Informatics

How to Model and Optimize Executable BPMN Process Models

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software & Information Engineering

by

Dragana Sunaric
Registration Number 11814569

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Inf. Dr.-Ing. Jürgen Dorn

Vienna, 1st January, 2022

Dragana Sunaric

Jürgen Dorn

Declaration of Authorship

Dragana Sunaric

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 1st January, 2022

Dragana Sunaric

Acknowledgements

write Acknowledgements

Abstract

write Abstract

Contents

CHAPTER 1

Introduction

write Introduction

CHAPTER 2

Automating Business Processes

The following chapter will provide an overview of the current research on executable Business Process Model and Notation (BPMN) models. First it will discuss the motivation behind process automation and using BPMN Workflow Management Systems and the benefit it might bring to an organization. Then it will state the differences between an conceptual BPMN model, that cannot be deployed on an Business Process Workflow Management System, and an executable BPMN model. Finally this chapter will list the steps necessary to turn an conceptual BPMN model executable.

2.1 Why automate Processes?

While process automation seems to be in itself a pursued goal, there are some quantifiable advantages of using a Business Process Management System Business Process Management System (BPMS) besides the possibility to automate certain process steps.

- **Shorten process lifetimes**: Managing processes manually requires handling tasks such as starting sub processes or handling handovers from one entity to another manually and can lead to unnecessary waiting periods between two operation tasks in the process. By implementing an Workflow Management System resource allocation and parallelization can be automated where possible to assure optimal use of process resources. [4]

- **Reduction in process cost**: By reducing process lifetimes and increase productivity due to better handling resources can reduce process costs [4] but due to the high price for Business Process Workflow Management Systems the overall cost does not have to decrease necessarily [5].

- **Workload reduction**: As stated earlier, managing processes needs to be done either by an Business Process Workflow Management System or manually which creates additional workload for an organization. Workloads for employees executing the processes are also kept steady due to dynamic resource assignment. [2][4]

- **Enforce rules**: Defining the processes that is then directly executed and controlled by an Business Process Workflow Management System enables the organization to enforce the execution of the process at it is designed. Following Rules and Protocols can partially be automated and enforcing guidelines and laws in an organization becomes easier. [2]

- **Create Transparency**: Using an Business Process Workflow Management System provides insights to the actually processes that are executed in the organization. It makes it easier to determine the performance of the processes by providing historical information of completed process instances and provides insight of the current status of processes that are still in progress. [4]

2.2 Executable vs Conceptual Process Models

Process Models are inherently Business-oriented as their purpose is initially to define and visualize the processes happening in an organization. Business-oriented or conceptual BPMN model are meant to be read by domain experts and contain usually implicit information known to these domain experts [2]. They are usually incomplete, meaning that not every possible outcome of a process is modeled. In fact, usually only the best-case scenario, also known as the 'happy-path', of a process is modeled in an conceptual BPMN model [3].

Because of this, business-oriented models cannot be executed in an Business Process Workflow Management System as they are but have to be turned into an executable BPMN model. executable BPMN model are meant for IT-experts and should be a technical representation of the business process while still begin understandable by domain experts. They should leave no room for interpretation as they have to contain all the information necessary for the process to be executed using an Business Process Workflow Management System. Besides the visual information, executable BPMN model also need to contain execution properties like interface definitions and Variables that are used by the Process called Process Variables.[2]

An algorithm on how to efficiently turn an conceptual BPMN model executable as stated in the book *Fundamentals of Business Process Management* [2] is described in the next section.

2.3 Making Process Models executable

As stated earlier, BPMN models can not directly be executed by a BPMS but have to be converted from an conceptual BPMN model into an executable BPMN model.

There are different approaches how an executable BPMN model can be derived from a business-oriented conceptual BPMN model. In [2] performing such a transformation is broken down into 5 steps:

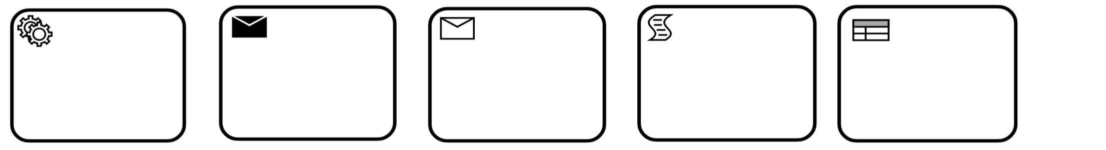
1. Identify the automation boundaries
2. Review manual tasks
3. Complete the process model
4. Bring the process model to an adequate granularity level
5. Specify execution properties

2.3.1 Identify the automation boundaries

The first step in turning an conceptual BPMN model in an executable BPMN model is to identify which steps can be automated using a Business Process Workflow Management System.

Tasks which can inherently be automated are called Automated tasks [2, p. 317]. Taking a look at the BPMN 2.0 standard an Automated task can be one of the following Task types:

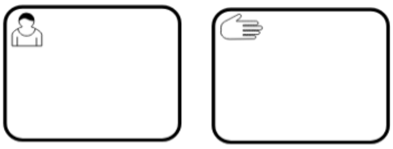
- **Service Task**: A Task that invokes a service. Can be a Webservice or a application code.
- **Send Task**: Used to send a message to an external participant (A participant that is not part of the process)
- **Receive Task**: Used to receive a message from an external participant
- **Script Task**: Executes a Script that can be interpreted by the BPMN Workflow Management System
- **Business Rule Task**: Executes a rule. (e.g. provides input for a business rule engine and gets the output of that calculation)



(a) Service Task (b) Send Task (c) Receive Task (d) Script Task (e) Business Rule Task

Figure 2.1: Automated Tasks according to the BPMN 2.0 standard [6]

Usually not every step of a Process can be fully automated. Processes can also have **Manual Tasks** and **User Tasks**. A **User Task** is a Task performed by a User with the aid of an Business Process Workflow Management System while a **Manual Task** does not use any help from a business process execution engine.



(a) User Task (b) Manual Task

Figure 2.2: Manual and User Tasks according to the BPMN 2.0 standard [6]

2.3.2 Review manual tasks

As mentioned earlier Manual Tasks are not automated and do also happen without the aid of a Business Process Workflow Management System. In this step we need to analyse if the identified manual tasks in our BPMN-model can be incorporated into the Business Process Workflow Management System. This can be done in two ways:

- **Automate the task**: Depending on the nature of the task that is currently performed manually, it might be possible to fully automate the task or to model the task as a Receive Task where the modelled task waits for a message that indicates the physical manual tasks completion.
- **Turn it into a User task**: If the task cannot be automated or the organization lacks the resources to automate the task yet, it might be possible to turn the Manual Task into a User Task. One possibility could be to dedicate a person in charge of the manual task to notify the BPMN Workflow Management System on the completion of the task via a worklist handler. [7]

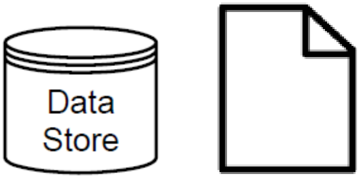
In the case that neither an Automated task nor an User Task is suitable for modelling the Manual Task one might also consider isolating the task and modelling the rest of the process. If this is also not possible due to the manual task being crucial for the expressiveness of the model it might be reconsidered if this process can or should be executed using a Business Process Workflow Management System[3, p. 228]

2.3.3 Complete the process model

Usually conceptual BPMN models are not complete and leave out certain informations that are seen as implicit knowledge or as not important by the person modelling the process but might be crucial if a complete picture of the process is needed for automation.

A common flaw in many conceptual models is ignoring errors and only implementing the 'happy path'. The 'happy-path' is the best-case scenario that can happen in the execution of a process. While it might be sufficient for a conceptual BPMN model, showing the process for a customer order, to not show what happens in case the product is out of stock or what happens if the payment does not work, a executable BPMN model has to take into account what happens in case an error occurs.

It is also necessary to model the input and output data of our tasks in this step using Data Stores and Data Objects.



(a) Data Store (b) Data Object

Figure 2.3: Data store and Data Objects according to the BPMN 2.0 standard [6]

2.3.4 Bring the process model to an adequate granularity level

The granularity of tasks modelled in an conceptual BPMN model does differ from the granularity needed in an executable BPMN model. The goal of process automation is not to automate as much as possible but to have a centralized BPMN Workflow Management System that does not only decide what task has to be done when but also who needs to be doing this task. [3]

Therefore consecutive tasks that are done by the same participant should be clustered together as one task to minimize handovers that have to be unnecessarily processed by the BPMN Workflow Management System. [2] However there are some exception to this rule:

- Tracking progress: In order to know how much the process has advanced it can be useful to split certain parts even if they are done by the same person.
- Handling exceptions: If for a set of tasks that it performed by the same participant different erros and exceptions can occur, it might be useful to keep these tasks separated.
- Managing Resources: Sometimes consecutive tasks are performed by participants that have the same role, but need to be done by two different participants. For example: two different people need to sign the same document in a process. In this case it is also necessary to disaggregate the task accordingly.

2.3.5 Specify execution properties

The final step in turning an conceptual BPMN model executable is to specify the details of the implementation details of our BPMN model. While the changes performed up to this step had an impact of the graphical representation, the execution properties are not graphically embedded into BPMN but are encoded into the Extensible Markup Language XML (XML) representation of the BPMN-model. [2] A schematic representation of the structure if BPMN is provided in 2.4. For the full specification of the BPMN-XML the XML-schema can be found on the OMG-Website[1].

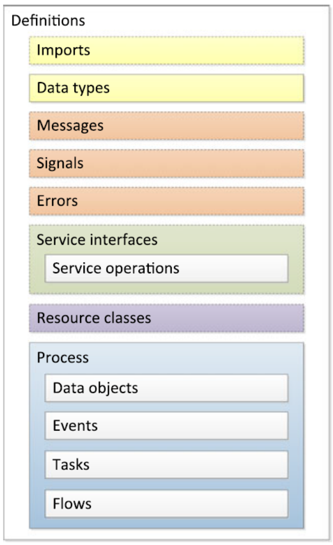


Figure 2.4: Structure of the BPMN format [2]

Process Variables

In order to use data in different elements of our process, we need process variables that can be read, created and modified during the processes execution. Every Process variable has a **data type** that can either be simple (strings, integers, doubles, booleans, dates, times, ...) or complex (composed of other types). A complex Type needs to be described as an XML Schema Definition XSD (XSD)-Schema File.

```
<xs:element name="person">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="name" type="xs:string"/>
                        <xs:element name="address" type="xs:string"/>
                        <xs:element name="city" type="xs:string"/>
                        <xs:element name="country" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

Listing 2.1: The XML-Schema Definiton for a complex type 'person'

```
<person>
        <name>Dragana Sunaric</name>
        <address>Wiedner-Hauptstrasse 5</address>
        <city>1040 Wien</city>
        <country>Austria</country>
</person>
```

Listing 2.2: An instance of the complex type 'person'

The definition of common Errors, Messages and Escalations that are thrown or listened to by Events and Tasks are aslo part of the execution properties. Every Element has at least an **id** that identifies the given Element and a descriptive **name** of the element. [6]

Messages

```
<message id="Message_ID" name="Message_NAME"/>
```

Errors

Errors additionally have an **errorCode** that specifies the given Error. Events can listen for this specific error code and trigger when it is thrown.

```
<error id="Error_ID" name="Error_NAME" errorCode="Error_CODE"/>
```

Escalations

Similar to errors, escalations additionally have an **escalationCode** that specifies the given Error and can be listened to by events.

```
<escalation id="Esc_ID" name="Esc_NAME" escalationCode="Esc_CODE"/>
```

Input and Output Variables

As mentioned earlier, Process Variables are active during the whole Process life-cycle. Apart from this data than can be accessed globally, it is also possible to define input and output values for each Task or Event in our process-model. These values are only visible within the Task or Event and have to be defined as an XSD-Schema File (similarly to complex process variables). [2]

Service Tasks

In order for service tasks to call external application or web-services, the interaction with the given service has to be defined in the process model. Connected services need to provide an service interface that describes the available service-operations and their parameters as well as return values. Service-operations can be synchronous, meaning the process instance waits for the operation to finish and to return a value or error code, or asynchronous, meaning the process does not wait for a response and carries on with the process after calling the service. Based on the service interface definition, input and output variables have to be defined for the service-call. The Business Process Workflow Management System does this by copying the above mentioned Input values of the Task into the service call and if necessary, copy the output values of the service call into the output values of the Service task. [2]

CHAPTER 3

Evaluating and optimizing executable Process Models

3.1 Six Sigma Approaches

The first set of analysation and optimization techniques discussed in this chapter originate from the Six Sigma initiative that aims to minimize the failure rate of Processes. The name originates from the Six sigma interval in a normal distribution that indicates the aimed success rate of 99.99966%. At the moment

3.1.1 Process Map

3.1.2 Check Sheets

3.1.3 Pareto Analysis

3.1.4 Cause and Effect Diagram

3.1.5 Root Cause Analysis

3.1.6 Quality Function Deployment (QFD)

3.2 Other Qualitative Measures

3.2.1 Value Added Analysis (VAA)

3.3 Quantitative Analysis

3.3.1 Performance Measures

3.3.2 Flow Analysis

3.3.3 Queues

3.3.4 Simulation

3.4 Benchmarking Processes

CHAPTER 4

Concept and Implementation

CHAPTER 5

Case Study

Write case study

CHAPTER 6

Conclusion

write conclusion

List of Figures

List of Tables

Listings

Glossary

'happy-path' The best-case scenario in a process. 4, 6

Automated task A task that can be automated in a BPMN Workflow Management System. 5, 6

BPMN Workflow Management System A software for executing workflows specified as BPMN models. 3, 5–7, 23

Business Process Management System see BPMN Workflow Management System. 3, 25

Business Process Workflow Management System see BPMN Workflow Management System. 3–6, 10, 23

Business Rule Task "provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide"[p. 163][6]. 5

conceptual BPMN model A BPMN model describing a business process that cannot be directly executed on a Business Process Workflow Management System. 3–7

Data Object "Data Objects representing a Collection of Data"[p. 206][6]. 7

Data Store "A DataStore provides a mechanism for Activities to retrieve or update stored information that will persist beyond the scope of the Process"[p. 208][6]. 7

executable BPMN model A software for executing workflows specified as BPMN models. 3–5, 7

Manual Task "A Task that is expected to be performed without the aid of any business process execution engine"[p. 163][6]. 6

Process Variables Process variables are managed by the BPMS engine to allow data exchange between process elements. 4

Receive Task “designed to wait for a Message to arrive from an external Participant“[p. 161][6]. 5, 6

Script Task “executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret“[p. 164][6]. 5

Send Task “designed to send a Message to an external Participant“[p. 159][6]. 5

Service Task “Task that uses some sort of service, which could be a Web service or an automated application“[6, p. 158]. 5

User Task “A typical “workflow” Task where a human performer performs the Task with the assistance of a software application“[p. 163][6]. 6

XML Extensible Markup Language - Defined by W3C[**?**]. 8, 21

XSD XML Schema Definition - Definition of the stucture of a XML File[**?**]. 8, 9

Acronyms

BPMN Business Process Model and Notation. 3–8, 23

BPMS Business Process Management System Business Process Management System. 3, 4

XML Extensible Markup Language XML. 8

XSD XML Schema Definition XSD. 8

Bibliography

[1] Bpmn xml schema specification. `https://www.omg.org/spec/BPMN/20100501/BPMN20.xsd`. Accessed: 2022-04.13.

[2] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management*. Springer Publishing Company, Incorporated, 2nd edition, 2018.

[3] J. Freund and B. Rücker. *Real-Life BPMN (4th Edition): Includes an Introduction to DMN*. Independently Published, 2019.

[4] Andreas Gadatsch. *Grundkurs Geschäftsprozess-Management: Analyse, Modellierung, Optimierung und Controlling von Prozessen*. Springer-Verlag, 2020.

[5] Horst Gruber and Christian Huemer. Profitability analysis of workflow management systems. In *2009 IEEE conference on commerce and enterprise computing*, pages 233–238. IEEE, 2009.

[6] OMG. Business Process Model and Notation (BPMN), Version 2.0.2, December 2013.

[7] Geno Stefanov. Business process automation with bpms. In *Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE)*, page 617. International Conference on Application of Information and Communication . . . , 2014.