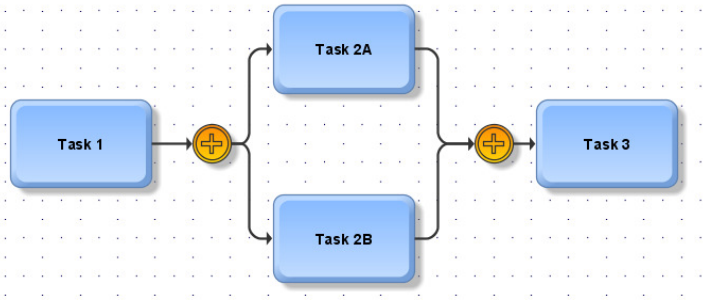


Figure 1: Synchonized Merge in BPMN

UML Activity diagram requires more execution details while it tends to be missing in BPMN because the lacking of execution semantic. BPMN also focuses more on collaboration view of the process and to

[1] Source: Workflow Pattern

improve the readability of the model, this focus will tend to drive BPMN model to a more general view and omit the detail which is necessary for implementation level.

4.3 Improving translation ability for BPMN models

To cover the limitation discussed in the previous section, modeler can alter his modeling method to improve the translation ability for the model. Using BPMN basic elements is more recommended because the modeler will need to specify the modeler at a more detail level which will increase the translation ability of the model. Modeler should avoid usage of semantics may cause confusion such as Loop element and Multi-instance element. In such case, the modeler can re-model the element with basic elements. Data object IO direction should be also specified clearly.

5. Further discussion on BPMN to UML automated translation

Business modeler and software engineer can work together to define a common ontology for both side. In this way, Modeler will design the BPMN model using the given ontology. There are existed effort to build an Ontology for business process such as Open Source Process Management Ontology (http://www.bpiresearch.com/Resources/RE_OSSOnt/re_ossont.htm) and BPMN Ontology (https://dkm.fbk.eu/index.php/BPMN_Ontology). These ontologies can be used to develop domain-specified ontologies for the business.

With the refined BPMN model with ontology, the semantic of the model is enriched and understandable by machine. It would be easier to identify the role and the responsibility for BPMN task based on the defined ontology.

Attempto Controlled English (ACE) can be used as a complementation tool for ontology. A natural-language parser can be used to extract semantic meaning in text description written in ACE. The extracted semantics will be used to match with the ontology to enhance the BPMN model semantics. Based on the semantics, UML artifacts can be identified and generated such as UML Classes, Methods or Parameters. Software engineer can reconstruct the generated artifacts to develop UML class diagram or UML state diagram.

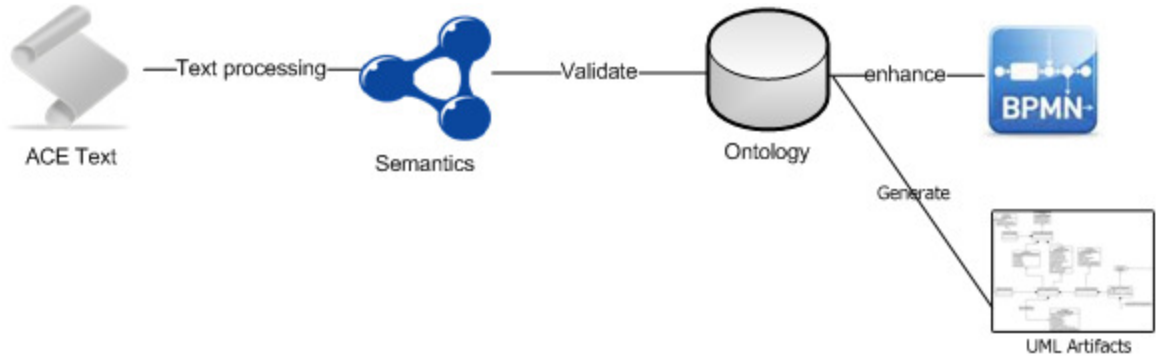


Figure 2: BPMN semantic enhancement using controlled English and ontology

Furthermore, Modeler can consume SysML for modeling language instead of UML. SysML is a profile of UML, specialized in System Modeling. SysML reduces UML's software centric and SysML semantics

are more flexible and expressive[1]. SysML Activity diagram contains additional elements to support for workflow modeling, improving the capability of process modeling for Activity diagram. The translated Activity diagram will remain better semantic than in UML Activity diagram. The below picture visualizes Exception handling feature, which is supported by SysML but not in UML.

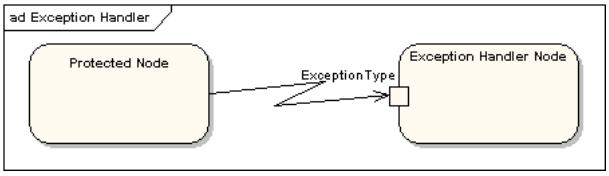


Figure 3: Exception handler is supported directly in SysML. Source: Sparxsystems.com

6. Conclusion

This report describes about a method to automate the translation process from BPMN to UML Activity diagram. BPMN is easier for business modeler to use while UML is more popular for software engineers. The method would allow modeler and everyone involved in the implementation to gain many benefits such as reducing the communication time or seamlessly synchronized BPMN model with the implementation.

However, the gap between BPMN and UML is too large so the method is limited under some situations. Addition methods can be applied to increase the feasibility such as taking advantages of business process ontology and Controlled English.

There are related approaches to convert BPMN to another language which is more feasible for implementation. Translating from BPMN to BPEL or WS-BPEL is more common and the translation can be done automatically when the modeler designs the BPMN model. BPEL is more close to BPMN so the translation is easier than in UML. The translated BPEL diagram can be executed by a BPEL engine to implement the logic from the original BPMN model.

Furthermore, BPMN Execution semantic is being developed in BPMN 2.0. With the execution semantic, the transformation will be easier. The translation rule can combine with the semantic to fully translate the model.

[1] Source: Wikipedia

Reference

- Wikipedia, http://wikipedia.org
- BPMN.org FAQ, http://www.bpmn.org/Documents/FAQ.htm
- BPM Modeling FAQ, http://www.bpmodeling.com/faq/
- BPMN 2.0 Specification, http://www.omg.org/spec/BPMN/2.0
- Cibrán , M.A.: Translating BPMN Models into UML Activities (2009)
- Daniela C. C. Peixoto1, Vitor A. Batista1, Ana P. Atayde1, Eduardo P. Borges1, Rodolfo F. Resende2, Clarindo Isaías P. S. Pádua1.: A Comparison of BPMN and UML 2.0 Activity Diagrams (2008)
- Eloranta, Kallio, Terho (2006): A Notation Evaluation of BPMN and UML AD (2006)
- Dieter, E.J.: Business Process Ontologies: Speeding up Business Process Implementation (2003)