ASSIGNMENT OF BACHELOR’S THESIS

Title: Automated Transformation of DEMO Models into BPMN
Student: Štěpán Tužil
Supervisor: Ing. Robert Pergl, Ph.D.
Study Programme: Informatics
Study Branch: Information Systems and Management
Department: Department of Software Engineering
Validity: Until the end of summer semester 2019/20

Instructions

The goal of this thesis is to practically evaluate the method of transformation from DEMO to BPMN models published in [1] through implementation of the described transformation and its application on a case study.
1. Acquaint yourself with the DEMO methodology and BPMN notation.
2. Study the paper [1] presenting a transformation of DEMO models into BPMN.
3. Evaluate the currently available tools for DEMO and BPMN modelling with the respect to possibility of the transformation implementation.
4. Realize the transformation from the selected DEMO tool into the selected BPMN tool.
5. Demonstrate the transformation on a case study.
6. Test and document your solution and evaluate its usability in enterprise practice.

References


Ing. Michal Valenta, Ph.D.
Head of Department
doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague December 22, 2018
I would like to thank my supervisor, Robert Pergl for his support and advices while creating the thesis. Then I would like to thank my parents and friends, for their support.
Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work for non-profit purposes only, in any way that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on May 15, 2019

............................................
Czech Technical University in Prague
Faculty of Information Technology
© 2019 Štěpán Tužil. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author’s permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Abstract

This thesis deals with automatic transformation of DEMO models into BPMN models. Both of these models and their notation is described in this work. The modeling tools for both types of models are reviewed here and their main advantages and disadvantage are listed.

Main tools made for this thesis, used for transformation of DEMO models to BPMN models, is based on Mraz’s method. This tools was successfully developed in Python, the scripting language.

BPMN models, created with this tool, are described, reviewed for their usability and their ability to correctly capture DEMO models properties.

Keywords  DEMO, BPMN, Businesses process management, Business process model, Python
Abstrakt

Tato bakalářská práce se zabývá automatickým převodem DEMO modelů do BPMN modelů. V práci se nachází popis obou těchto modelovacích technik. Dále jsou v teoretické části zhodnoceny nástroje pro tvorbu těchto modelů a sumarizace jejich hlavních kladů a záporů.

Hlavní nástroj vytvořený pro tuto práci, sloužící pro převod DEMO modelů do BPMN modelů, je založen na Mrázově metodě. Tento nástroj byl úspěšně vytvořen ve skriptovacím jazyce Python.

U vytvořených modelů, pomocí tohoto nástroje, je posouzena kvalita převodu, schopnost zachycení vlastností DEMO notace v BPMN notaci a je zhodnocena jejich použitelnost v praxi.

**Klíčová slova**  DEMO, BPMN, Procesní řízení, Procesní model, Python
# Contents

1 Introduction ................................. 1

2 Goals and methodology .................. 3
   2.1 Goals .................................. 3
   2.2 Methodology ............................ 3

3 Review ...................................... 5
   3.1 Definition of basic concepts .......... 5
   3.2 BPMN Notation .......................... 8
   3.3 DEMO Models ............................ 13

4 Analysis .................................... 25
   4.1 Modeling tools .......................... 25
   4.2 Mraz’s transformation approach ..... 28
   4.3 Transformation for happy-flow ...... 32

5 Design and implementation ............. 35
   5.1 Transformation tool .................... 35

6 Evaluation .................................. 37
   6.1 Testing .................................. 37
   6.2 The solution in the context of BPM .. 38

7 Contribution ............................... 39
   7.1 Usefulness in BPM ...................... 39
   7.2 Future research ........................ 39

Conclusion .................................. 41

Bibliography ................................. 43
## List of Figures

3.1 Basic flow chart [1] .............................................. 7
3.2 Symbols of activities – Task and Subprocess .................. 9
3.3 Types of starting events ........................................ 10
3.4 Types of intermediate events ................................. 10
3.5 Ending events .................................................... 11
3.6 Symbols of gateways ............................................ 11
3.7 Symbols of connecting tools [2] ................................. 12
3.8 Pool with two swimlanes ....................................... 12
3.9 Data object and data store ...................................... 13
3.10 Organization theorem [3] ....................................... 15
3.12 The basic transaction pattern [4] ............................... 17
3.14 The complete transaction pattern [5] ......................... 19
3.15 The distinction axiom [6] ..................................... 20
3.16 DEMO XML structure ........................................ 21

4.1 Demoworld.nl Drawing pallet [7] ............................... 26
4.2 Camunda modeler workspace ................................. 27
4.3 Launching child transactions by using counter [5] .......... 29
4.4 Revokes in BPMN notation .................................. 30
4.5 All types of transaction in BPMN notation without revokes .... 31
4.6 OCD of Bakery model ....................................... 32
4.7 PSD of Bakery model ...................................... 32
4.8 Happy-flow diagram in BPMN ................................. 33

6.1 Camunda Enterprise ........................................... 38

B.1 Whole transformed model part 1 ............................... 48
B.2 Whole transformed model part 2 ............................... 49
## List of source codes

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transactions kinds</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>Composite actor roles</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>Elementary actor roles</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>Entity types</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>Connections</td>
<td>24</td>
</tr>
</tbody>
</table>
While businesses are expanding, they found the need for more effective inside business processes. One approach to help make these processes more effective is to implement business process management methods. Business process management needs to describe these processes. Representation of these processes needs to be precise and understandable.

There are two main approaches to describing these processes. First one is using DEMO models. These models are very precise and complete – they describe every event, which can happen during process. The main disadvantage is that they are hard to read. Second approach are models in BPMN notation. Those models are easy to understand but they do not show every possible event, which can happen in process.

This thesis are trying to describe and implement automatic transformation from DEMO models to BPMN by approach described in other thesis. Another motivation for this thesis was to show complex DEMO models in more human readable form of BPMN notation for description purposes.

In the first part of thesis is theoretical description of Business process modeling and introduction of both models. First description of BPMN notation and than DEMO models.

In the second part we describe used transformation method used in our application and evaluation of its advantages and disadvantages.

In the last part we analyze the usability of this application in business environment.

This thesis are based on paper by Ondřej Mráz Converting DEMO PSI Transaction Pattern into BPMN: A Complete Method.
Chapter 2

Goals and methodology

2.1 Goals

The first goal of this thesis was to introduce myself to DEMO and BPMN notation, and to evaluate modeling tools for these notations.

The main goal of this thesis is to make transformation tool, which will be able to take DEMO model on input and be able to produce model in BPMN notation, by Mraz’s method.

Then try executability of the resulting models in Camunda Enterprise.

2.2 Methodology

In this section is described the methodology by which I analyzed the problem and get the results. The first problem was to find the right tools, which would suit our solution. I evaluate and describe the most used tools for this models. This description is written in another section.

Next we have to describe the right XML format, of these two notations, which is not standardized and would be suitable for selected tools.

After that, I made the transformation tool, which would transform DEMO models to BPMN notation as described by Ondřej Mráž in [5].

The last part describe the execution of resulting models and its usage in business process management.
3.1 Definition of basic concepts

In the first chapter are defined basic concepts needed to understand the rest of the thesis.

3.1.1 Business process

Business process have many different definitions. By intuitively it is activity, which leads to goal. More complex definition say

“A business process consists of a set of activities that are performed in coordination in an organization in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.” [8]

Business process describes:

- Which activities are performed during its execution.
- Which organization units are involved in process.
- What input and output data are used.
- Which events and problems can happened during process.

3.1.2 Business process management

Business process management can be defined as follows: “Business Process Management (BPM) is a management approach used for describing, controlling, modeling, and optimizing business processes. Business Process Management builds a bridge between various business processes and forms the basis
3. Review

for organizational and information-technological initiatives aiming at improving the value-added chain within companies and between companies and their suppliers. [9]

The goal of business process management is not only the process management but to continuously improve processes and from this perspective it can be considered as the “process optimization process.”

Individual goal, which can be achieved with assistance of business process management are:

- Improve the alignment of all aspects of an organization with customer requirements
- Increase process transparency
- Promote business efficiency
- Maximally exploit employee potential
- Improve product quality
- Reduce costs [9]

Business process management is strongly oriented on improving quality of processes. But for improving process, we have to describe it. The ways in which processes can be described are discussed in following chapter.

3.1.3 Business process modeling

While working with processes it is very important to describe them, if possible in visual form, to be easily understood by a wide range of users. The same applies to changing or optimize processes, we need to describe it.

Business process modeling (BPM) deals with these issues. Process model can be described as follows:

“Business model describe, typically in graphic form, activities, events and its connections, that make up business process.” [10]

Process can be descried graphically or by text. Graphic description is more widely used, for its better clarity. I have briefly described flowcharts, the most basic graphical models, in next section. DEMO and BPMN are essential for this thesis therefore there is chapter for each of them.

3.1.3.1 Flowcharts

Flowcharts are the most basic visual models for description of processes. They are often used in IT for description of algorithms. They are popular mainly for their simplicity and intelligibility.
3.1. Definition of basic concepts

Their main disadvantages are summarized in:
They are not suitable for description of complex processes, because the resulting model would be excessive and confusing. Another drawback is excessive simplification of process and problematic transformation of existing models. [11]

Basic elements of flowcharts by:

- Start and end symbols
- Arrows – Defining direction of the flow in chart
- Steps of process – representation of individual activities
- Decision – division of process by condition
- Inputs and outputs [1]

Example of the basic flowchart on (fig. 3.1).

Figure 3.1: Basic flow chart [1]
3.2 BPMN Notation

3.2.1 Introduction to BPMN notation

BPMN (Business Process Model and Notation) is graphical notation used for description of business processes, developed by OMG (Object Management Group) organization. Their motivation was to develop notation, which is understandable for all from managers, through analysts to ordinary users:

"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."[12]

Nowadays the BPMN notation is probably the most used notation used for modeling of business processes. Its main strengths include standardization and possibility to automatize models. Plus there are many useful tools for modeling and working with BPMN notation. Reading the models and understanding them is easy, because of similarity to basic flowcharts.

Main disadvantage of BPMN is, that it is only notation for describing individual elements a rules for its usage. There is no methodology, which would specify the procedure for making these models proper – valid, complete and clear.

3.2.2 Types of diagrams

There are three types of process diagrams. The most used one is Collaboration diagram, which is presented since first version of BPMN. Choreography and Conversation diagrams were added later in BPMN 2.0. and they are not that used as Collaboration one.

3.2.2.1 Collaboration diagram

Collaboration diagram is to capture flow of the process. It is mostly composed of two or more pools, which represent participants and their communication inside the process. Processes can be divided into two groups, private and public. Private processes take place inside one pool and only the messages can cross border of this pool. Public processes show communication between private processes and another participant or process. There are only activities, which have purpose of communication.

3.2.2.2 Choreography diagram

Choreography diagram is focused on interaction between individual participants of one process. It can be used for analysis of their communication or
improvement of their coordination. Because of its purpose to show communication between two and more actors, it have to exist between an least two pools. Every activity in diagram represent communication between two or more participants of the process.

3.2.2.3 Conversation diagram

Conversation diagram is simplified version of collaboration diagram. It presents, when the process participants communication with each other. There are used graphical elements not used in the other two – Conversation Node and Conversation Link.

3.2.3 Basic elements

BPMN contains many elements, in this chapter are presented only those, which are important for this work. The description is based on [12]. All elements are same for all three types of BPMN diagrams (described above in this thesis), although their usage is little different.

3.2.3.1 Activities

Activity symbolize on unit of work. It occurs in two forms:

- **Task** – is atomic activity, which can not be further divided.
- **Subprocess** – is activity type represents separated process, which is shown as one element, for better clarity. It contains process, which has its own starting and ending event, and another activities.

Symbols used for representation of the activities are shown below on (fig. 3.2).

![Symbols of activities – Task and Subprocess](image)

3.2.3.2 Events

Events, as the name suggests, represent events, which can occur during the process. They can influence the process in many different ways. There are three types of events:
3. Review

- Starting events – They represent start of the process. There do not have to be a starting event, but if the process has an ending event, then it has to have a starting event. There are five types of starting events. The most important for this work are shown below (fig. 3.3). The most important is the empty starting event, for which there is no defined event, which would lead to its start.

Figure 3.3: Types of starting events

- Intermediate events – Events occurring during process. They can be divided into two categories:
  - Intermediate throwing events – Throwing events initiate some action.
  - Intermediate catching events – Catching events are awaiting some action.

There exist many types of these events, but for this thesis, only two are the most important. Message type, which is used for sending or receiving a message. This type of communication can be used only between pools. Signal type is similar to message type, except the signal has no intended receiver.

Figure 3.4: Types of intermediate events

- Ending events – They represent the end of the process. It has the same rules as starting events, there can be many ending events or none. There are nine kinds of ending events. For this thesis, only two are important. Empty ending event, when the process ends without initiation of action, and message ending event, which sends a message to another process after ending. Both of them are shown below (fig. 3.5).

3.2.3.3 Gateways

Gateways are used for changing the flow of process. It is the only element with the ability to fork or to merge the flows of the process, based on some condition. Gateways have many kinds, which are used:
3.2. BPMN Notation

Figure 3.5: Ending events

- **Exclusive** – Divide flow of the process into many alternative paths, and based on condition decide, on which path will the flow continue. There have to be chosen only one path.

- **Inclusive** – They are quite similar as exclusive gateways, but they choose more then one path for the flow.

- **Parallel** – Parallel gateways are used to create parallel flows and their synchronization. Incoming flow is divided, without any condition, into more flows. If parallel gateways is used for merging of the flows, the process waits for all flows, to come, before it can continue.

- **Complex** – Complex gateways are used for dividing and merging of flows based on complex condition, which can not be done with other types of gateways.

- **Event based** – Path selection is based on the event, which occurred during process. Usually used in combination with intermediate message event.

All gateways are shown below on (fig. 3.6).

Figure 3.6: Symbols of gateways

3.2.3.4 Connecting objects

Connecting objects are connecting other elements, like activities or gateways. There are four types, all shown on (fig. 3.7):

- **Flow** – It determines, in what order the flow flows. Its source and target have to be activity, event or gateway. Sequential flow from a pool must not interfere with another pool, same as it must not cross the border of subprocess.

- **Conditional flow** – It is used, when connecting, between two elements, depends on condition.
3. Review

- Message flow – Represents communication between sender and receiver of a message. Flow of the message must not be used for communication between entities in same pool.

- Association – Its purpose is to connect artifacts (groups, annotations) with other elements in process.

![Symbols of connecting tools](image)

Figure 3.7: Symbols of connecting tools [2]

3.2.3.5 Swimlanes and pools

Swimlanes and pools are used for separation of process actors and partition of tasks into units. Representation of these elements is shown on (fig. 3.8).

- Swimlane – Usage of swimlanes in BPMN is precisely defined. Usually they are used for distinction between actors in pool and clearer arrangement of activities. For example they can represent individual participants of the process.

- Pool – Pool determines boundaries of process. Its boundaries can only be crossed by message flow. It can shown model of process, white box, or it can be shown without details as black box. Pool can have one or more swimlane.

![Pool with two swimlanes](image)

Figure 3.8: Pool with two swimlanes
3.2.3.6 Data

Data element is used for graphical representation of data. There are two types, both shown on (fig. 3.9):

- Data object – Data object symbolize data, which exist temporarily in instance of process. It can be seen in two types, the data input and data output.
- Data store – Data store represent data, which are available permanently, in database for example.

Data elements are connected to other elements by data association.

![Data object and data store](image)

Figure 3.9: Data object and data store

3.2.3.7 Artifact

Artifact are used for adding other informations to a model. There are used two types of artifacts:

- Annotations – Annotations allow you to describe the business process and flow objects in more detail. It makes BPMN models more readable.
- Groups – Groups organize tasks or processes that have significance in the overall process.

3.3 DEMO Models

In the first section of this chapter we describe DEMO methodology (*Design and Engineering Methodology for Organizations*), motivation for its design, theoretical basics and description of models, with which we can make it.

In the second section we describe XML format, in which the DEMO models are written.

3.3.1 DEMO notation

3.3.1.1 Motivation and corporate ontology

DEMO is methodology created for modeling and to analyze business processes, made by prof. Jan Dietz. Motivation for its creation was, by [13], increasing
3. Review

complexity of businesses and insufficiency of current modeling models. Communication and interaction between process participant is quite important, but there was not a modeling technique, sufficient enough, to capture them on required level. As stated in [14], DEMO is able to show view on the essence of businesses without realization and implementation, the ontological view – Enterprise Ontology.

The goal of enterprise ontology is to make models of organizations, which are able to provide comprehensive view on its internal functions regardless of implementation. These models should fulfill these five conditions called C4E, by [6]:

1. Uniqueness – While modeling process, we always get the same model.
2. Completeness – Resulting model is complete, with all mandatory parts.
3. Consistency – It is clear, how particular steps follow.
4. Conciseness – Model is compact and does not contain redundant parts.
5. Essentiality – Model show only the essence of business, without the implementation.

3.3.1.2 PSI Theory

The $\psi$ theory is, by [13], theoretical foundation, on which the DEMO stands. It is theory consisting of four axioms and one theorem. The goal of this theory is to show essence of business freed from inside implementation. In this DEMO and PSI theory have common goal.

3.3.1.2.1 Organization theorem

The organization theorem state, that organization is heterogeneous system consisting of three homogeneous systems:

- B-organization Business
- I-organization Intellect
- D-organization Document

These three are dependent: D-organization support I-organization and it support B-organization. Representation of these connections can be see (fig. 3.10). For DEMO modeling the B-organization is the most important, which is located on top of the pyramid. This level is the highest and provides complete knowledge of essence of the business, without implementation and realization.
3.3. DEMO Models

3.3.1.2.2 Operation axiom

First axiom tells us that the implementation independent essence of an organization consists of activities done by actors. How are these activities connected and, in which order are they done, is described in operation axiom. It also tells us, that all coordination acts are done in transaction.

Actors, by their action, perform coordination and production acts, which produce coordination and production facts.

Coordination act (C-act) produce coordination fact (C-fact). This action is done by one actor, the performer, and is addressed to another actor, the addressee. Actors communicate, with help of these acts, about the resulting production fact. All C-facts make up the C-world.

Production acts (P-acts) lead to production of the resulting P-facts, which are called P-world. P-acts and P-facts produce new, tangible or intangible, products. The P-fact starts to exist, just after the performance of C-acts, state and accept. Graphical representation of operation axiom is shown below on (fig. 3.11).

![Graphical representation of the operation axiom](image-url)
3. Review

3.3.1.2.3 Transaction axiom

Transaction axiom describes, how are actions, from operation axiom, connected and in which patterns they are used. This axiom tells us that all coordination acts are performed in universal pattern, called transaction.

Every transaction is consists of two conversations, which is perceived as sequence of C-acts between two actors with goal to produce P-fact.

Transaction is between two actors, the first actor Initiator request a product and the second actor Executor, who produce the product.

Whole transaction can be divided into three parts:

- **Order phase** – Actors discuss what and when to produce it.
- **Execution phase** – Executor makes product.
- **Result phase** – Actors discuss if the result is as expected, after that is accepted. After this phase is finished the result starts to exist by PSI theory.

As stated above, transaction are done in universal pattern. This pattern is called primary transaction pattern, its extension is called standard transaction pattern, and the most complex is complete transaction pattern.

3.3.1.2.3.1 Basic transaction pattern

For basic description of transaction is used basic transaction pattern, which shows only ideal run of the transaction without any revokes.

Transaction process is here described on an example from bakery: Customer comes to bakery, and orders a bread. Baker agrees and sells the bread to the customer. Then the customer leaves the bakery.

Transaction pattern describe sequence of C-(f)acts leading to producing of P-(f)act:

- Request – Initiator requests the creation of P-fact. It is request for a bread, in our example.
- Promise – Executor promises creation of the P-fact. Baker agrees.
- State – Executor states, that the product is done. Baker tells the customer, that bread is ready.
- Accept – Initiator accepts product. Costumer is happy with the outcome.
3.3. DEMO Models

Process of the basic transaction pattern is shown on (fig. 3.12). C-acts are represented as white squares and C-facst as white circles. The gray square is P-act, rhombus is P-fact. There is shown both actors and at same time, in which phase, the transaction parts, belongs to.

![Figure 3.12: The basic transaction pattern](image)

3.3.1.2.3.2 Standard transaction pattern

In the standard transaction pattern is possible to show decline of request by executor and state by initiator.

Declination of a request leads to declined state. In this state, the initiator decide, if quit whole transaction or create a new request. From the example: The baker announce, that he can not bake this bread and the customer decide if he walks away or orders another type of good.

Declination of a state leads to rejected state. Executor decide if the dec- lination was rightful, and either stop the transaction or get back to the state accordingly. From the example: The state reject would appear if the customer would not be satisfied with the bread. The baker would decide if it is rightful, and either stop the transaction or not.

Standard transaction pattern is shown below (fig. 3.13), description of the pattern is same as above.
3. Review

3.3.1.2.3.3 Complete transaction pattern

The last is complete transaction pattern, which adds possibility to declina-
tion of any completed C-fact. This declination is done by revokes. The revoke
can happen, only if both actors agree on it. Described on our example:

- Request revoke – Customer announced, what kind of bread he wanted,
  but then he changed his order.

- Promise revoke – Baker promised to bake the bread, but find out, he
does not have the right type of flour.

- State revoke – Baker announced, than bread is done, but then found
  that it is burned.

- Accept revoke – At home, customer found that the bread was burned,
  so he decided to return it.

Complete transaction pattern shown below on (fig. 3.14).

Figure 3.13: The standard pattern of a transaction [5]
3.3. DEMO Models

3.3.1.2.4 Composite axiom

Composite axiom tells us, how are transaction connected and dependent. That every transaction is enclosed in some other transaction, customers transaction or self-activated.

According to Dietz this axiom provides the basis for a well-founded definition of the notion of business process. Which states:

“\textit{A business process is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment (external activation) or a request by an internal actor role to itself (self-activation).}” [6]

3.3.1.2.5 Distinction axiom

Distinction axiom is based on the idea, that the acts of actors include three human abilities, called performa, informa and forma. The highest level are performa activities, which goal is to produce new things. To perform the performa activity, it usually takes several informa activities, which contain, what is the content of the information. Informa activities are above the forma activities, which contains in what ways are information preserved and transmitted. Only the performa activities are shown in process models.

Distinction axiom is illustrated on (fig. 3.15).
3.3.2 DEMO XML format

In the first part of the section we show diagram which show how DEMO are divided in XML format. In the second part are samples of the code for individual elements.

3.3.2.1 XML diagram

In this section is shown structured of DEMO model saved in XML format by VISI Standard.
3.3. DEMO Models

Figure 3.16: DEMO XML structure
3. Review

3.3.2.2 Basic elements

In this section are examples of basic DEMO elements in VISI Standard, with commentary.

3.3.2.2.1 Transactions kinds

```xml
<TransactionKinds>
  <!-- Unique ID of transactions kind -->
  <TransactionKind Id="8d5413e0">
    <!-- Unique Identification -->
    <Identification>T01</Identification>
    <!-- Unique name -->
    <Name>completing</Name>
    <!-- Transaction sort -->
    <TransactionSort>original</TransactionSort>
  </TransactionKind>

  <TransactionKind Id="e0c5fb1e">
    <Identification>T02</Identification>
    <Name>baking</Name>
    <TransactionSort>original</TransactionSort>
  </TransactionKind>
</TransactionKinds>
```

Code listing 1: Transactions kinds

3.3.2.2.2 Composite actor roles

```xml
<CompositeActorRoles>
  <CompositeActorRole Id="d322c9c0">
    <Identification>CA01</Identification>
    <Name>customer</Name>
  </CompositeActorRole>
</CompositeActorRoles>
```

Code listing 2: Composite actor roles
3.3.2.2.3 Elementary actor roles

```xml
<ElementaryActorRoles>
  <ElementaryActorRole Id="5f93420b">
    <Identification>A01</Identification>
    <Name>completer</Name>
  </ElementaryActorRole>
  <ElementaryActorRole Id="22c6b7df">
    <Identification>A02</Identification>
    <Name>baker</Name>
  </ElementaryActorRole>
</ElementaryActorRoles>
```

Code listing 3: Elementary actor roles

3.3.2.2.4 Entity types

```xml
<EntityTypes>
  <EntityType Id="f21b1d42">
    <Name>pizza</Name>
  </EntityType>
  <EntityType Id="65abdfba">
    <Name>customer</Name>
  </EntityType>
</EntityTypes>
```

Code listing 4: Entity types
3. Review

3.3.2.2.5 Connections

```xml
<Connections>
  <Connection Id="7e67fd26">
    <!-- Connection type -->
    <InitiatorCAR>
      <!-- Unique ID of starting element -->
      <FromCompositeActorRole>
        d322c9c0</FromCompositeActorRole>
      <!-- Unique ID of ending element -->
      <ToTransactionKind>8d5413e0</ToTransactionKind>
    </InitiatorCAR>
  </Connection>
  <Connection Id="8bb0aea6">
    <WaitConditionTPSK>
      <FromTransactionProcessStepKind>
        729cc896</FromTransactionProcessStepKind>
      <ToTransactionProcessStepKind>
        468dee3f</ToTransactionProcessStepKind>
    </WaitConditionTPSK>
  </Connection>
</Connections>
```

Code listing 5: Connections
4.1 Modeling tools

In this chapter we discuss modeling tools for both the DEMO models and the BPMN notation.

4.1.1 DEMO modeling tools

DEMO models are not yet widely used for business process models. So it is quite hard to find good modeling tools. We have listed two of them and discussed their usability.

4.1.1.1 Demoworld.nl

Demoworld is on-line modeling tools made for modeling and simulating DEMO models. It is easy to use as it has drawing palette (fig. 4.1) with all elements used in DEMO models.

As mentioned earlier, Demoworld let you simulate your model and animate processes inside the model.

The main drawback of this tool for this thesis, was its inability to export DEMO models to XML format, which was crucial for us.
4. Analysis

4.1.1.2 Plena

Plena modeling tools, by TEEC2 [15], is in form of Enterprise Architect plugin. It works almost the same as the Demoworld.nl tool but it comes with possibility to export in XML format.

Its only disadvantage is the necessity to have license for Enterprise Architect, which is paid.

4.1.1.3 Summary

For this work we have chosen the PLENA tool, for its main advantages of exporting DEMO models in XML format. But for normal user the Demoworld seems as the best choice, for its ability to simulate DEMO models.
4.1.2 BPMN modeling tools

There are many tools for BPMN notation for its use in Enterprise engineering and its certifications.

We picked two tools for discussion and compare their usability.

4.1.2.1 Camunda modeler

Camunda modeler has many great features. This modeler is free to use and is quite intuitive, when it comes to modeling.

It has drawing pallet and it automatically make XML format document from your drawing. Plus it is possible to insert plain XML format text and it will draw your BPMN model.

Another advantages of this modeler is the possibility to model another models apart from BPMN, as are DMN and CMMN.

The main reason, for choosing this tool, was another tool from Camunda, the Camunda Enterprise, which has the possibility to simulate models.

![Camunda modeler workspace](image)

Figure 4.2: Camunda modeler workspace

4.1.2.2 bpmn.io

Bpmn.io is on-line modeler based developed by Camunda developers. It is basically the same engine as Camunda, so it has same advantages as Camunda.

4.1.2.3 Summary

For this work we have chosen the Camunda desktop modeler, for its usability without the internet. But both modelers are equally good.
4. Analysis

4.2 Mraz’s transformation approach

For the main part, these thesis are based on paper *Converting DEMO PSI Transaction Pattern into BPMN: A Complete Method* [5] by Ondřej Mráz.

4.2.1 Transformation method

In this section is summarized the approach we used for automatic transformation.

4.2.1.1 C-acts

C-acts are essentially activities that take place in order specified by the transaction pattern. BPMN has the concept of activities and the order is specified by sequence flows. As C-acts are atomic, the appropriate activity type is *task*.

4.2.1.2 C-facts

C-fact becomes existent in the world as a consequence of performing a C-act. In Mraz’s paper they use the BPMN 2.0 data store, for C-facts representation. Into this data store is stored the state of the transaction. This data store is connected to every C-act activity by an association.

4.2.1.3 P-(f)acts

According to the PSI-theory, the P-fact starts to exist based on acceptance of the product, so P-(f)acts can be expressed by an activity only. For optimization of implementation, we store them similarly to the C-facts described above.

4.2.1.4 Actors

We used swimlanes for representation of actors from DEMO models because they are isomorphic by [16].

4.2.1.5 The Composition Axiom

In his paper they discussed two approaches, from which we decided to use the one, where for every transaction there will be one whole diagram. On one hand this approach leads to huge diagrams but on the other hand its much easier to use in automatic transformation.

This approach consists of two part:

1. **Launching a child transaction** in specific place in the parent transaction. The child transaction must be started just after creating a specific
4.2. Mraz’s transformation approach

C-fact. We have decided to use cycle counter for creating child transactions. Modeling by cycle (fig. 4.3) means, that the model contains an activity counting, how many times the activity was run. After this activity there is a gateway. If the counter has not reached the number of child transaction instances to spawn, the process goes into message throwing event to start a child transaction instance and then the process returns to the counting activity. This happens 0..N times, as required.

2. **Blocking execution of the parent process** until the child process has not reached the given state (creating a C-fact being waited on). This blocking can be realized by BPMN catching event condition in the parent process waiting for a specific condition before the given C-act. Here, a conditional event must be used instead of a signal event, as we do not wait just for a signal, but for a specific instance in case of multiple child transaction instances.

We used message connections between transactions as suggested in Mraz’s paper. See below (fig. 4.3).

![Figure 4.3: Launching child transactions by using counter [5]](image)

4.2.1.6 Revokes

A revoke works in the following steps according to the transaction pattern. First, the revoking actor asks the other actor for granting the revoke. The other actor allows or refuses. If the revoke is allowed, the main process returns to the appropriate state. We model this by using simple BPMN subprocess with a set of appropriate activities (fig. 4.4).

4.2.1.7 The resulting BPMN model

In contrast to Mraz’s paper, we had to change the resulting model. Where the subprocesses in revokes, were used, but this approach was not possible for us to replicate, so we used revokes without subprocesses. All transaction patterns without revokes are shown on (fig. 4.5).

As stated in Mraz’s paper, the resulting models, for processes with more then two transactions, are huge and quite hard to read by casual reader.
4. Analysis

Figure 4.4: Revokes in BPMN notation
4.2. Mraz’s transformation approach

Figure 4.5: All types of transaction in BPMN notation without revokes
4. Analysis

4.2.1.8 Example - Bakery

As an example for the demonstration of our method we used Bakery model quite similar to classic Case Volley example as seen in Mraz’s Paper. It was selected for its simplicity, yet including the substantial constructs. There are OCD (fig. 4.6) and PSD (fig. 4.7) diagrams of this example.

The process has two transactions and three actors. The full transformed BPMN model converted by the described method is in appendix of this thesis.

![OCD of Bakery model](figure4_6)

![PSD of Bakery model](figure4_7)

4.3 Transformation for happy-flow

In this section is described transformation, without transformation of revokes, data stores and declinations. Only happy-flow pattern of transactions is modeled in BPMN.

This section is outside the scope of this thesis, and should lead to future research in DEMO to BPMN transactions.

Below, on (fig. 4.8), is shown happy-flow pattern of the DEMO example described above. These models are fully executable.
4.3. Transformation for happy-flow

Figure 4.8: Happy-flow diagram in BPMN
Chapter 5

Design and implementation

In this chapter the design and implementation of transformation tool is described.

5.1 Transformation tool

In this chapter is described the transaction tool made for this thesis and discussion about its usability.

5.1.1 Tool

The main tool is made in Python, the programing language. This language was chosen because of its clear code and speed.

The tools consists of two part, which are described below.

5.1.1.1 DEMO model processing

The first part of the tool, is used for parsing of DEMO models in its XML form, which was described above.

The most interesting thing is, that it is the first tool ever made for parsing DEMO models by VISI standard. Every part of DEMO model is easily imported from this part, which is useful for future research in DEMO modeling.

5.1.1.2 Transformation

In the second part, the parsed elements are assigned to its counterparts from BPMN model, which was described above.

In the last parts, the main transformation process happens. The assigned elements are converted into new XML file, which is in BPMN format. This parts was the hardest, because of the XML file for BPMN has hard-coded positions of the elements in the models.
In this chapter is described the testing of result models and its usability in business process management.

6.1 Testing

This section describe the setting of resulting models and how to execute them.

6.1.1 Execution of resulting model

For the execution of resulting model was used Camunda Enterprise, which allows execution of BPMN models modeled in Camunda modeler.

6.1.1.1 Model preparation

The main transformation tools prepare XML file of BPMN model. This XML file can be easily opened in Camunda Modeler, which allows us to add all the things needed for execution of the model. To execute model in Camunda Enterprise tool, we need to:

- **Transactions** – set executable variable true or false, for every transaction we want to execute
- **Messages** – set messages to match its throw and catch events
- **Signals** – set signal name to match its throw and catch events
- **Exclusive gateways** – set string variable to be the answers to questions

6.1.1.2 Execution

The Camunda Enterprise tools was used for execution of transformed models. This tools is made up of many modules, which helps with BPMN models,
6. Evaluation

their execution, optimization and many more. Generated models shown in Camunda Enterprise (fig. 6.1).

![Camunda Enterprise](image)

Figure 6.1: Camunda Enterprise

This tool was chosen for its simplicity and availability.

The execution has three main parts:

1. Setting the generated model in Camunda modeler as described above.
2. Deployment of the model to the Cockpit, Camunda Enterprise module.
3. Execution of the model in the Cockpit

It was tested in this work, that the generated models are executable after setting them in Camunda Modeler. It would be interesting in future research to generate executable models right away.

6.2 The solution in the context of BPM

The possibility to execute the resulting models is important for business process management, because the processes, in resulting models, can be evaluated for its effectiveness and changed accordingly.

Camunda enterprise provide useful tools for finding not effective parts of process and offers ways how to change them.
Chapter 7

Contribution

7.1 Usefulness in BPM

Both of these notations, DEMO and BPMN, are useful in business process management and other levels of management occurring in organizational structures. It provides human readable charts and models, which help capture all levels of processes, from the low level ones, like filling application and sending it to another person, to high level, which capture functioning of the company as a whole.

These models are also useful for project managers, who use them to capture processes while designing new features in old projects or while trying to improve old processes in a company. These models provide better view on business processes, which helps find sections of the process, which are problematic or not effective.

7.2 Future research

This thesis showed first automatic transformation of DEMO models to BPMN notation. The Mráz’s method was used for its suitability for this problem, but this method has many disadvantages, which were mentioned above.

Future research should focus on other methods of transformation, alike the happy-flow transformation shown here. These types of transformations should be more human readable to be useful for business managers.

I will focus on these types of transformation in my future research.
Conclusion

The goal of this thesis was to get acquainted with DEMO models and BPMN notation, the modeling tools for modeling these models. Another goal was to review these tools, create tool for transaction DEMO models to BPMN models and to review created models for their usability.

In this thesis were shown two modeling tools for each model. Further there was described the Mraz’s method, for transformation of these models, on which was based the transformation tool made for this thesis.

Then the transformation tools is shown, and the result are described and reviewed. The was successfully made, so the goal of this thesis was fulfilled. Furthermore the provides possibility to transform DEMO model to happy-flow in BPMN notation, which was made beyond scope of the thesis assignment.

In would be interesting to made tools for transformation of DEMO models to more simpler BPMN models, in the future research. Parts of the tools could be used, mainly the part for processing DEMO models by VISI standard.
Bibliography


Acronyms

BPMN  Business Process Model and Notation
CMMN  Case Management Model and Notation
DEMO  Design and Engineering Methodology for Organizations
DMN   Decision Model and Notation
OMG   Object Management Group
PSD   Process Structure Diagram
PSI   Performance in Social Interaction
XML   Extensible markup language
Pictures of models

In this chapter are shown full transformed models.

B.1 Mraz’s transformation method

Below is shown full transformed model in BPMN notation using Mraz’s method.

B.2 Happy-flow method

Below is shown full transformed model in BPMN notation using happy-flow method.
Figure B.1: Whole transformed model part 1
Figure B.2: Whole transformed model part 2
B. Pictures of models

Figure B.3: Full Happy-flow diagram in BPMN part 1
Figure B.4: Full Happy-flow diagram in BPMN part 2
Appendix

Contents of enclosed CD

readme.txt...................... the file with CD contents description
src................................. the directory of source codes
thesis...................... the directory of $\LaTeX$ source codes of the thesis
transformingTool.............. the directory of Python source codes
text............................. the thesis text directory
thesis.pdf...................... the thesis text in PDF format
modelingTools........................ modeling tools used
camundamodeler.zip........... camunda modeler in zip compression