INVESTIGATING THE BENEFITS OF MODELING BUSINESS PROCESSES IN BPMN + DMN

Word Count: 10.686

Joachim Bossuyt
Student Number: 01105705

Supervisor: Prof. dr. Frederik Gailly

Master's Dissertation submitted to obtain the degree of:
Master of Science in Business Engineering

Academic Year: 2016 - 2017
INVESTIGATING THE BENEFITS OF MODELING BUSINESS PROCESSES IN BPMN + DMN

Word Count: 10,686

Joachim Bossuyt
Student Number: 01105705

Supervisor: Prof. dr. Frederik Gailly

Master's Dissertation submitted to obtain the degree of:

Master of Science in Business Engineering

Academic Year: 2016 - 2017
PERMISSION

I declare that the content of this Master's Dissertation can be consulted if all sources are mentioned. It may only be reproduced for personal use and/or academic purposes.

Signature

Joachim Bossuyt
PREFACE

This master’s dissertation is the final stage in obtaining the degree of Master of Science in Business Engineering: Operations Management from Ghent University. This paper is part of the UGent MIS Research Group.

First of all, I want to thank my supervisor, prof. dr. Frederik Gailly, who was always available to assist me during the last two years. His input was invaluable to the quality of my work. Furthermore, I also want to thank Steven Mertens for his important feedback during the final stages of this project.

Secondly, I want to thank my friends who I met before and during my time as a student at Ghent University and my partner, Jolien, as well as my student-colleagues of 180 Degrees Consulting, who all were a source of inspiration to me. It was a great pleasure to lead this amazing student organization, and I hope to remain part of it through its alumni network after my graduation.

Finally, I’d like to thank my family, and in particular my mother and father who kept supporting me in every way possible. I cannot describe how grateful I am for the sacrifices they made throughout their lives for me. Thank you for the opportunities you gave and keep giving me since the day I was born.

Ghent, 23 May 2017

Joachim Bossuyt
# TABLE OF CONTENTS

LIST OF ABBREVIATIONS .................................................................................. V
LIST OF FIGURES ............................................................................................ VI
LIST OF TABLES ............................................................................................... VII
LIST OF EQUATIONS ....................................................................................... VIII

INTRODUCTION ................................................................................................. 1

1. DECISION MAKING IN BPMN ..................................................................... 3
2. DECISION MODEL AND NOTATION ............................................................. 5
3. BPMN + DMN ............................................................................................... 7
   3.1 PROCESS CLASSIFICATION ................................................................. 7
   3.2 EXTRACTING DECISION LOGIC FROM THE PROCESS MODEL .......... 8
   3.3 BENEFITS OF BPMN + DMN ............................................................... 10

4. THE COMPLEXITY OF A BPMN MODEL ..................................................... 13
   4.1 COMPONENTS OF COMPLEXITY ......................................................... 16
      4.1.1 ACTIVITY COMPLEXITY ......................................................... 16
      4.1.2 CONTROL-FLOW COMPLEXITY ............................................. 16
      4.1.3 DATA-FLOW COMPLEXITY ..................................................... 22
      4.1.4 RESOURCE COMPLEXITY ....................................................... 26
   4.2 QUALITY DOMAINS .............................................................................. 26
      4.2.1 COUPLING .................................................................................. 27
      4.2.2 COHESION .................................................................................. 29
      4.2.3 MODULARITY ........................................................................... 29
   4.3 CONCLUSION ......................................................................................... 30

5. REDUCING COMPLEXITY BY USING BPMN + DMN ................................. 32
   5.1 CASE STUDY ......................................................................................... 32
      5.1.1 BPMN MODEL ........................................................................... 32
5.1.2 BPMN + DMN MODEL ........................................................................................................ 35
5.2 FINDINGS .................................................................................................................................. 38
6. CONCLUSION ............................................................................................................................... 42
   6.1 CONTRIBUTIONS ...................................................................................................................... 42
   6.2 LIMITATIONS AND FUTURE RESEARCH ............................................................................ 43
REFERENCES ................................................................................................................................. IX
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>BahnCard</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Model and Notation</td>
</tr>
<tr>
<td>CFC</td>
<td>Control-Flow Complexity</td>
</tr>
<tr>
<td>DF</td>
<td>Data-Flow</td>
</tr>
<tr>
<td>DMN</td>
<td>Decision Model and Notation</td>
</tr>
<tr>
<td>DRD</td>
<td>Decision Requirements Diagram</td>
</tr>
<tr>
<td>HKM</td>
<td>Henry and Kafura’s Modularity</td>
</tr>
<tr>
<td>HR</td>
<td>Human Resources</td>
</tr>
<tr>
<td>IS</td>
<td>Information Systems</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>LoC</td>
<td>Lines of Code</td>
</tr>
<tr>
<td>ND</td>
<td>Nesting Depth</td>
</tr>
<tr>
<td>NOA</td>
<td>Number of Activities</td>
</tr>
<tr>
<td>NOAC</td>
<td>Number of Activities and Control-Flow Elements</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1: Spaghetti-like business process model illustrating the misuse of BPMN. Adapted from Batoulis et al. (2015). ................................................................. 3

Figure 2: The business process management spectrum. Reprinted from Di Ciccio et al. (2015). ................................................................. 7

Figure 3: Loan Application (BPMN). Adapted from Object Management Group (2015). ....................................................................................... 8

Figure 4: Loan Application (BPMN + DMN). Reprinted from Object Management Group (2015). ................................................................. 10

Figure 5: Loan Application (BPMN). Adapted from Cardoso (2008). ................. 15

Figure 6: Business process model (BPMN) of the loyalty program with detailed decision logic. Adapted from Batoulis & Weske (2017). ....................................................... 34

Figure 7: Business process model (BPMN + DMN) of the loyalty program. Adapted from Batoulis & Weske (2017). ........................................................................... 36

Figure 8: Decision requirements diagram of the loyalty program. Adapted from Batoulis & Weske (2017). ........................................................................... 37
LIST OF TABLES

Table 1: DRD Components. Reprinted from Object Management Group (2015)........ 6
Table 2: The control-flow complexity of the Loan Application process. Adapted from Cardoso (2008)............................................................................................................ 18
Table 3: The maximum and mean nesting depth of the Loan Application process... 20
Table 4: Cognitive weights for BPM elements. Adapted from Gruhn & Laue (2006). 21
Table 5: The coupled activities in the Loan Application process................................ 28
Table 6: Decision logic level: a decision table representing the discount rules. Adapted from Batoulis & Weske (2017)............................................................................................................ 37
Table 7: The coupled activities in the process model (BPMN)................................ 39
Table 8: The coupled activities in the process model (BPMN + DMN)................ 39
Table 9: Measuring the complexity for both business process models (BPMN vs. BPMN + DMN). .................................................................................................................. 40
LIST OF EQUATIONS


INTRODUCTION

Lately, designers more and more mistakenly incorporate detailed decision logic in business process models. Business Process Model and Notation (BPMN), a standard for business process modelling introduced by The Object Management Group (2011), was never designed to extensively model decision logic. However, businesses can benefit greatly from modelling their decision logic within their business processes, yet there was no clear method developed to do this (Batoulis, Bazhenova, Decker, Meyer, & Weske, 2015). The Object Management Group (2015) filled this gap by introducing a new modelling language, Decision Model and Notation (DMN), which allows designers to model decision logic independently, or in combination with the already established standard BPMN.

In this paper, we will investigate the benefits of using a combination of the modelling languages BPMN and DMN to model both business processes as well as decision logic. More specifically, the existing research suggests that BPMN + DMN will lead to a separation of concerns by extracting the decision logic from the process model. This principle is lend from the software domain, and in this context it means that the decision model is deemed irrelevant when looking from the process model's point of view, and vice versa. The hypothesis is that this will further lead to reduced complexity in the business process model, which will be the focus of this master’s dissertation (Batoulis et al., 2015; Biard, Bigand, Bourey, & Le Mauff, 2015; Taylor, Fish, Vanthienen, & Vincent, 2013).

This paper is structured as follows. In Section 1, we discuss how designers (wrongly) incorporate decision logic into the business process model and the consequences that come with it. Section 2 introduces the most important aspects of the new decision modelling language, DMN, published by the Object Management Group in 2015. In Section 3, we present the notions of modelling in BPMN + DMN. Afterwards, we look into the benefits of BPMN + DMN by means of a literature review. One of the suggested advantages is using a combination of BPMN + DMN to achieve a separation of concerns, which in turn would lead to reduced complexity in the process model. In Section 4, it is then investigated how the complexity of a business process model can be measured. Additionally, we adapt an existing metric from the software domain to
measure the data-flow complexity of a process model. Furthermore, the discussed metrics will be illustrated on a running example to clarify their usage. In Section 5, a practical case-study is presented to discuss the complexity reduction when transforming the process model from BPMN to BPMN + DMN. Finally, in Section 6, conclusions, limitations and suggestions for future research are formulated.
1. DECISION MAKING IN BPMN

In BPMN most decisions are modelled as gateways. When more rules are incorporated in the model, the complexity of the model increases very rapidly. Many players in industries like banking, health care and insurance incorporate decision logic directly into the process model. BPMN, however, was never designed to represent decision logic in such detail, and misusing it like this generally leads to very complex, spaghetti-like models, as depicted in Figure 1 (Batoulis et al., 2015). Increasing complexity leads, in turn, to hidden mistakes and errors by the process users (Repa & Zeleznsik, 2014).

![Spaghetti-like business process model](image)

Figure 1: Spaghetti-like business process model illustrating the misuse of BPMN. Adapted from Batoulis et al. (2015).

However, according to IBM, companies can benefit greatly from integrating decision logic in their business processes, since it may lead to the following competitive advantages (Underdahl, 2011):

- It guides employees through the decision-making process.
- It allows automating routing and processing of tasks, reducing the amount of human intervention, and consequently reducing (human) errors.
• It can prevent compliance pitfalls (e.g. inaccurate reporting, unallowable costs, misallocation of costs, etc.)
• It allows firms to respond rapidly to changing environments.

Designers thus started wrongly incorporating decision logic into the process model, because there existed no standard to model decisions yet (Taylor et al., 2013). The Object Management Group filled this gap by introducing a new modelling language in 2015, Decision Model and Notation (DMN). The purpose of DMN is to become a common notation (just like BPMN) that is understandable by all business users, so that it will ensure interchangeability of decision and process models across organizations.
2. DECISION MODEL AND NOTATION

Since DMN is a fairly new language, we will first give a brief overview of the elements used to model a decision. For a detailed description, we refer the reader to the DMN specification (Object Management Group, 2015).

A decision model, or decision requirements diagram can be modelled using a combination of four “elements” and three “requirements”. The Object Management Group (2015) summarized these elements and requirements in the table below:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision</td>
<td>A decision denotes the act of determining an output from a number of inputs, using decision logic which may reference one or more business knowledge models.</td>
<td>Decision</td>
</tr>
<tr>
<td>Business Knowledge Model</td>
<td>A business knowledge model denotes a function encapsulating business knowledge (e.g., as business rules, a decision table, or an analytic model).</td>
<td>Business knowledge</td>
</tr>
<tr>
<td>Input Data</td>
<td>An input data element denotes information used as an input by one or more decisions. When enclosed within a knowledge model, it denotes the parameters to the knowledge model.</td>
<td>Input data</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
<td>Notation</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Knowledge Source</td>
<td>A knowledge source denotes an authority for a business knowledge model or decision.</td>
<td><img src="image" alt="Knowledge source" /></td>
</tr>
<tr>
<td>Information Requirement</td>
<td>An information requirement denotes input data or a decision output being used as one of the inputs of a decision.</td>
<td><img src="image" alt="Information requirement" /></td>
</tr>
<tr>
<td>Knowledge Requirement</td>
<td>A knowledge requirement denotes the invocation of a business knowledge model.</td>
<td><img src="image" alt="Knowledge requirement" /></td>
</tr>
<tr>
<td>Authority Requirement</td>
<td>An authority requirement denotes the dependence of a DRD element on another DRD element that acts as a source of guidance or knowledge.</td>
<td><img src="image" alt="Authority requirement" /></td>
</tr>
</tbody>
</table>

3. BPMN + DMN

3.1 PROCESS CLASSIFICATION

Before introducing BPMN + DMN, it is important to acknowledge that different types of business processes exist. The reasoning here is the possibility that BPMN + DMN might not be useful for all types of processes.

Di Cicco, Marrella, & Russo (2015) classify business processes according to their degree of structure and predictability in the so-called business process management spectrum. They define five types of processes: (1) structured processes, (2) structured processes with ad hoc exceptions, (3) unstructured processes with pre-defined segments, (4) loosely structured processes, and (5) unstructured processes, as depicted in Figure 2.

![Figure 2: The business process management spectrum. Reprinted from Di Cicco et al. (2015).]
BPMN mainly targets structured processes with or without ad hoc exceptions. These type of processes are highly predictable and repeatable, with low flexibility requirements (e.g. administrative or manufacturing processes) (Mertens, Gailly, & Poels, 2015). Ad hoc exceptions are external events (e.g. order cancellations, payment errors, etc.) that cause deviations from the pre-defined process. We believe that BPMN might also be of value for unstructured processes with pre-defined segments, since such a segment falls de facto in one of the first two categories (i.e. structured process with or without ad hoc exceptions). BPMN, however, will not be able to grasp the entire (unstructured) process. Finally, the loosely structured and unstructured processes are beyond the scope of BPMN. In such processes the execution order and possibly even the different activities are not defined yet. These two types belong to the knowledge-intensive side of the business process spectrum and are nearly impossible to model correctly beforehand (Di Ciccio et al., 2015). It is, thus, our opinion that modelling in BPMN + DMN is only useful at the structured side of the business process management spectrum (i.e. processes with low flexibility, which are highly predictable and repeatable).

3.2 EXTRACTING DECISION LOGIC FROM THE PROCESS MODEL

The process model below (Figure 3) describes the business rules to follow with regards to a customer applying for a loan (i.e. a structured process). In Figure 4 the decision logic is extracted from the process model by replacing the activities and gateways by a single activity, the so-called Business Rule Task (“Decide Routing”). The process model is then complemented by a decision model to compensate for the missing information with regards to the decision (i.e. deciding on the loan).

![Loan Application (BPMN)](image)

Figure 3: Loan Application (BPMN). Adapted from Object Management Group (2015).
The model in Figure 4 is, in modelling theory, the correct way to model a process, while Figure 3 shows how designers wrongly incorporate decision logic in a process model (i.e. so-called gateway waterfall). The decision logic is represented in a DMN model on the decision requirements level. In this model it is clear that the “Routing”-decision is dependent on two sub-decisions (“Application Risk” and “Eligibility”). Both sub-decisions are using the input data “Application”, and are subject to the business knowledge “Application Risk Score Model” and “Application Risk Category Table”, and the “Eligibility Rules”, respectively. The latter are represented by a decision table on the decision logic level. Verhelst (1980) defines decision tables as follows: “A decision table is a table, representing the exhaustive set of mutual exclusive conditional expressions, within a predefined problem area.” Note that other ways to represent business rules may be used (i.e. binary decision trees, propositional rules, oblique rules, etc.). However, according to Huysmans, Dejaeger, Mues, Vanthienen, & Baesens’ (2011) research, decision tables are easier to understand and perform better with regards to accuracy, response time, and answer confidence for a set of problem-solving tasks. In this paper, we follow Huysmans et al.’s view and will thus make use of decision tables to represent the decision logic.
3.3 BENEFITS OF BPMN + DMN

The limited research that has been conducted since the release of DMN in 2015, unanimously concludes that extracting decision logic from the process model will lead to a separation of concerns (Batoulis et al., 2015; Biard et al., 2015; Taylor et al., 2013). This principle was introduced by Dijkstra (1982) in 1974. He writes: “This [separation of concerns] is what I mean by “focussing one’s attention upon some aspect”: it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect’s point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously.” (p. 61). The principle is primarily used in the software domain, where it refers to the fact that different aspects of functionality should not influence each other. In the context of BPMN + DMN, it means that the decision model is irrelevant when looking from the process model’s point of view, and vice versa.
Batoulis et al. (2015) write that process models with detailed decision logic result in complex, spaghetti-like models (see also Figure 1). In their research, it is proposed to separate decision logic from the process model to improve precision, readability and maintainability of both models. Moreover, it is assumed that the business process model would become less complex without detailed decision logic in it. Biard et al. (2015) reaffirm in their paper that gateways resulting from decision logic and multi-criteria decisions lead to labyrinth-like process models. They agree that separating the process from the decision logic would lead to a separation of concerns, which will further reduce complexity in the business process model. Taylor et al. (2013) also suggest that separating decision and process models would help manage complexity.

In this paper, we will primarily focus on this supposed complexity reduction as a result of the separation of concerns principle. It is therefore argued that the direct consequences of this simplification can be considered as indirect benefits of using BPMN + DMN. First of all, a simplified business process model is easier to read and maintain. The latter is especially the case when changing a small detail in the decision logic impacts the whole process model. When extracting decision logic from the process model, both models can consider each other as a black-box, and thus modifications in one model would not influence the other (Batoulis et al., 2015). Secondly, simpler business processes are considered to be more agile. Replacing the complex decision fragment with a single Business Rule Task does not only clarify the process, it also makes it easier for the business to eventually make a change to it. Taylor et al. (2013) argue that a company is only as agile as its business processes, and thus this simplification leads to a more agile business. Thirdly, the purpose of both BPMN and DMN is to facilitate interchangeability and reusability of process and decision models. Add this to the fact that business process models are key in designing information systems, development and maintenance costs can be reduced significantly. Moreover, well designed information systems may lead to additional cost reductions through operational excellence (Recker, zur Muehlen, Siau, Erickson, & Indulska, 2009; Sánchez-González, Ruiz, García, & Cardoso, 2011). Finally, business process models are frequently used as a tool to communicate between different stakeholders. In this context, the process model plays a significant role in aligning business with IT. A better understanding of the business process model (i.e. the business) will undoubtedly lead to better designed information systems (i.e. IT), and
thus better alignment between the two (Taylor et al., 2013). This, again, will further translate in additional savings with regards to maintenance and operational costs (Recker et al., 2009).

Next to the reduction in complexity of the process model three other benefits of BPMN + DMN have been identified in past research. First, by modelling the decision in itself, the logic behind it becomes much clearer for the stakeholders. This integration guides employees through the decision-making process, and can prevent compliance pitfalls (Underdahl, 2011). Secondly, detailed decision logic leads to opportunities with regards to automated decision-making, and thus automated (or straight-through) processing (Taylor et al., 2013). In that case, there is no more need for human intervention during the process, freeing up expensive resources to more value-adding activities. Moreover, reducing human intervention evidently leads to less human error (Repa & Zeleznsik, 2014). Thirdly, decision modelling allows a company to use big data analytics more effectively. With the current hype in big data, companies are more and more looking for data-driven decision-making tools to improve business results, customer understanding, and risk management (Marsh, Pane, & Hamilton, 2006). Since DMN allows for explicit decision modelling (i.e. the decision is broken down into simpler components), it becomes clear to where exactly the decision could be improved. This, in turn, has a positive effect on the processes that rely on that decision (Taylor et al., 2013).

Past research has thus identified four key benefits of using BPMN + DMN: (1) reduced complexity in the process model through the separation of concerns principle, (2) clearer decision logic for all stakeholders, (3) increased automation or straight-through processing, and (4) more effective big data analytics.

In this paper we will elaborate on the first advantage, i.e. the suggestion that separating decision logic from the process model leads to a separation of concerns, which in turn reduces the complexity of the business process model. In order to validate this hypothesis, it is important to understand what drives complexity in a business process model, and more specifically in BPMN models.
4. THE COMPLEXITY OF A BPMN MODEL

Quite some research with regards to the complexity of business process models has been conducted, and researchers unanimously agree that there does not exist one single metric to measure the complexity of a business process model (Cardoso, 2005b; Kluza & Nalepa, 2012; Muketha, Ghani, Selamat, & Atan, 2010; Vanderfeesten, Cardoso, Mendling, Reijers, & van der Aalst, 2007).

According to Cardoso (2005b) there are four components to consider when measuring the overall complexity of a business process model: (1) activity complexity, (2) control-flow complexity, (3) data-flow complexity and (4) resource complexity. In later research, Vanderfeesten et al. (2007) propose investigating five domains to assess the quality of a business process model: (1) size, (2) complexity, (3) coupling, (4) cohesion, and (5) modularity. Although Vanderfeesten et al. refer to these five domains as quality metrics, they are related to the four complexity components laid out by Cardoso.

Besides Cardoso’s and Vanderfeesten et al.’s view there exists dozens of additional research that can be consulted to quantify the complexity of a BPMN model. Kluza, Nalepa, & Lisiecki (2012) and Pavlicek, Hronza, & Pavlickova (2016) mention most, if not all, existing metrics in their research. The purpose of this section, however, is not to go over all existing metrics, but rather provide a strong basis to compare the complexity of a process model in BPMN and BPMN + DMN. The research conducted by Cardoso, Vanderfeesten et al. and their critics will form a good starting point to compare the process models modelled in BPMN and BPMN + DMN. Furthermore, we will develop an alternative metric to measure the data-flow complexity as none of the existing research proposes a solid approach to capture this component.

The process model depicted in Figure 5 will serve as a running example to illustrate the metrics that will be discussed. The process starts with the submission of a loan application by a customer. The clerk first files the application documents. When an error is found, the customer is asked for corrections, who can then decide to reply to or ignore this request. If the application was filled in correctly, it is checked which type of loan the customer is applying for. For each loan type, a different workflow is followed (i.e. home, student or car loan). In the end, the client is notified on the bank’s decision,
after which the clerk saves and archives the application. Depending on the technology available within the bank, the clerk can opt to keep a backup, remote, and/or local copy. In the model, all activity descriptions include a character between brackets (e.g. [A]) to ease referencing in the text.
Figure 5: Loan Application (BPMN). Adapted from Cardoso (2008).
4.1 COMPONENTS OF COMPLEXITY

4.1.1 ACTIVITY COMPLEXITY

The activity complexity is the complexity that arises from the amount of activities in the business process model. This component originates from the lines of code (LoC) metric used in programming, which simply counts the lines of code in a software program (Jones, 1986). Cardoso, Mendling, Neumann, & Reijers (2006) translated Jones’ metric to the BPM domain. First of all, they define the NOA-metric which counts the number of activities in the process model. Secondly, the NOAC-metric is equal to the NOA-value plus the amount of control-flow elements (i.e. gateways). Since all processes can be modelled in a structured way (i.e. all split-gateways can be complemented by join-gateways) using BPMN, it is sufficient to count the control-flow structures corresponding to splits (Gruhn & Laue, 2006a).

In the running example (Figure 5), the NOA is equal to 18 and the NOAC is equal to 25 (eighteen activities + seven split-gateways).

4.1.2 CONTROL-FLOW COMPLEXITY

Cardoso (2008) particularly pays attention to the control-flow complexity in his research. He defines control-flow complexity as the degree to which processes are difficult to analyse, understand, or explain. The control-flow (complexity) is influenced by loops, splits, joins, starting and ending points in the business process model.

In 2005, Cardoso started developing a metric to measure the control-flow complexity of a process model through his CFC-metric. He based his research on the cyclomatic number, formulated by McCabe (1976), which measures all possible control flows in a software program. Simply put, the cyclomatic number is equal to the number of binary decisions that have to be made plus one. Non-binary decisions with n possible solutions are counted as n-1 binary decisions. A program with a low cyclomatic number (i.e. few decisions) is easier to understand than a program with a high cyclomatic number (Cardoso, 2006).
Cardoso’s CFC-metric is, just like the cyclomatic number, based on the amount of (non-)binary decisions to be made. It is comparable with the number of mental states that a designer must consider when developing the model. For each OR-, XOR- and AND-split (each with n outgoing paths) in the business process model a number is added to the CFC-metric based on the number of paths that must be processed:

- OR-split: exactly $2^n - 1$ possible paths must be processed (a path is either followed or not).
- XOR-split: exactly one from the n possible paths must be processed, but the developer (or designer) still has to consider all possible states (n), thus n is added to the CFC.
- AND-split: exactly n paths must be processed, yet the developer has to consider just one state since the outcome of an AND-split is to always execute all outgoing paths, thus only 1 is added to the CFC (Cardoso, 2006).

The absolute control-flow complexity is the sum of the CFC-values of all XOR-, OR- and AND-splits in a process model P:

$$CFC_{abs}(P) = \sum CFC_{XOR-split}(i) + \sum CFC_{OR-split}(j) + \sum CFC_{AND-split}(k)$$

with $i \in XOR-splits of P$, $j \in OR-splits of P$, and $k \in AND-splits of P$.


Next to the absolute control-flow complexity, the relative control-flow complexity can be used as an additional measurement. The latter metric explicitly takes the amount of XOR-, OR- and AND-splits in the business process model into account. This allows for comparison between models that differ significantly in size. The relative control-flow complexity is calculated as follows:

$$CFC_{rel}(P) = \frac{CFC_{abs}(P)}{\#XOR-split(P) + \#OR-split(P) + \#AND-split(P)}$$


After multiple experiments, Sánchez-González et al. (2011) came up with threshold values for Cardoso’s CFC-metric. With these values, every process model can be classified in one of five categories, ranging from “very easy to understand” to “very
difficult to understand”. It is important to note, however, that these experiments were conducted with test subjects that have a relatively weak theoretical knowledge about BPMN models (undergraduate students). Furthermore, the threshold values fluctuate depending on the stakeholder’s experience and previous knowledge with BPMN models. It is clear that an experienced professional with, say, twenty years of experience will have a different and better understanding of the same model than an undergraduate student.

The model in Figure 5 consists of 4 XOR-split gateways, 1 AND-split gateway, 1 OR-split gateway, and 1 event-based split gateway. In this case, the latter is equivalent to a XOR-split (i.e. either the client responds or (s)he doesn’t). For each XOR-, AND-, and OR-split n, 1 and $2^n – 1$ is added to the CFC metric, respectively, with n the number of outgoing paths. The results are summarized in Table 2.

<table>
<thead>
<tr>
<th>Split</th>
<th>Outgoing Paths (n)</th>
<th>CFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFC$_{XOR[A]}$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CFC$_{EVENT[B]}$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CFC$_{XOR[C]}$</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>CFC$_{XOR[D]}$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CFC$_{AND[E]}$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CFC$_{XOR[F]}$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CFC$_{OR[O]}$</td>
<td>3</td>
<td>$2^3 – 1$</td>
</tr>
<tr>
<td>CFC$_{abs}$</td>
<td>-</td>
<td>$= 2 + 2 + 3 + 2 + 1 + 2 + (2^3 – 1)$</td>
</tr>
<tr>
<td>CFC$_{rel}$</td>
<td>-</td>
<td>$= \frac{19}{7}$</td>
</tr>
</tbody>
</table>

$= 2.71$

Table 2: The control-flow complexity of the Loan Application process. Adapted from Cardoso (2008).

Making use of Sánchez-González et al.’s (2011) thresholds, the business process model depicted in Figure 5 is very easy to understand (no more than 6 XOR-splits, 1 OR-split, and 1 AND-split).
Cardoso agrees that his CFC-metric should be accompanied with other metrics to measure the overall complexity of a BPMN model, but Gruhn & Laue (2006b) claim that using the CFC-metric to measure the control-flow complexity alone will not suffice. First of all, they mention that Cardoso does not take into account any information about the structure of the process model (Gruhn & Laue, 2006a). The claim is that a structured model will be much easier to understand (i.e. lower complexity) than an unstructured model with the same CFC value. To incorporate the structure of the process model, they suggest using the maximum and mean nesting depth (ND) metrics, both originating from research in the software domain. Gruhn & Laue (2006a) stated: “The nesting depth of an action is the number of decisions in the control flow that are necessary to perform that action. The greater these two metrics, the more complex the process model.” (p. 5). However, their concise definition makes it not so clear to calculate the (maximum) nesting depth in practice. After digging deeper into existing research, we found that the nesting depth is equal to the amount of decisions to be taken between a particular split- and join-gateway (Sun & Hou, 2014).

Following the definition of Sun & Hou (2014), we calculated the maximum and mean nesting depth of the process model (Figure 5). The splits after activity [A], [B], [C] and [O] are not nested (i.e. they don’t occur between another split-join-structure). The splits after activity [D], [E] and [F] occur directly after the XOR[C]-split, and thus they are considered nested. Their nesting depths are equal to two (since they are the second decision to be made within the XOR[C]-structure). The maximum nesting depth of the process model is thus equal to two. For the mean nesting depth, we first count the maximum amount of decisions to be taken during the process. For the model in Figure 5 this is equal to five: (1) XOR[A], (2) EVENT[B], (3) XOR[C], (4) XOR[D], XOR[E] or XOR[F], and (5) OR[O]. The nesting depth occurring during these decisions is equal to one for decision (1), (3) and (5). Since decision (2) comes directly after (1), its nesting depth is equal to two. Decision (4) comes after (3), which is classified as a XOR-split, and thus only one path can be followed in any case. The nesting depth occurring at that moment is therefore equal to the average of the nesting depths (i.e. \( \frac{1}{3} (2 + 2 + 2) = 2 \)). The mean nesting depth of the model is thus equal to

\[
\frac{1+2+1+\frac{1}{3}(2+2+2)+1}{5} = \frac{7}{5} = 1.4.
\]
The calculations with regards to the nesting depth are summarized in Table 3.

<table>
<thead>
<tr>
<th>Split</th>
<th>Nesting Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFC\textsubscript{XOR}[A]</td>
<td>1</td>
</tr>
<tr>
<td>CFC\textsubscript{EVENT}[B]</td>
<td>2</td>
</tr>
<tr>
<td>CFC\textsubscript{XOR}[C]</td>
<td>1</td>
</tr>
<tr>
<td>CFC\textsubscript{XOR}[D]</td>
<td>2</td>
</tr>
<tr>
<td>CFC\textsubscript{AND}[E]</td>
<td>2</td>
</tr>
<tr>
<td>CFC\textsubscript{XOR}[F]</td>
<td>2</td>
</tr>
<tr>
<td>CFC\textsubscript{OR}[O]</td>
<td>1</td>
</tr>
</tbody>
</table>

Maximum ND = 2

Mean ND = \frac{1 + 2 + 1 + \frac{1}{3} (2 + 2 + 2) + 1}{5} = \frac{7}{5} = 1.4

Table 3: The maximum and mean nesting depth of the Loan Application process.

Secondly, Gruhn & Laue propose using cognitive weights, based on Shao and Wang’s research. Shao & Wang (2003) define cognitive weights as “a metric to measure the effort required for comprehending a piece of software.” (Gruhn & Laue, 2006a, p. 8). Gruhn & Laue further remark that not all these weights are relevant to business process models, and that there might exist cognitive weights for the BPM domain that have not been covered in the software domain (e.g. cancellation in BPMN). They investigated all relevant cognitive weights, and made the transition from the software domain to the business process domain. Their findings are summarized in Table 4. Additionally, the weights have been practiced on process model (Figure 5), where applicable.
<table>
<thead>
<tr>
<th>Workflow</th>
<th>BPM</th>
<th>Software</th>
<th>Weight $W_i$</th>
<th>Loan Application $x_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclusive Choice</td>
<td>XOR-split (exactly one of two branches is chosen) with XOR-join</td>
<td>Branching with if-then structure</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>XOR-split (exactly one of three or more branches is chosen) with XOR-join</td>
<td>Branching with case structure</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Parallel Split and</td>
<td>AND-split activating all outgoing links in parallel; AND-join</td>
<td>Execution of control flows in parallel</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Synchronization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple Choice and</td>
<td>OR-split (a number of branches is chosen from two or more branches)</td>
<td>Branching with case structure,</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Synchronizing Merge</td>
<td>with OR-join</td>
<td>followed by parallel execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composite Task</td>
<td>Sub-process</td>
<td>Call of a user-defined function</td>
<td>2</td>
<td>N/A</td>
</tr>
<tr>
<td>Multiple Instances</td>
<td>Multiple Instance Activity</td>
<td>Branching, followed by parallel</td>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>Patterns</td>
<td></td>
<td>execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cancel Activity</td>
<td>Cancellation of a single activity</td>
<td>Non-existent</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>Cancel Case</td>
<td>Cancellation of multiple activities</td>
<td>Call of a user-defined function</td>
<td>2</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 4: Cognitive weights for BPM elements. Adapted from Gruhn & Laue (2006).
In what follows, we clarify the cognitive weights for the process model in Figure 5 (i.e. the last column of Table 4). First of all, there are eight consecutive steps in the workflow (from the start-event “Loan Application Received” to the regular end-event “Finish Loan Application”). Furthermore, the model consists of four XOR-splits (of which two with, and two without XOR-join) where exactly one path is chosen out of two. However, since all BPMN models can be modelled in a structured way (i.e. all splits can, in theory, be complemented by joins), we assume a value of four for this particular weight. Moreover, there is one XOR-split where one path is chosen out of a possible three. Additionally, the model contains one AND-split (activating two outgoing paths), as well as one OR-split (where at least one path is chosen out of a possible three). The last four weights (i.e. “Composite Task”, “Multiple Instances Patterns”, “Cancel Activity”, and “Cancel Case”) are not applicable, since they do not occur in the model depicted in Figure 5. The cognitive weights-metric for this particular process model is thus equal to $W_i \cdot x_i = 1 \cdot 8 + 2 \cdot 4 + 3 \cdot 1 + 4 \cdot 1 + 7 \cdot 1 = 30$.

To measure the control-flow complexity of a process model it is thus advised to use multiple metrics. Complementary to Cardoso’s CFC-metric, Gruhn & Laue propose using additional measures like maximum and mean nesting depth, as well as Shao & Wang’s cognitive weight metrics adopted from the software domain.

4.1.3 DATA-FLOW COMPLEXITY

OMG (2011) explicitly states in its specification that BPMN is not a data-flow language. It was merely designed to present a high-level overview of the flow of data, and the associations of data artefacts (i.e. data objects or data stores) with activities. For more complex representations of data flows, model designers should turn their attention to data-flow diagrams or (extended) entity-relationship diagrams. They further mention that data and information modelling is out of scope for BPMN. OMG thus argues for a separation of concerns (data, process and decision models should be separated), yet some researchers advocate the use of hybrid models. These models incorporate detailed data and decision modelling in the business process (Cruz, Machado, & Santos, 2012; Meyer, Pufahl, Fahland, & Weske, 2013). Although the usage of hybrid models is out of scope for this text, their existence makes it clear that a significant difference in data-flow complexity between various process models may occur. Thus,
there is certainly a need to measure the data-flow complexity of a business process model.

The existing research with regards to data-flow complexity in the BPM domain is rather limited and still in its infancy (Cardoso et al., 2006). Cardoso (2005a) mentions that the data-flow complexity increases with the complexity of the model’s data structures, the number of formal parameters and the mapping between the data of the activities. According to his research, a data-flow complexity metric should consist of three sub-metrics: (1) data complexity, (2) interface complexity, and (3) interface integration complexity. He has, however, not proposed a holistic approach to capture the data-flow complexity component.

Most, if not all, metrics mentioned above (e.g. LoC, CFC, maximum and mean nesting depth, cognitive weights) stem from the software domain. Intuitively, a metric to measure the data-flow complexity of business process models could as well be derived from the same domain.

Indeed, Oviedo (1980) proposed a metric to determine the overall complexity of a software program by taking the weighted sum of the control-flow and data-flow complexity. Since measuring control-flow complexity has already been discussed in detail, we will focus solely on measuring the data-flow complexity using Oviedo’s DF-metric. Rook (1990) writes: “The data-flow metric is based on the number of variables referenced, but not defined in a program block. A variable definition is equivalent to the assignment of a value, while a reference is the use of a variable in an expression or as output.” (p. 458). To clarify, a program block is a piece of code that is executed as one unit. For example, in Java the code between curly brackets { … } would be classified as a program block.
The formula for Oviedo’s DF-metric is clarified by Weyuker (1988). She defines \( R_i \) as the set of variable definitions that reach block \( n_i \) (i.e. locally available variable definitions). If \( V_i \) is the set of variables whose references are locally exposed in block \( n_i \), then that block’s data-flow complexity is equal to:

\[
DF_i = \sum_{j=1}^{\|V_i\|} DEF(v_j)
\]

where \( DEF(v_j) \) is equal to the number of available definitions of variable \( v_j \) in set \( R_i \).


The total data-flow complexity of a program is then the sum of the data-flow complexity of all its blocks (Weyuker, 1988).

The challenge is to map these definitions to the business process domain, which will allow us to measure the data-flow complexity of a process model. It is first required to find the business process equivalents of: (1) a software block, (2) a variable definition, (3) a variable reference, (4) a locally available variable definition, and (5) a locally exposed variable reference. If we are able to do so, we can easily substitute these elements in Oviedo’s DF-formula.

Vanderfeesten et al. (2007) summarize the similarities between business processes and software design as follows: “A software program is usually partitioned into modules or functions (i.e. activities), which take in a group of inputs and provide some output. Similar to this compositional structure, a business process model consists of activities, each of which contains smaller steps on elementary data elements.” (p. 1).

This insight allows us to map Oviedo’s definitions to the BPM domain:

- A software block resembles the activities in a business process that are executed as a unit.
- A variable definition is equivalent to modifying a data element in the process of completing an activity.
- A variable reference is equivalent to using a data element in the process of completing an activity.
- A locally available variable definition in the activity is equivalent to the definition of the data element in the activity.
- A locally exposed variable reference in the activity is equivalent to the reference to a data element which is not preceded in the activity by a definition of that same data element.

To calculate the data-flow complexity of an activity, the following logic applies: If \( V_i \) is the set of data elements whose references are locally exposed in activity \( n_i \), then that activity’s data-flow complexity is equal to Equation 3, where \( \text{DEF}(v_j) \) is equal to the number of available definitions of data element \( v_j \) in set \( R_i \), with \( R_i \) the set of data element definitions that reach activity \( n_i \).

To clarify, in a process model, some activities make use of (i.e. reference) a data object, while others modify (i.e. define) it. In the BPM domain, the data-flow complexity of an activity is simply equal to the sum of all prior definitions of locally exposed data objects that reach this particular activity. For completeness’ sake, we speak of a locally exposed object when that particular object is not redefined within the same activity.

When modelling the process depicted in Figure 5 several data objects have been added. Let’s assume that we take a process instance dealing with a student loan (i.e. follow the flow to activity [E]) as an example. The process starts with activity [A], where the data object “Loan Application” is defined for the first time. It is redefined (i.e. locally available) in the activities [C], [J] and [N]. The object is referred to in the activities [C], [J], [O] and [R]. Furthermore, the variable is locally exposed in activities [A], [O] and [R] (i.e. the data object is not redefined within the same activity). The object, however, is not locally exposed in activity [C] and [J], since it is redefined in the same activity (i.e. adding data with regards to the type and statistics of the loan, respectively). To calculate the DF-value of the activity [R], for example, we simply need to count all prior definitions of locally exposed variables that reach this activity. This is equal to two, as the locally exposed data object “Loan Application” is (re)defined in [A] and [N]. The same logic applies to calculate the DF-value for the remaining activities.
4.1.4 RESOURCE COMPLEXITY

The fourth and last component, according to Cardoso (2005b), of business process complexity is the resource complexity. In any process, certain activities make use of resources (HR, IT, IS, etc.) that are necessary to complete a certain activity successfully. Intuitively, a high amount of required resources will result in a high resource complexity (Cardoso, 2006). Although BPMN allows linkage between resources and activities (through pools and swim-lane constructs), it is explicitly stated in the specification that the modelling of organizational structures and resources is out of scope for BPMN (Object Management Group, 2011). Again, OMG argues for a separation of concerns (details with regards to the process and the resources used in the process should be separated) in its specification. However, some researchers advocate modelling (detailed) resources in the process model (Martinho & Domingos, 2014; Stroppi, Chiotti, & Villarreal, 2011). Motivations for this point of view are the critical roles that humans play in automated business processes, the distribution of the work to employees, the importance of physical resources used, etc. In this case, specific extensions exist to handle the modelling of resources in the process model. Here, it is argued that modelling resources should be avoided using BPMN, since it is a high-level modelling language (i.e. the operations of a single activity are never modelled in BPMN). If a designer still wants to add more details with regards to resources, (s)he is advised to use an extension on the BPMN language or adopt a language that better fits his or her requirements. While DMN has an (indirect) resource view through the “Knowledge Source” and “Authority Requirement”, the main goal of this paper is to compare the process model modelled in BPMN and BPMN + DMN. It is therefore, in this context, sensible to question the relevance of resource complexity in comparison with the activity, control-flow and data-flow complexity components.

4.2 QUALITY DOMAINS

Vanderfeesten et al. (2007) propose investigating five domains to assess the quality of a business process model: (1) size, (2) complexity, (3) coupling, (4) cohesion, and (5) modularity. These concepts stem from the software domain, and have been used in the past to determine the quality of a software program.
The first two domains (size and complexity) have already been discussed in detail. The size domain simply relates to the activity complexity, while the complexity domain as defined by Vanderfeesten et al. refers solely to control-flow complexity. Thus, the three remaining domains to discuss are: (1) coupling, (2) cohesion, and (3) modularity.

4.2.1 COUPLING

Remember from before that Cardoso (2005a) defined three sub-components under data-flow complexity: (1) data complexity, (2) interface complexity, and (3) interface integration complexity. The third sub-component (i.e. interface integration) refers to coupling. It is the number of interconnections among the activities in the model, the degree of interdependence between activities in the process model (Vanderfeesten et al., 2007).

The most promising metric to capture the degree of coupling is defined by Reijers & Vanderfeesten (2004). They count the overlap of data elements for each pair of activities. Two activities are coupled (or linked) if they contain one or more shared data elements. To calculate the coupling value, activities are selected pair-wise, after which the coupled pairs are counted. In their research Vanderfeesten et al. (2007) complement the coupling metric with a cohesion metric, which will be discussed in the next subsection.

The coupling domain proposed by Vanderfeesten et al. (2007) captures the interdependence of activities on each other and the data elements they use, and is thus deemed highly relevant for the overall complexity of a business process model. The metric that will be used in this paper is the one proposed by Reijers & Vanderfeesten (2004).

To calculate the coupling value for the business process model depicted in Figure 5, we follow the steps laid out by Reijers & Vanderfeesten (2004). We first determine which activities are linked. In this case, all activities that are defining or using the same data object are considered to be coupled (i.e. value of ‘1’ in Table 5). For example, activity [A] is linked with activities [C], [G], [H], [J], [K], [L], [N], [O], and [R] since these all define or use the data object. The total amount of links in the process model is equal
to 90 (sum of rows or columns in Table 5). The maximum possible amount of links (i.e. when all activities are linked with each other) is calculated as follows: $NOA \times (NOA - 1) = 18 \times 17 = 306$. The coupling value ($k$) is then calculated as $k = \frac{90}{306} = 0.2941$.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | ∑ |
| - | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 9 |
| 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | - | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 9 |
| 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | - | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | - | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | - | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | - | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | - | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | - | 0 | 0 | 1 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | - | 9 |
| 9 | 0 | 9 | 0 | 0 | 9 | 9 | 0 | 9 | 9 | 9 | 0 | 9 | 9 | 0 | 0 | 9 | 90 |

Table 5: The coupled activities in the Loan Application process.
4.2.2 COHESION

Cohesion is defined as the coherence within the parts (i.e. activities) of the model. It focuses on the content of an activity, and examines how its operations are linked with each other (Reijers & Vanderfeesten, 2004).

Cohesion is deemed irrelevant for business process models modelled in BPMN, since it is a high-level modelling language (i.e. the operations of an activity are never modelled in BPMN). Another argument supporting its irrelevance for the purpose of this paper is that this metric is only used to derive the coupling-cohesion ratio. The latter assists the model designer in selecting the best alternative out of multiple designs, but it does not focus on the complexity of a single design (Vanderfeesten et al., 2007).

4.2.3 MODULARITY

Modularity refers to splitting up a model into several parts (i.e. sub-processes, links, etc.). Vanderfeesten et al. (2007) argue that modularity metrics are not relevant for business processes, since activities are usually considered as black-boxes. However, on a higher level it is possible to add sub-processes, link events, etc. to the model, which allows easier reading, understanding and maintainability of the process model. Furthermore, with DMN in the back of our heads, decision constructs will be replaced by Business Rule Tasks (which the user will probably be able to expand), and thus modularity in the model might change significantly.

Indeed, the first sentence in Reijers & Mendling (2008) research is: “The use of subprocesses in large process models is an important step in modeling practice to handle complexity.” (p. 1). They mention several advantages like reusability, scalability and easier understanding of the process model. After empirical testing they state that the understandability significantly increases when dealing with large models where modularity is applied to a high extent. Turetken, Rompen, Vanderfeesten, Dikici & van Moll (2016) found similar results in their research.
Cardoso et al. (2006) classify modularity as a sub-component of data-flow complexity under interface complexity. They propose using an adaptation of Henry & Kafura’s (1981) Information-Flow metric, but Abreu, da Porciúncula, Freitas, & Costa’s (2010) alternative adaptation from the same Information-Flow metric looks a lot more promising. They propose the following formula to measure modularity in a business process model:

\[
HKM = \text{total activities} \times (\text{number of start events} \times \text{number of end events})^2
\]


The total activities can be calculated by simply counting the activities (possibly including control-flow elements), as discussed in activity complexity.

The process model in Figure 5 contains one start-event (i.e. “Loan Application Received”) and two end-events (i.e. “Loan Application Terminated” and “Finish Loan Application”). The HKM-value of the process model is thus equal to \(18 \times (1 \times 2)^2 = 72\), when using the NOA metric to determine the total amount of activities.

4.3 CONCLUSION

It is clear that the complexity of a business process model can never be comprised in a single metric. According to Cardoso (2005b), there are four important areas that add to the complexity of a business process model, i.e. activity, control-flow, data-flow and resource complexity. On the other hand, Vanderfeesten et al. (2007) propose exploring five domains to measure the quality of a process model, i.e. size, (control-flow) complexity, coupling, cohesion and modularity.

After thorough analysis, both views are actually closely intertwined. We can link Cardoso’s activity and control-flow complexity with Vanderfeesten et al.’s size and complexity domains, respectively. Furthermore, the coupling and modularity domains are identified as part of the data-flow complexity.
To summarize, the activity complexity (or size) can be measured by simply counting the activities (possibly including control-flow elements) in the model. Secondly, to measure the control-flow complexity, we will make use of Cardoso’s CFC-metric in combination with the maximum and mean nesting depth, as well as some cognitive weight metrics adopted from the software domain. Thirdly, the data-flow complexity will be measured using an adaptation of Oviedo’s DF-metric from the software domain in combination with coupling and modularity metrics. Finally, the resource complexity and cohesion will not be measured as their contribution to the overall complexity is predicted to be rather limited.
5. REDUCING COMPLEXITY BY USING BPMN + DMN

In what follows, we will present a case that can be modelled as a structured process in BPMN. Hereby, we will incorporate detailed decision logic into the process model, just like many practitioners started doing in recent years. Finally, we will transform the model to BPMN + DMN, after which we will investigate and discuss the reduction in complexity that occurs due to this transformation.

5.1 CASE STUDY

The case, adapted from Batoulis & Weske’s (2017) working paper, describes the loyalty program of German railway company Deutsche Bahn, through the use of the BahnCard (BC). This card offers discounts up to 70% on ticket prices. Two types of BahnCards exist, namely the BC 25 and BC 50, offering a minimum discount of 25% and 50%, respectively. With each purchase, a customer receives additional points on his or her card (one point for each euro spent). When a customer accumulates more than 1,000 points the discount increases from 25% to 60% for the BC 25, and from 50% to 70% for the BC 50.

5.1.1 BPMN MODEL

It is not an easy task to represent the decision logic in such detail in a BPMN model without being overly complex, since the standard was never designed for this. However, since many real-life businesses keep using BPMN to incorporate detailed decision logic in the process model, we deemed it interesting to try it as well.

The process (Figure 6) starts when a booking from a customer is received. It is first checked whether or not a BahnCard has been linked to the order. If a BahnCard was not linked, the railway asks the customer to opt in for the loyalty program, which a customer can decline or accept. If the customer declines, no discount is granted and the booking is completed. If the customer accepts the offer (or made the booking with an already existing BahnCard), the railway first checks the type (i.e. BC 25 or BC 50). Afterwards, the amount of points on the card is checked. On basis of this data, the
correct discount is applied. Finally, the points are added to the customer's BahnCard, and the booking is complete.

It is important to note that, according to modelling theory, the amount of the discount is actually not relevant to the control-flow (i.e. there is no difference between granting a discount of 25, 50, 60 or 70% with regards to the flow of the process). However, if this information is not added to the model, there is no way of knowing which discount to grant the customer. Therefore, the model in Figure 6 is a practical example of wrongly, yet understandably, incorporating decision logic in a business process model, which quickly leads to increased complexity.
Figure 6: Business process model (BPMN) of the loyalty program with detailed decision logic. Adapted from Batoulis & Weske (2017).
5.1.2 BPMN + DMN MODEL

DMN allows us to extract the decision logic from the business process model, as discussed in Section 3. Hereby, the main goal is to simplify the decision construct (i.e. in Figure 6: starting from activity [C] until the XOR-join gateway before activity [J]). In this case, we can replace the decision fragment with a single Business Rule Task “Manage Discount” (Figure 7). As mentioned before, the flow is not influenced by the outcome of the decision, since it does not matter how much of a discount should be applied, but rather the fact that a discount must be applied. This is a significant difference with the process model depicted in Figure 6, where the missing information (i.e. the amount of discount to grant) was repeatedly modelled as a different flow.

Of course, the model in Figure 7 is missing crucial information regarding the level of the discount. That's why the process model is complemented with a DMN model. The decision requirements diagram (Figure 8) carries all necessary information with regards to the structure of the decision. The “BahnCard” serves as input data for the decision “Manage Discount”, which is subject to the business knowledge model “Discount Rules”. The latter are represented in a decision table at the decision logic level (Table 6).

To summarize, rather than one business process model in BPMN (Figure 6), we now supposedly have a less complex process model in BPMN (Figure 7) complemented by a DMN model, which consists of a decision requirements diagram (Figure 8) and a decision table (Table 6).
Figure 7: Business process model (BPMN + DMN) of the loyalty program. Adapted from Batoulis & Weske (2017).
Figure 8: Decision requirements diagram of the loyalty program. Adapted from Batoulis & Weske (2017).

<table>
<thead>
<tr>
<th>UC</th>
<th>BahnCard Type</th>
<th>BahnCard Points</th>
<th>Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>&lt; 1000</td>
<td>25%</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>&lt; 1000</td>
<td>50%</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>≥ 1000</td>
<td>60%</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>≥ 1000</td>
<td>70%</td>
</tr>
</tbody>
</table>

Table 6: Decision logic level: a decision table representing the discount rules. Adapted from Batoulis & Weske (2017).
5.2 FINDINGS

To assess the difference in complexity between the process model in BPMN and the model in BPMN + DMN quantitatively, we will make use of the metrics proposed in Section 4. Shao & Wang’s cognitive weights are not included in our analysis, as they are particularly useful for business process models with more advanced constructs like sub-processes, multiple instance activities or cancellation of activities, none of which occur here. Furthermore, with regards to the data-flow complexity, the modularity metric will not be practiced either. The latter is particularly useful when using sub-processes or link events in the business process model.

Let’s start with measuring the activity complexity for both process models. In the first model (Figure 6), the NOA and NOAC amount to 11 and 16 (eleven activities plus five split-gateways), respectively, while in the second process model (Figure 7), they are equal to 6 and 8 (six activities plus two split-gateways), respectively. Secondly, with regards to the control-flow complexity, we will make use of the CFC\textsubscript{abs}, CFC\textsubscript{rel}, maximum and mean nesting depth metrics. For the BPMN model (Figure 6), the CFC\textsubscript{abs} is equal to 10 (two paths coming from five XOR-splits). The maximum nesting depth amounts to 2 (maximum two decisions must be made between a split- and join-gateway). During the four decisions (i.e. (1) checking the BahnCard, (2) offering the BahnCard, (3) checking the BahnCard Type and (4) checking the BahnCard Points) that have to be made, the following nesting depths occur: $1 + 2 + 1 + \frac{1}{2}(2 + 2) = 6$, and the mean nesting depth is thus equal to 1.5 ($= \frac{6}{4}$). The model in Figure 7, on the other hand, has a CFC\textsubscript{abs} equal to 4 (two paths coming from two XOR-splits), while the maximum and mean nesting depths amount to 2 (maximum two decisions between a particular split- and join-gateway) and 1.5 ($= \frac{3}{2}$; during the two decisions, i.e. checking the BahnCard and offering the BahnCard, that have to be made, the following nesting depths occur: $1 + 2 = 3$), respectively. Finally, to measure the data-flow complexity we are going to use our adaptation of Oviedo’s DF-metric as well as the coupling metric. Due to the fact that the data object “BahnCard” is never defined and referenced within the same activities (i.e. there are no locally exposed variables), the DF-metric is zero for both process models. To measure coupling, we first have to determine the linked activities in both models (Table 7 and Table 8 for the process model in BPMN and
BPMN + DMN, respectively). The maximum amount of linked activities is equal to $NOA \times (NOA - 1)$. For the model in Figure 6 the coupling value is thus equal to $\frac{20}{11 \times 10} = 0.18$, while for Figure 7 it is equal to $\frac{6}{5 \times 5} = 0.2$.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>Σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Σ</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 7: The coupled activities in the process model (BPMN).

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>Σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Σ</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 8: The coupled activities in the process model (BPMN + DMN).
The results of the complexity metrics are summarized in Table 9.

<table>
<thead>
<tr>
<th>Complexity Type</th>
<th>Metric</th>
<th>BPMN</th>
<th>BPMN + DMN</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Complexity</td>
<td>NOA</td>
<td>11</td>
<td>6</td>
<td>-45%</td>
</tr>
<tr>
<td></td>
<td>NOAC</td>
<td>16</td>
<td>8</td>
<td>-50%</td>
</tr>
<tr>
<td>Control-Flow Complexity</td>
<td>CFC_{abs}</td>
<td>10</td>
<td>4</td>
<td>-60%</td>
</tr>
<tr>
<td></td>
<td>CFC_{rel}</td>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Maximum Nesting Depth</td>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Mean Nesting Depth</td>
<td>1.5</td>
<td>1.5</td>
<td>-</td>
</tr>
<tr>
<td>Data-Flow Complexity</td>
<td>DF</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Coupling</td>
<td>0.18</td>
<td>0.20</td>
<td>+11%</td>
</tr>
</tbody>
</table>

Table 9: Measuring the complexity for both business process models (BPMN vs. BPMN + DMN).

Although we cannot generalize our findings based on this single case, we can already distinguish four trends when taking a closer look at the numbers.

First of all, the NOA decreases by 45%, while the NOAC reduces even more with 50%, averaging a 47.5% decrease in activity complexity. This is, obviously, due to the replacement of the decision construct in the BPMN model (consisting of seven activities and three split-gateways) by a single Business Rule Task.

Secondly, just like the activity complexity, the control-flow complexity decreases when extracting the decision logic from the process model. However, the change is solely based on the decrease in the CFC_{abs}, with a staggering 60%. The reason that its relative counterpart (i.e. CFC_{rel}) does not change is because both the numerator (i.e. CFC_{abs}) and the denominator (i.e. number of gateways) decrease in the same fashion.

Thirdly, we notice that both nesting depth metrics do not change when extracting the decision logic from the process model. This is due to the fact that transforming the process from BPMN to BPMN + DMN only bears influence on the decision construct that is replaced by the Business Rule Task. When nesting also occurs outside of this construct (as is the case in Figure 6), it is logical that transforming the model does not have a considerable impact (or none, in this case) on the nesting depth.
Finally, the coupling metric surprisingly increases by 11% when extracting the decision logic from the process model. Nonetheless, this can be rationally explained. While the number of links (i.e. numerator) decreases from twenty in the BPMN model to a mere six in the BPMN + DMN model, the maximum possible links (i.e. denominator) plummets from 110 to 30 due to the huge reduction in the NOA-metric (from 11 to 6). In another case, the coupling metric could as well have decreased (e.g. when the NOA does not decrease as much, the reduction in maximum possible links would have been less extreme).

Overall, we can conclude from this case-study that the complexity of the process model decreases when using BPMN + DMN, in comparison to BPMN only. There is a decrease in both the activity and the control-flow complexity (although the CFC\(_{rel}\) and nesting depth metrics do not change). The data-flow complexity, and more specifically the coupling metric increases after the transformation. However, since the relative importance of each metric is not yet known, it is not possible to elaborate on the extent of the decrease.
6. CONCLUSION

6.1 CONTRIBUTIONS

BPMN is considered the standard for business process modelling. However, organizations recently started wrongly incorporating detailed decision logic into the process model (Batoulis et al., 2015). The Object Management Group responded by introducing DMN, a new standard for decision modelling language, which can be used independently or in combination with BPMN.

We first explored the benefits of BPMN + DMN by means of a literature review in Section 3. The existing research suggested that extracting the decision logic from the process model would lead to a separation of concerns, which in turn leads to reduced complexity in the process model. Next to the complexity reduction, we identified three other advantages of BPMN + DMN in the literature: (1) clearer decision logic for all stakeholders, (2) increased automation or straight-through processing, and (3) more effective big data analytics (Batoulis et al., 2015; Biard et al., 2015; Taylor et al., 2013). Centralizing the supposed advantages of the new language BPMN + DMN is considered a first contribution of this master's dissertation.

In Section 4, we investigated how the complexity of a BPMN model is measured. There, it became clear that measuring the complexity of a business process model is not an easy task. Past research noted that the complexity of a BPMN model could never be captured within a single metric. Furthermore, the existing research identified some components (or domains) of complexity, but there is little agreement on which metric should be used in which context. This paper linked the viewpoints of Cardoso, Vanderfeesten et al. and their most important critics. To measure the complexity of a business process model, it is advocated in this paper to differentiate between the activity, control-flow and data-flow complexity, which also includes coupling and modularity. Moreover, we concluded that the resource complexity and cohesion are limited contributors to the overall complexity. To measure these components of complexity, we proposed several existing metrics. More importantly, however, we developed an adaptation of an existing metric from the software domain to capture the data-flow complexity. Our holistic approach to measuring the complexity of a process
model in BPMN, in combination with the practical application on a real-life business process is considered an important contribution to the existing research.

Finally, in Section 5, we investigated the complexity reduction when transforming a process model from BPMN to BPMN + DMN by means of a case-study (based on Batoulis & Weske’s (2017) working paper). We first developed a process model with detailed decision logic in it. Afterwards, we extracted the decision logic from the process model by transforming it to a BPMN + DMN model. While comparing the complexity, we identified four trends. First of all, we noticed a clear reduction in the activity complexity. Secondly, the control-flow complexity decreased significantly as well, albeit solely due to the reduction of the absolute CFC metric. The CFC_{rel} did not change due to an equal decrease in both the CFC_{abs} and the number of gateways (i.e. numerator and denominator, respectively). Thirdly, the nesting depth metrics did not alter when transforming the model from BPMN to BPMN + DMN. This is because the transformation only has an effect on the decision construct that is being replaced. When nesting occurs outside of this construct, it is logical that the transformation does not have a considerable impact. Finally, we noted that the coupling value increased when transforming the process model from BPMN to BPMN + DMN, as a consequence of the major decrease in the NOA metric. In the end, we concluded that the complexity of the process model decreases when using BPMN + DMN, in comparison to BPMN only. However, it is not advised to elaborate on the extent of this change, since the relative importance of the different complexity metrics is not yet known. Overall, this exploratory investigation on the basis of a real business case and its identified complexity shifts are perceived as a beneficial contribution to the brand-new research domain of BPMN + DMN.

6.2 LIMITATIONS AND FUTURE RESEARCH

In this subsection, we describe the limitations of the research conducted in this paper, and how it can be improved by means of future research.

The first limitation stems from the fact that very few complexity metrics (in the BPM domain) have been empirically validated (including our adaptation of the DF-metric). Consequently, these metrics are hard to interpret due to a lack of threshold values.
Furthermore, the relative importance of the different metrics is not yet known. Future research focussing on validating the existing metrics would certainly be valuable in this context. This, in combination with threshold values and identifying the relative importance of all metrics, would assist in a better understanding of the differences in complexity between two models.

Secondly, we did not define a clear methodology as to why we selected certain metrics to compare the complexity of two business process models. Consequently, it is not proven that the identified metrics are best suited for this task.

Thirdly, the findings presented in this paper are based on a single case, where not all metrics could be applied. Therefore, it is unsure whether practitioners will actually perceive a complexity reduction in the process model when using BPMN + DMN, instead of solely BPMN (i.e. our results cannot be generalized). The next step is to conduct experiments in a controlled setting (e.g. with students) to confirm our initial findings. This master’s dissertation could certainly serve as preliminary research for experiments in the nearly untapped domain of (BPMN +) DMN. In the long term, it would be valuable to investigate how BPMN + DMN is actually used in practice (i.e. in a non-controlled setting).
REFERENCES


