Analysis of BPMN Models

Addis Gebremichael

addisalemayehu.gebremichael@student.uantwerpen.be

Abstract

The Business Process Modeling Notation (BPMN) is a standard notation for capturing business processes, especially at the level of domain analysis and high-level systems design. The highly diverse constructs found in BPMN and the lack of an unambiguous definition of the notation makes it possible to create models with semantic errors. Such errors can lead to significant system development associated costs and can also be difficult to recover. Hence, it is quite essential to verify the semantic correctness of business process models designed with BPMN. This paper, thus, proposes a mapping from BPMN to a formal language, namely Petri nets, for which efficient analysis techniques can be made.

Keywords: BPMN, process modeling, petri nets, business processes, workflow nets

1. Introduction

The Object Management Group (OMG) has developed a standard Business Process Model and Notation (BPMN)[1]. The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes[1].Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.

The next section highlights factors that contribute to the development of BPMN and further sections introduce the basic constructs of BPMN followed by a discussion to the choice of petri nets as a good candidate for formally defining the semantics of BPMN models. Finally, the paper concludes with a work plan schedule for the practical project that demonstrates this study.

2. BPMN Overview

The control-flow perspective (modeling the ordering of activities) is often the backbone of a process model[2]. However, other perspectives such as the resource perspective (modeling roles, organizational units, authorizations, etc.), the data perspective (modeling decisions, data creation, forms, etc.), the time perspective (modeling durations, deadlines, etc.), and the function perspective (describing activities and related applications) are also essential for comprehensive process models[2]. BPMN constructs capture these different process modeling perspectives. In addition, BPMN provides businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner.

Another factor that drove the development of BPMN is that,historically, business process models developed by business people have been technically separated from the process representations required by systems designed to implement and execute those processes[3]. Thus, there was a need to manually translate the original business process models to the execution models that is highly subject to errors. To help alleviate the modeling technical gap, a key goal in the effort to develop BPMN was to create a bridge from the business-oriented process modeling notation to IT-oriented execution languages. Hence, this was achieved by mapping BPMN process models on to the Business Process Execution Language for Web Services (BPEL4WS),i.e. a standard for defining business processes at the implementation level[3].

3. BPMN Elements

It should be emphasized that a modeling language has to create a simple and easily understandable mechanism for capturing relevant entities, while at the same time being able to handle the complexity inherent to the system. The approach taken to handle these two conflicting requirements, during the development of BPMN, was to organize the graphical aspects of the notation into specific categories[1].

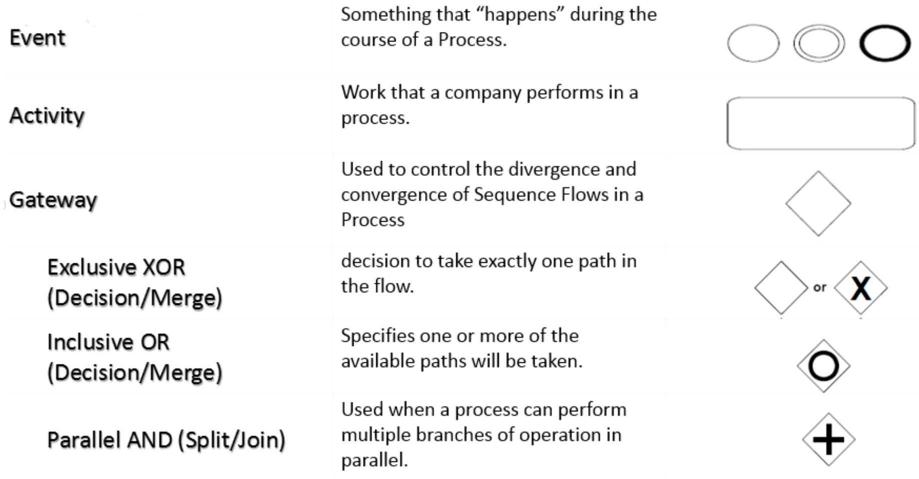
This provides a small set of notation categories so that the reader of a BPMN model can easily recognize the basic types of elements and understand the diagram.Within the basic categories of elements, additional variation and information can be added to support the requirements for complexity without dramatically changing the basic look and feel of the model.

The proposed mapping , however, focuses on the control-flow perspective of BPMN, that is, the subset of the notation that deals with the order in which activities and events are allowed to occur. It does not deal with its non-functional features (i.e. artifacts, groups and associations) and organizational modeling features (i.e. lanes and pools). Also, the proposed mapping is specifically designed to produce Petri nets that are suitable for static analysis.

The five basic categories of elements according to BPMN v2.0.2 are [1]:

- Flow Objects
- Connecting Objects
- Swimlanes
- Data
- Artifacts

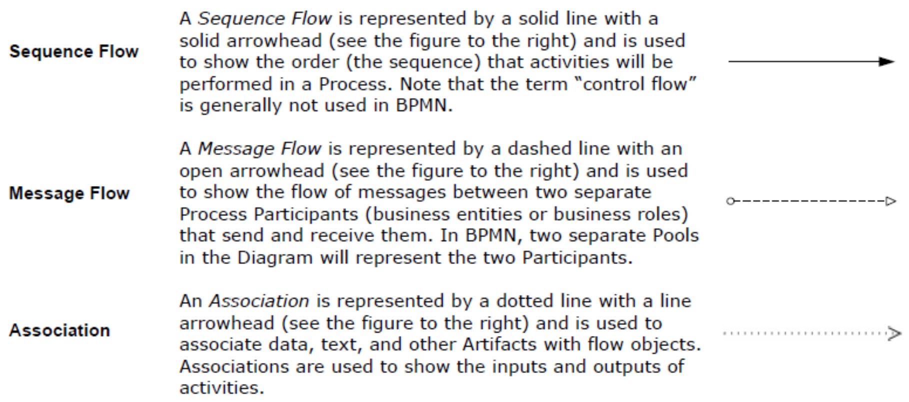
Flow Objects. These are the main graphical elements to define the behavior of a Business Process. There are three Flow objects as described in figure 1



Event	Something that "happens" during the course of a Process.	○ ◎ ⬤
Activity	Work that a company performs in a process.	▭
Gateway	Used to control the divergence and convergence of Sequence Flows in a Process	◇
Exclusive XOR (Decision/Merge)	decision to take exactly one path in the flow.	◇ or X
Inclusive OR (Decision/Merge)	Specifies one or more of the available paths will be taken.	O
Parallel AND (Split/Join)	Used when a process can perform multiple branches of operation in parallel.	+

Figure 1: Core BPMN flow objects

Connecting Objects. The Flow Objects are connected together in a model to create the basic skeletal structure of a business process. The basic Connecting Objects that provide this function are described in figure 2



Sequence Flow	A *Sequence Flow* is represented by a solid line with a solid arrowhead (see the figure to the right) and is used to show the order (the sequence) that activities will be performed in a Process. Note that the term "control flow" is generally not used in BPMN.
Message Flow	A *Message Flow* is represented by a dashed line with an open arrowhead (see the figure to the right) and is used to show the flow of messages between two separate Process Participants (business entities or business roles) that send and receive them. In BPMN, two separate Pools in the Diagram will represent the two Participants.
Association	An *Association* is represented by a dotted line with a line arrowhead (see the figure to the right) and is used to associate data, text, and other Artifacts with flow objects. Associations are used to show the inputs and outputs of activities.

Figure 2: BPMN connecting elements

Swimlanes. Many process modeling methodologies utilizes the concept of swimlanes as a mechanism to organize activities into separate visual categories in order to illustrate different functional capabilities or responsibilities. BPMN supports swimlanes with two main constructs.

Pools are used when the diagram involves two separate business entities or participants (see Figure 3) and are physically separated in the diagram. The activities within separate Pools are considered self-contained Processes. Thus, the Sequence Flow may not cross the boundary of a Pool.

Message Flow is defined as being the mechanism to show the communication between two participants, and,thus, must connect between two Pools (or the objects within the Pools). Lanes are more closely related to the traditional swimlane process modeling methodologies. Lanes are often used to separate the activities associated with a specific company function or role (see figure 4). Sequence Flow may cross the boundaries of Lanes within a Pool, but Message Flow may not be used between Flow Objects in Lanes of the same Pool.

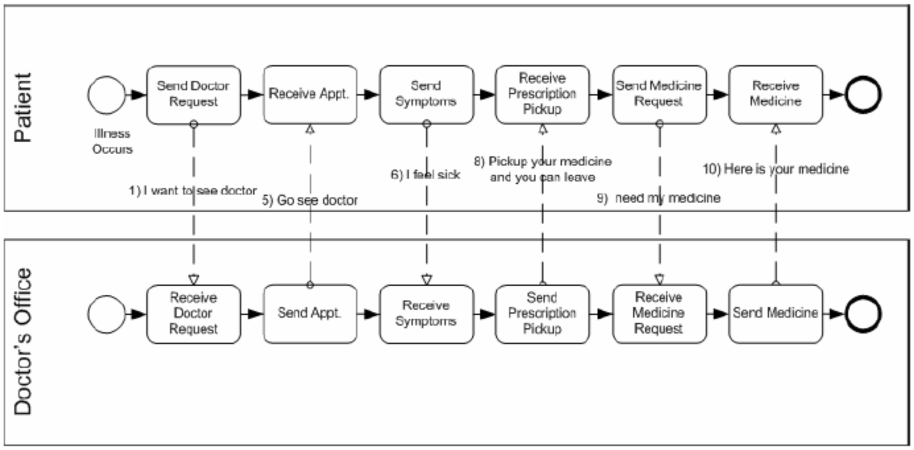


Figure 3: An Example of a BPD with Pools

Data and Artifacts. Data is represented with elements like data objects, data inputs, data outputs and data stores. The data object element is described in the following table. In addition to data, artifacts are used to provide additional information about the Process. There are two standardized Artifacts, but modelers or modeling tools are free to add as many Artifacts as necessary. The current set of Artifacts are namely group and text annotation as described in figure 5

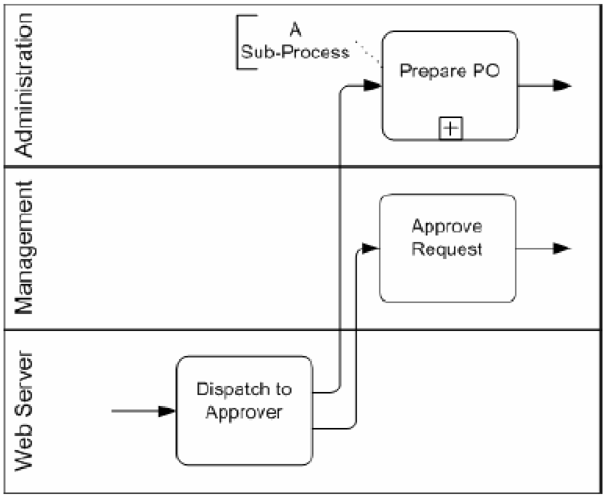
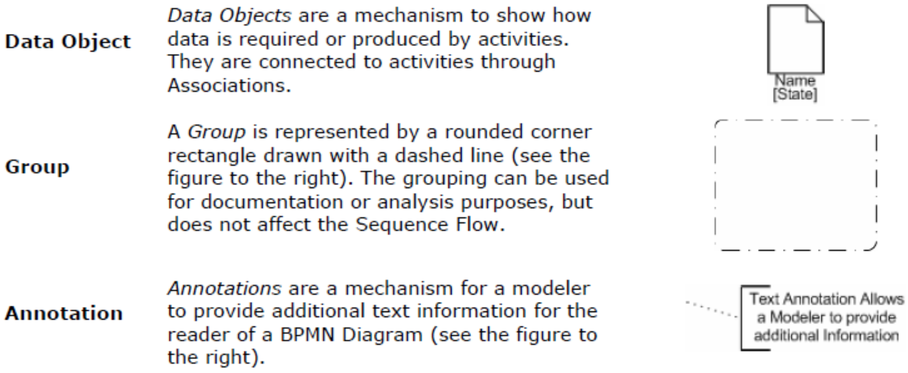


Figure 4: A Segment of a Process with Lanes



Data Object	*Data Objects* are a mechanism to show how data is required or produced by activities. They are connected to activities through Associations.	Name [State]
Group	A *Group* is represented by a rounded corner rectangle drawn with a dashed line (see the figure to the right). The grouping can be used for documentation or analysis purposes, but does not affect the Sequence Flow.	
Annotation	*Annotations* are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram (see the figure to the right).	Text Annotation Allows a Modeler to provide additional Information

Figure 5: Data object and Artifact elements

4. BPMN Model Analysis

BPMN models are a great mechanism for business process modeling since they are developed with constraints imposed by businesses and their work flow management techniques. However, these notations lack unambiguous definition [4] and the mix of these constructs can lead the designer to create models with semantic errors. Hence why the ability to check for semantic correctness is valuable. The proposed project then employes a model driven engineering approach to first develop a visual modeling formalism for a subset of BPMN and provide denotational semantics for BPMN by mapping on to Petri Nets so as to conduct analysis on the correctness of models in BPMN .

4.1. Why petri nets for mapping?

Petri nets are a well-founded process modeling technique and play an even more prominent role in BPM as they are graphical and able to model concurrency[2]. Since the invention of classical Petri nets in the sixties, they have been used to model and analyse all kinds of processes with applications ranging from protocols, hardware, and embedded systems to flexible manufacturing systems, user interaction, and business processes. Petri nets are particularly suited to model behavior of systems in terms of flow", be it the flow of control or flow of objects or information[4]. This feature makes Petri nets a good candidate for formally defining the semantics of BPMN models, since BPMN is also flow-oriented.

Constraints may be imposed on the structure of Petri nets depending on the intended purpose. In this paper, the aim is at generating Petri nets that conform to the following restrictions, also known as workflow nets[4]. Intuitively, a workflow net models the execution of one instance of a business process, from its creation up to its completion.THis is achieved by having a unique begin place (i.e. a single place is not the target of any arc), a unique end place (i.e. a single place that is not the source of any arc), and every other place and transition is on a directed path from the unique source place to the unique sink place. In other words, workflow nets have a distinguished start place and a distinguished end place. For example, the Petri net in figure 6 is a workflow net.

4.2. Analysis Criterion

Petri nets are marked by the availability of many analysis techniques. Clearly, this is a great asset in favor of the use of Petri nets for business process modeling.

The technique to be used is reachability analysis as follows[4]:

- *Absence of dead tasks,* i.e. there are no tasks that can never be performed within a model. It can be checked through the absence of dead transitions within the corresponding net. In other words, for any transition there is a reachable state enabling it.
- *Proper completion,* i.e. any process instance eventually reaches a completion state.As formulated in the previous section, a process instance is completed if it has reached the end event and there are no enabled tasks. In Petri net terms, it implies any reachable state having a token in the sink place does not have a token in any of the other places.

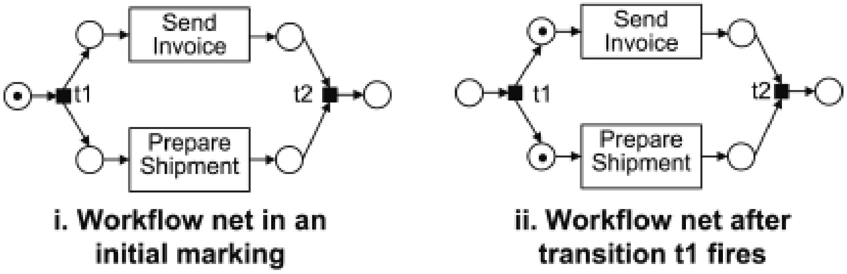

Send Invoice
Prepare Shipment
t1
t2
i. Workflow net in an initial marking
Send Invoice
Prepare Shipment
t1
t2
ii. Workflow net after transition t1 fires

Figure 6: Sample workflow nets in two different states

5. Project Schedule

For the practical project, first a visual modeling formalism for a subset of BPMN will be build using AToMPM. Then a set of transformation rules and scheduling of the rules will be constructed for mapping on to petri nets constructs. Thereafter, generate code for the PNML platform to perform analysis. Finally, test case BPMN models will be chosen to conduct analysis and the results will be further discussed.

References

[1] Object Management Group, http://www.omg.org/spec/BPMN, Business Process Model and Notation (BPMN).

[2] W. P. van der Aalst, Business process management: A comprehensive survey, ISRN Software Engineering 2013 (2012) 37. doi:w.m.p.v.d.aalst@tue.nl.

[3] S. A. White, Introduction to BPMN, Tech. rep., IBM Corporation.

[4] R. M. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in BPMN, Information and Software Technology 50 (12) (2008) 1281–1294. doi:http://eprints.qut.edu.au/7115/.