UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Software Engineering Curriculum

Mihkel Alavere

On the Differences Between Process Models by Novice and Expert Modellers

Master's Thesis (30 EAP)

Supervisors: Marlon Dumas

Matthias Kunze

Tartu 2015

On the Differences Between Process Models by Novice and Expert Modellers

Abstract:

During the last decade, business process modelling has gained popularity as a way to make the processes of a company explicit and to support the analysis, improvement, implementation and monitoring of business processes. High-quality business process models can therefore support an organization in its continuous improvement efforts. Previous research however has found that the quality of business process models in commercial use is highly heterogeneous. These previous studies have largely focused on models produced by expert users, while the question of what typical errors are made by novice modellers has thus far been left relatively unexplored. Yet, insights into the question of typical errors of novice users (relative to expert ones) can inform the design of process modelling learning material. This thesis contains two studies aimed at providing some answers to the above question. The first part of this thesis examines process models produced by novices and investigates typical errors in their syntax and style. The second part aims at identifying differences between models of the same business process produced by expert vs. novice modellers.

Keywords:

Business process modelling, business process model quality, BPMN.

Algajate ja ekspertide äriprotsessi mudelite võrdlus

Lühikokkuvõte:

Viimase kümnekonna aasta jooksul on äriprotsesside modelleerimine saanud järjest suurema tähelepanu osaliseks, kuna see võimaldab ettevõtetel analüüsida, täiustada ja seirata äriprotsesse. Niisiis võimaldavad kvaliteetsed äriprotsesside mudelid firma efektiivsust tõsta. Senised uuringud on aga näidanud, et äriringkonnas kasutatavate mudelite kvaliteet varieerub kõvasti. Need uuringud on peaasjalikult keskendunud kogenud äriprotsesside loojatele, samas kui algajate tüüpilised vead on jäänud tähelepanuta. Just algajate tavalisemate vigade uurimine on aga kasulik, et tõhustada õppematerjalide arendamist. Käesolev töö sisaldab kahte uuringut, mille eesmärk on pakkuda vastuseid nendele küsimustele. Esimene neist uurib algajate poolt tehtud äriprotsesside mudeleid ja selgitab välja, milliseid vigu nad kõige enam teevad. Teine uuring võrdleb algajate ja professionaalide poolt tehtud mudeleid ja toob välja erinevused nende vahel.

Võtmesõnad:

Äriprotsesside modelleerimine, äriprotsessi mudelite kvaliteet.

TABLE OF CONTENTS

1. INTRODUCTION

Business process management is a corporative methodology which aims to optimise the effectiveness by improving business processes in an organisation. Usually the process flow is described in the terms of a workflow. Using process models helps companies reduce time and cost by reassigning activities to be performed by the personnel. Optimisation can furthermore be achieved through automating workflows and finding bottlenecks.

With the increasing popularity of the field of business process management, some research has already been done which focuses on what types of mistakes experienced modellers make in their models. However, no one has thoroughly inspected how novices design their models and what errors they make. The goal of this paper is to fix that problem by looking at their models and observe what the prevailing mistakes are. Furthermore, although some analysis has been conducted that inspect the differences novices and experts exhibit, it has thus far been very high-level. This document aim is to focus on differences of the actual models and the modelling elements.

Being able to differentiate between novices and experts can be very useful in real world applications. One use for this is an automatic check to filter out some models from a collection of models made by beginners. Because they make mistakes more regularly, their models can then be selected for further inspection to find mistakes in them. This would reduce the workload since only a part of the models would have to be reviewed.

Furthermore, the lecturers in universities can refine their curriculum to address different modellers. The models that are made by amateurs can then be inspected more closely when discussing modelling with this group of people. The same also applies to errors made by professionals. The professors can also use this knowledge to determine whether a student’s homework was created by a professional modeller, thus more easily discovering cheating.

Being able to assess which model was created by whom is useful for students, as well. They can filter out models that were made by skilled modellers and observe how they solved a particular problem. It is also possible to inspect which elements they used and which structural solutions they used.

This paper aims to answer a number of question regarding how amateurs and experts design their models. The questions include: “What typical mistakes are made by novice process modellers?” and “What distinguishes process models produced by novices with comparable process models produced by experts?”.

The document is divided into three parts. Chapter 2 provides an overview of related work in the field of modelling, in particular the way novices model their models and works describing differences between experts and novices. Chapter 3 presents the description and analyses used to inspect models made by novices. Chapter 4 gives an overview of the differences found during the examination of models made by novices and experts. The paper ends with a conclusion.

2. STATE OF THE ART

This chapter reviews previous work related to the research questions posed in the introduction. The chapter begins with a discussion regarding related work on the differences between novice and expert modellers in the field of conceptual data modelling as well as process modelling. Next the chapter reviews related work on process modelling metrics (Section 2.2) and guidelines (Section 2.3), which can be used for assessing a given process model. Section 2.4 analyses research on the relative usage frequency of different BPMN modelling elements. Finally, section 2.5 discusses existing process model similarity metrics, which can be used to compare pairs of related process model and in particular process models produced by expert and novice users.

2.1 Novice vs expert modellers

One of the first studies which studied how novice and expert designers differ in the field of modelling was done in 1990 by Dinesh Batra and Joseph G. Davis [1]. They examined the differences between the two groups in the field of conceptual data modelling. The study characterised both the experts' and novices' approach to modelling and highlights the dissimilarities. From their findings they provided guidelines for the development of knowledge-based tools for conceptual data-modelling task.

Additional research examining how novices build their models was done by Jan Recker *et al.* [2] in 2010. They analysed how novices who have little to no knowledge of modelling design business process models using only paper and pencils. The study provided some understanding regarding how beginner modellers conceptualise and externalise business processes. Their finding reveal that there's a total of five design ranges from textual, mixed to graphical designs. Afterwards they analyse which of them are the most prevalent and the quality traits of the design outcomes.

Both of the document focus rather on the higher-level aspects of modelling. We cannot use the conclusions found by Batra and Davis in our study because they don't compare process models *per se,* which is our goal. The second document mainly focuses on models that are drawn on paper and not on actual process modelling. As such it differs from our aim which is to distinguish novices and experts based on the structure, measurements and usage of elements of process models. As such, those results are of little use to us. Nevertheless, it provides an interesting insight into previous analysis of how novices build their models.

2.2 Process model metrics

A relatively simple way of comparing proficient modellers to novices is to use various metrics that have been developed to describe business process models.

In 2007 Jan Mendling [3] presented an approach for the verification and prediction of errors in business process models. In his paper he firstly provided an overview of already existing metrics. Secondly, he introduced new metrics that capture important process model concepts such as partitionability, connector interplay, cyclicity, and concurrency. Lastly, he discussed their theoretical connection with error probability.

Mendling's research was followed by Mathias Kunze's, [4] who analysed existing process models in the BPMAI [5] collection using existing metrics. BPMAI is a repository which contains tens of thousands of process. He listed a total of five metrics and then used them to calculate various characteristics of BPMN and EPC models and analysed the results. Additionally, he inspected the average sizes and the frequency of use of modelling constructs in both BPMN and EPC models.

These two sources were a good inspiration to compare two groups of modellers by their level of expertise based on the size of the models and what elements they use in their models. Furthermore, the metrics and the methods they used to differentiate experts and novices can be reused later on. The results of the values of each metric would be useful for providing a general assumption of what the values should approximately look like.

2.3 Modelling guidelines

In 2008 Jan Mendling *et al.* [6] proposed seven guidelines titled 7PGM for process modelling. These guidelines govern aspects from building a process model from scratch to improving existing models. The rules mainly advise how to improve modelling from the structural viewpoint. For example, it recommends to use as few elements in the model as possible as this reduces the probability of errors cropping up.

Darius Silingas and Edita Mileviciene [7] presented a number of bad practices in business process modelling and suggest methods to improve erroneous versions. They call the mistakes “bad smells” based on the mistakes in software development. Using a good combination for learning proper business process modelling they identify the problem, discuss what the better practice is and explore how to refactor the model into better quality business process models. More specifically, they look at how elements should be titled and how gateways, events and loops should properly be implemented.

In his book Bruce Silver [8], a renowned trainer in the field of business process management, explained which BPMN elements process modellers need to understand in two levels, including where and how to use each element. Level 1 (the Descriptive modelling subclass of BPMN 2.0) is a palette of shapes and symbols primarily carried over from traditional flowcharting. Level 2 (the Analytic subclass) expands the palette to be able to describe event-triggered behaviour, critical to modelling exception handling. In addition to the rules in his book, he also published a single list of the rules of BPMN in his blog [9]. These rules listed there consist of both the official rules from the BPMN spec and his own “method and style” rules.

Having these sources which list guidelines and rules is a useful for analysing how novices design their models. The list of rules composed by Bruce Silver will be used for third chapter to analyse what errors novices make and how often they occur. The fourth chapter will comprise of a mix rules and guidelines by Jan Mendling, Darius Silingas and Edita Mileviciene and observations made by studying models made by unskilled modellers.

2.4 Modelling elements

In 2008, Michael zur Muehlen and Jan Recker [10] presented a paper in which they analysed the usage of every modelling element in BPMN. They recognised that because BPMN offers such a wide range of modelling constructs, not all of the constructs are equally represented in practice. Furthermore, they analysed how frequently the elements are used in particular environments – academic, consulting and general use. The study was carried based on a total of 120 diagrams.

It is interesting to note that the authors fell out with Bruce Silver over the subject of usage of BPMN elements. In a blog post [11] after the paper was published, Zur Muehlen and Recker stated: "BPMN training programs could benefit from a structure that introduces students to the most commonly used subset first before moving on to advanced modeling concepts." They explained that because some constructs are so rarely used they should be completely disregarded. Silver, on the other hand, argued that the less known elements should not be ignored – instead, the students have to be properly introduced to the elements and that the market still has to grow into the "unnecessary complexity".

In our study we can also take advantage of the idea counting various modelling elements in different user groups. The models are inspected by looking at each model, counting the elements and comparing the results between different user groups.

2.5 Process model similarity measures

One option for comparing how much experts and novices differ is to use similarity measures. Similarity measures assess how similar or different two or more models are to each another. Thus, it can be used in a way that compares the variety of differences between novice models to the variety differences of expert models. Since there are myriad similarity metrics that can be used to contrast process models, a categorisation and description of them is also provided here – a number of papers review similarity metrics by evaluating them.

Remco Dijkman [12] presented a classification which describes the prevalent differences business process models exhibit. The classification he developed is useful for merging process models when a merger of companies takes place. He came up with 9 ways to differentiate models. The contrasting methods, however, were unsuitable for comparing novices to experts.

Chen Li, Manfred Reichert, and Andreas Wombacher [13] provided a method to quantitatively measure the distance and similarity between two process models based on the efforts of model transformation. They evaluate the distance and similarity between two process models based how many changes are needed to turn one model into another. These changes are high-level change operations, such as adding, deleting or moving activities. Although the paper lists some good measures for measuring the difference between models, the methods were overly complicated and are not put to use in this work.

Matthias Kunze [14] has written a document about various similarity measurement methods. The first portion of the paper introduces the notion of business process management as a whole as well as Petri nets, successor relations and process alignments. This is followed by a framework to search for business process models with the help of similarity measures. The paragraph explains how a match is decided and presents a distance function that guides the ranking of search results. Kunze splits similarity into structural similarity and behavioural similarity with either group having their own methods for computing similarity. The final chapter analyses the search techniques and assesses the quality and performance of them.

Michael Becker and Ralf Laue [15] provide a comprehensive survey on techniques to define and calculate similarity measures. Additionally, they provide a survey on techniques to define and calculate similarity measures between PMs. The research discovered that hardly a measure fulfils all desirable properties and that that different similarity measures rank the similarity between PMs very differently. They conclude that there is no universal similarity measure for all situations, but rather that different measures excel in different circumstances.

Another document exploring similarity metrics was written by Remco Dijkman, Marlon Dumas and Luciano García-Bañuelos. [16] They provide a method for ranking process models in a repository according to their similarity. Four graph matching algorithms were evaluated, ranging from a greedy one to a relatively exhaustive one. They showed that the precision of a fast, greedy algorithm is comparable to many exhaustive algorithms.

Juntao Gao and Li Zhang [17] have introduced the concept of semantic similarity to compare business process models. Semantic similarity tries to solve the problem of different modellers of different organisations using different ways to represent business processes. They also conducted an experiment to evaluate the method in which 20 subjects were given 20 model pairs. They had to estimate how similar the models were the each other and the results were then compared with the computational similarity measurement. The paper concludes that the method still needs more projects to be properly verified, though.

As a complement to semantic similarity, Tao Jin, Jianmin Wang, and Lijie Wen [18] proposed yet another similarity measure – structural similarity. In their work they explain the exact details of how to use the similarity and determine whether the algorithm is effective or not through experiments. They summarise that their approach to similarity search is efficient.

Initially, similarity measures were going to be used to differentiate proficient and beginner modellers. Later, though, this plan was dropped as using the measures would have been too time-consuming. Nevertheless, it should be noted that using similarity measures is perfectly valid for such research.

3. CHARACTERISATION OF NOVICE PROCESS MODELLERS

This chapter of the thesis summarises the results of an analysis aimed at investigating what mistakes novice modellers most often make based on the BPMN rules written by Bruce Silver [8]. The chapter is divided into three sections. Firstly, section 3.1 provides an overview of the dataset used as the basis of this chapter. Additionally, the section presents hypotheses predicting what mistakes novices struggle with the most. Section 3.2 describes the rules that were used to check what errors novices commit and the results of the analysis. Lastly, section 3.3 discusses the nature of the mistakes and examines why those mistakes occur and pronounces verdicts regarding the hypotheses.

3.1 Hypotheses and dataset

In order to begin determining differences of models made by novices and experts the models needed to be found. It was important that the models be based on the same scenario. This permitted a more efficient comparison because it enabled the comparison of models side-by-side. Thus, structural similarities were much easier to determine.

Although the inspected models were not based on a similar scenario and thus the analysis was less informative, utile information could still be gathered from those models. Knowing that all models within a collection were written by professionals or beginners also came in handy. The models offered a general overview of the way models are created by novices compared to professionals and vice-versa. For instance, it was possible to investigate the overall structure of the models – this included the use of start and end nodes, the habits of naming tasks and events, usage of gateways etc.

The models that were inspected were obtained from BPMAI. BPMAI (BPM Academic Initiative) is an initiative which fosters teaching and research in business process management by providing a set of exercises and a professional tool to be used as a service free of charge. The BPM Academic Initiative Model Collection repository contains tens of thousands of process models made by over ten thousand students and lecturers. The models vary in language, modelling language, complexity, and domain.

BPMAI provides the possibility of downloading a large collection of business process models based on various attributes. Models can be filtered by language, modelling language, connectedness, size and date of creation. Additionally, it is possible to single out the last revision of each model or all versions of a model. All the models were created using the program Signavio. The process of choosing the filters is depicted in Figure 1.

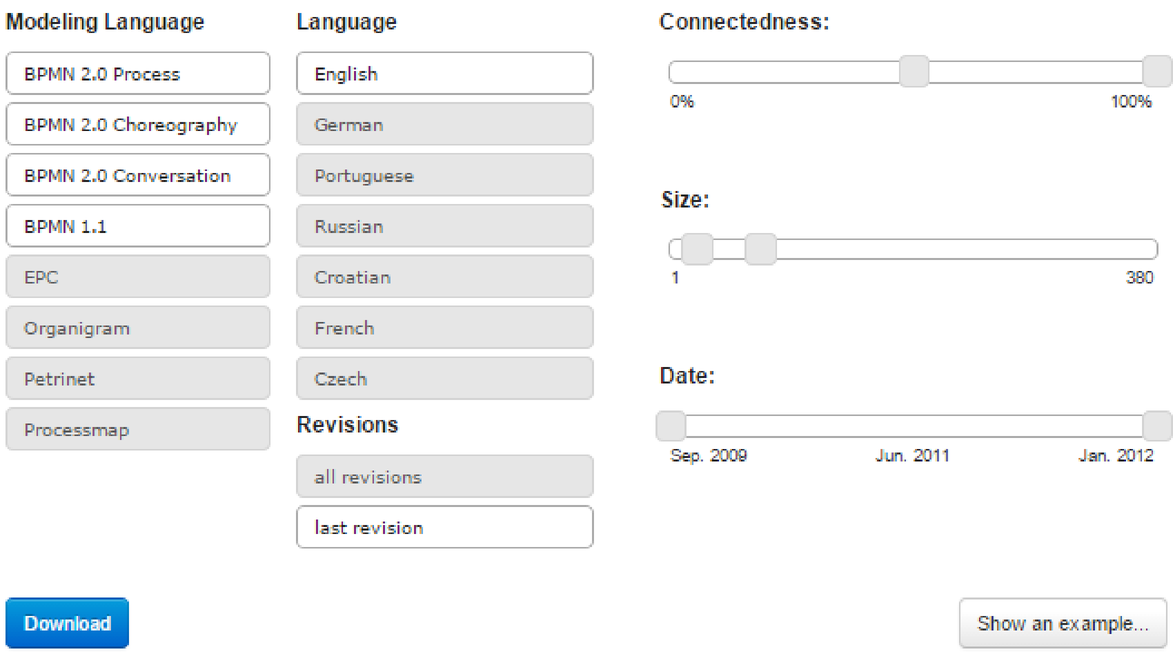

Figure 1. Downloading models.

First of all a small collection of models was downloaded to get a general impression of how the system works and what the output looks like. Each model was organised into a folder, which consisted of the date the model was made and the title. Every folder contained a JSON-file, which held the data about the design of the model, and a SVG-file, which provided a visual overview of the model.

The next step was to search for the necessary models which were actually going to be used. Initially, the models written in English with connectedness over 50% and size of at least 190 nodes were filtered. The hope was that selecting only the bigger models would result in fewer models and thus allow to select the models that originated from the same case. Unfortunately, this did not turn out to be the case. Instead, it became apparent that almost all of the models were made by the same person or people and simply contained different versions of the same model. Furthermore, due to some unknown reason about a third of the models neither had any labels on the model elements nor titles of the model in the folder name. For those instances, this job was nearly impossible.

After realising this it was decided that the search should be broadened by accepting models with connectedness of 50% - 100% and size of 10 or more. The date filter was set to from September 2009 to January 2012. Most importantly, models made in German were included this time. This search resulted in a little over 10,000 models. This query already returned much better models to work on. Compared to English models, the German models had a higher proportion of models that retained included the labels. Eventually it was decided that different filter options had to be tested to find the most appropriate models.

To enhance the filtering even further, a Java program was written to structure the models in a more orderly fashion. One function of the program relocates all models from its directories sorted by author into a general folder containing all of the models. Since the author could not be identified anyway, the loss of the identification of the author had no negative impact. Using this, all the models that still had their original labels and could now be put in exactly the same folder for easier analysis. Furthermore, the models were also automatically sorted by date. The second function of the program finds models containing a specific keyword and puts them into a separate folder. This eased the process of finding a large group of related models considering many of the models included the same keyword in their title. These keywords contained words such as “Exercise”, “Assignment” or “Übung”. This was the most decisive factor in finding usable models from BPMAI.

After finding the optimal query and a method to properly organise the models, the search for similar models began. From some of the names of the models mentioned previously it was assumed that a large number of business process models which are based on the same scenario are homework. This indeed turned out to be correct, as there was a great deal of models made on the same date and having the same keyword in their title.

One of the tools that streamlines the process of examination of models is Promnicat. [19] Promnicat is a framework developed in Germany that is used to import various business process models into one generic database. This enables research on the entire collection across the boundaries of business process modelling notations. Additional features include the analysis of properties of process models. The most significant properties include clustering, label analysis, model alignment and metrics. Promnicat only analyses business process models that are saved in JSON-format. One of the most popular modelling programs, Signavio, allows its process models to be saved in that format. The second widely used business process modelling program, Bizagi, however, does not. Thus, in this study Promnicat was used to occasionally calculate various metrics of business process models from BPMAI to get an insight into the models, although it was unsuitable for anything more.

Before commencing with the analysis, various hypotheses were formulated based on previous experience with business process modelling. The hypotheses are based on different aspects of business process models. The formulated hypotheses are as follows.

- H1. Novices do not always use obligatory start and events.
- H2. Novices use gateways incorrectly, such that extra, unnecessary flows are created.
- H3. Novices confuse the usage of message and sequence flows.
- H4. Novices do not use message flows to transfer messages across pools.
- H5. Novices mix up the throw and catch system.
- H6. Novices do not label the message flows with the necessary message name.

After the hypotheses had been formulated, the analysis of the models was started. The analysis of the models was used to pronounce a verdict regarding the hypotheses.

3.2 Analysis

The basis of this section was taken from a blog post [20] by Business Process Watch, hosted by Bruce Silver. This business specialises in BPMN method and style training, hosting a list of rules a modeller should always follow. The page includes two types of rules: the official rules from the official specification of BPMN and the "method and style" rules used by Silver himself.

The method of inspecting the mistakes made by novices was as follows.

Firstly, the models to be inspected had to be found. It was decided that models which were made by students were the best candidates – this guaranteed that the modellers at least have some experience regarding the practice of modelling. Furthermore, the elements in these models are properly connected and syntactically correct. Typically, these models were exercises and homework given to the students.

Secondly, a list of all the errors that were encountered was made. For every error, the models the error appeared in was included. Thus, it was possible to get a good overview of the most frequent mistakes novices make.

After looking through a set of 9 models trying to find errors in each model it became apparent that many of the rules mentioned previously were never broken in any of the models. Thus it was decided to exclude these rules from further inspection in order to simplify the process, since the probability of them reoccurring was rather marginal. Interestingly, there were also rules to which none of the models adhered to. Afterwards a more thorough analysis of a bigger set of models was conducted.

For all the models inspected the proportion of average number of errors in the models was measured. Figure 2 lists the distribution of average number of faults in a model.

Additional measurements were conducted which analysed how many models a specific rule was violated in. The results of that is displayed in figure 3.

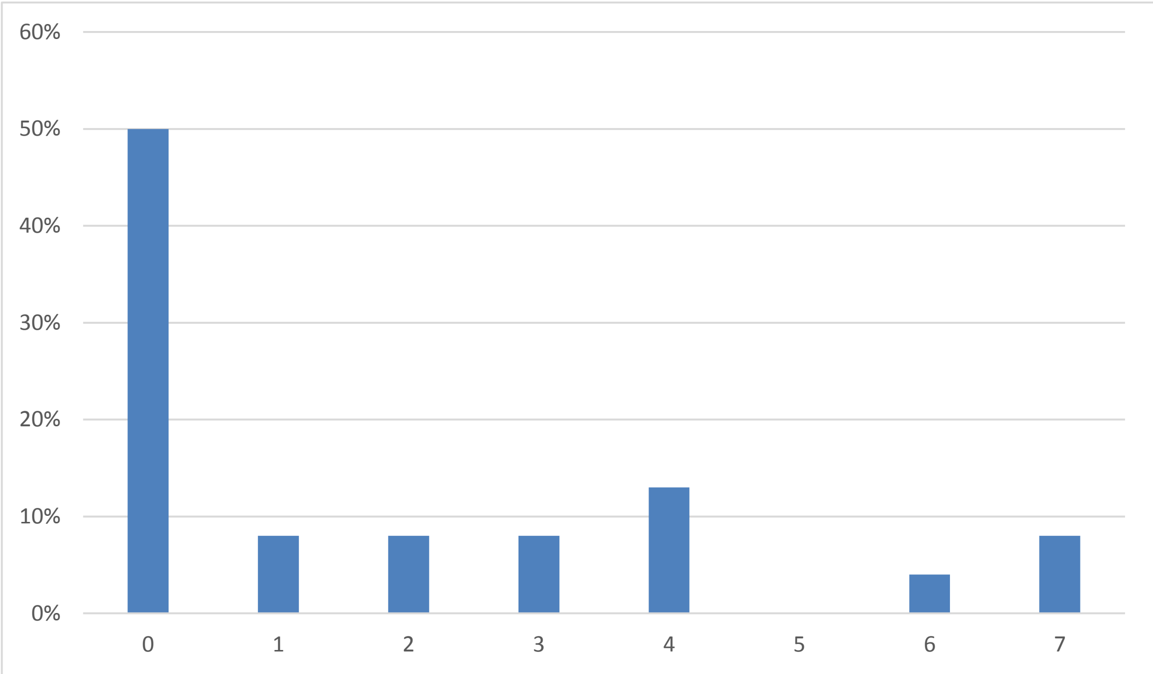

Figure 2. Distribution of average number of faults in a model.

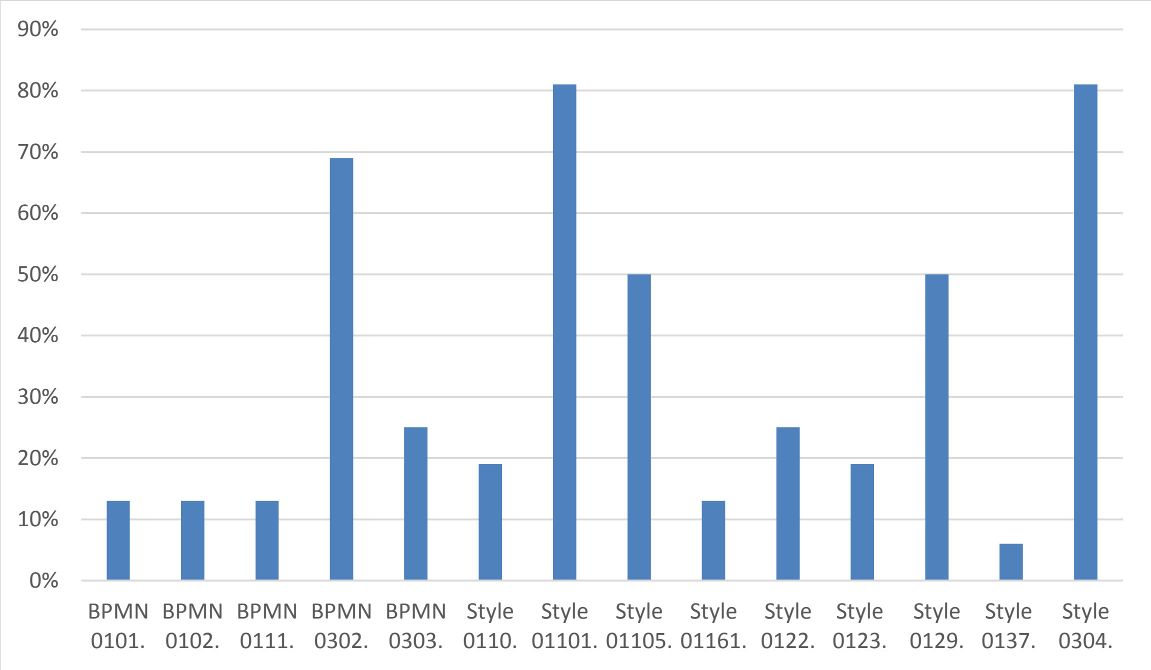

Figure 3. Rules and the frequency the models failed to follow them.

3.3 Discussion

We can conclude a lot from the list of rules and how often the rules are broken when modelling.

The analysis shows that from the list of all the rules, style rules are more often violated than syntactic rules of BPMN. In fact, of all rule violations encountered, only 39% of the errors belong to the syntactic rule group. The majority, 61% belong to the style group. This makes sense as most modellers would be aware of the essential syntax of BPMN, however they might not know the good style of modelling business processes. Another reason might be that the guidelines listed by the author of the style rules are neither vigorously enforced nor well-known.

50% of the models that were reviewed contained at least one error: 8% contained just one error, 8% contained two different types of errors, 8% contained three errors, 13% contained four types of errors, 4% contained six errors and 8% models displayed seven different errors. The rest of the models did not contain any errors. From this it is possible to deduce that it is surprisingly unlikely that a random model will be entirely error-free.

Two BPMN rule stand out amongst other rules – BPMN 0302 and BPMN 0303. Of all the syntactic rules, these rules are the two most often broken – BPMN 0302 is broken in 69% and BPMN 0303 in 25% of the models. Together they make up 71% of all BPMN errors broken. BPMN rule 0302 states that a message flow can only come from a message end or intermediate event; send, user, or service task; subprocess; or black box pool. BPMN 0303 requires that a message flow only go to a message start or intermediate event; receive, user, or service task; subprocess; or black box pool. It is interesting to observe that of all BPMN rules, this one is the most widely broken rule. The explanation for it is that oftentimes, BPMN modellers are not interested in making a differentiation between different types of tasks in business-oriented process models. It might be perceived that such nuances should be introduced only when the models are destined to be used to implement an IT system. Users of BPMN predominantly write models that are there to understand the business process but not necessarily to implement an IT system to support the process.

The BPMN rules BPMN 0101 and BPMN 0102 specify that all flow objects, with a couple of exceptions, must have an incoming and an outgoing sequence flow. More precisely, These rule are quite fundamental – they state that all the components in the model must be connected in order for the model to follow a proper flow. It was very intriguing to discover that even though the rules are very elementary, some modellers still failed to adhere to these basic rules. Despite the fact that some modellers are indeed unaware that all components must be connected, these mistakes most likely happen because modellers fail to pay enough attention to properly connect the components.

The last BPMN syntactic rule that was violated was BPMN 0111. It states that a start event in a subprocess must have a None trigger. Instead of having a None trigger, some models had a specific trigger type selected. In 90% of the cases, the trigger type was message event. In those instances, the modellers intended to emphasise that the process continues after receiving a message. Nevertheless, this is an improper way of using start events in subprocesses.

Two of the style rules that are predominantly breached are rules 01101 and 0304. The two of them make up nearly half of all broken style rules. The next most frequent rules being broken are style rules 01105 and 0125, each appearing in 50% of all the models. The remaining rules account for less than a quarter of all style rule violations.

First of the two prevalent rules, style rule 01101, states that a message start event should be labelled “Receive [message name]“. The reason this rule is breached so frequently is most likely because modellers are simply not taught to label the message events. Although some of the models did include labelled message start events, instead of the format “Receive [message name]“ they used “[message name] received“. Nonetheless, in the current circumstances this format was considered erroneous as well.

The other most breached rule is style rule 0304. It requires that a message flow be labelled with the name of the message. Curiously, a small amount of models that chiefly did label their message flows left out the label for some message flows. The explanation for this could be that the modellers simply fail to notice the missing label.

Style rules 01105 and 0129 state that all start and end events should be labelled. For both of these rules, 50% of the models failed to follow them. There are different opinions regarding if and when start and end events need to be labelled. Whereas the author of these style rules states that all start and events should have a label, the Bizagi article “Best practices in modelling” [21] declares that when start and end events only have one instance of them being used, they are not to be labelled. However, the events in the models that were reviewed had multiple start or end events or were not labelled at all. Of these two different regulations, it is recommended to follow the guidelines of the Bizagi guide.

The rest of the style rules accounted for less than 25% of all style rule violations.

Also noteworthy are style rules 0110, 0122 and 0123. All these rules govern the use of message flows together with message events. More precisely, rule 0110 says that a message start event should have an incoming message flow, rule 0122 states that a catching message event should have incoming message flow and rule 0123 says that a throwing message event should have outgoing message flow. Combined, these three make up 18% of all style rule violations. Most of the time when these rules are broken the modellers add message events and keep the flow within the limits of the same pool with the use of sequence flows. This is incorrect, as throwing and catching message events must have message flows that travel across pools connecting them. It was furthermore observed that the usage of message and sequence flows is often subject to much confusion among modellers. Modellers frequently use sequence flows where a message flow is more appropriate and vice-versa.

Based on these findings, a verdict can now be pronounced regarding the hypotheses.

- H1. Novices do not always use obligatory start and events – False. The models revealed that novices do indeed use events to indicate the start and end of processes. None of the checked models lacked start or end events.
- H2. Novices use gateways incorrectly, such that extra, unnecessary flows are created – False. The gateways in the models were all properly used, there were no redundant flows attached to them.
- H3. Novices confuse the usage of message and sequence flows – Mostly false. For the most part, the models displayed proper use of message and sequence flows. This means that inter-pool flows were marked using sequence flows and message flows were used to transfer messages between pools. There were only a few models that used sequence flows for both purposes.
- H4. Novices do not use message flows to transfer messages across pools – True. Instead of using message events to transfer messages between pools, occasionally novices transfer data between message events inside the same pool using sequence flows. They should be using message events and message flows between pools.
- H5. Novices mix up the throw and catch system – False. All of the modellers displayed appropriate handling of throw and catch information – data travels from throwing events to catching events.
- H6. Novices do not label the message flows with the necessary message name – True. In more than 80% of the models the message flows lacked the required message name.

4. COMPARISON OF NOVICE AND EXPERT MODELLERS

This chapter takes a closer look at how to distinguish skilled modellers from novices. The first section describes how the necessary set of models was found for the analysis. This segment also contains hypotheses that were formulated theorising how the models would differ. Section 4.2 lists the metrics that were used to compare the models as well as the results of the measurements. Section 4.3 discusses the results of the measurements comparing expert models to novice models case by case. This is followed by an examination regarding what the reason behind the differences are. Finally, the outcome to the formulated hypotheses was compared and a verdict pronounced regarding them.

4.1 Hypotheses and dataset

The models from BPMAI that were used in the previous chapter were well suited for the analysis of beginner modellers as they were mainly students. However, in order to compare professionals to novices, models with at least one professional as an author were needed. The first attempt was made trying to finding collections of models from various websites on the Internet. This, however this was not very fruitful as the models were either exclusively made by only one type of modellers or there was no way to determine the author type of the models. Instead, the supervisors provided the necessary models for this purpose.

The models that were used in chapter were homework submissions made by students taking the course Business Process Management at University of Tartu during 2009-2013. The objective of this course is to introduce students to the principles and methods of business process management. The course emphasises the role of business process modelling as an instrument to understand and analyse business operations, and to drive the design of Information Technology (IT) solutions to support the automation of business processes. The models were organised into pairs, which consisted of one model produced by a team that included at least one expert user, and one process model produced by a team of pure novices. The “pure” novice process model was the most error-free model from a pool of a dozen such alternative process models. Here an expert modeller is defined as one who has modelled business processes in a professional environment.

In total four different modelling scenarios were obtained, of which two contained two scenarios – an As-Is scenario and a To-Be scenario. Thus, in total six model pairs compared. The As-Is is a detailed presentation of the process as it is performed currently. It collects data and highlights opportunities for improvement. The To-Be is either a new process addressing the problems from the As-Is analysis or a completely new process design that takes into account information from the As-Is analysis. We acknowledge that this study bears certain limitations as the number of examined models is rather low.

Although there are many modelling languages in existence, such as BPMN, YAWL and EPC, the most prevalent language used in the University of Tartu is BPMN. This was also the format used to create most of the models. There was no universal composition for the homework as they all contained different files. Each homework had a PDF-file, some also included a word-document. A few homework also included a Bizagi model. For some assignments, a presentation was also necessary – in this case a PowerPoint presentation was annexed.

Before commencing with the analysis, hypotheses were put forward, predicting how experts and novices would differ.

- H1. Professionals build their models with less nodes, as they know how to express what is needed with the least amount of clutter possible.
- H2. Models made by professionals are less dense as they tend to focus on one, primary scenario.
- H3. Professionals make less syntactic errors as they are familiar with the essential BPMN modelling syntax.
- H4. Professionals make less style errors because they’ve worked in this field for a while.

4.2 Analysis

The first step in measuring quantitative measures was to determine which measures to actually use. Ultimately, two groups of measures were chosen for model analysis.

The first body of measures was a number of rules a modeller should always follow. The rules were listed and then research was carried out comparing how often they were broken by novices versus how much they were broken by experts. The rules that were used here were the same ones as in the previous chapter. There the occurrences of rules being broken was counted. Thus, it is possible to measure the number of times that the rules are broken by novices and then compare it to how much the experts break them. In addition to the rules mentioned before, some new rules were added – these were primarily generated based on simple inspection of the models in the collection spotting obvious, recurring errors.

The second group of measures can roughly be broken down into general characteristics of models and the elements they contain.

Counting the elements in models was pretty straightforward. These include things like subprocesses, comments, usage of labels etc. For each modelling element the number of times they occurred in each model was counted and then conclusions were derived based on that.

The general characteristics were found from looking at already existing papers listing methods of measuring models. The main source was Jan Mendling [3]. He listed a total of 28 metrics used for measuring business process models. Of those, 15 are related to size and 13 to the structure and state of models. The models were inspected and the ones most suited for the analysis had to be chosen. This meant that they had to be simple enough for measuring and comparing, yet also reasonably meaningful. Of the 15 size metrics 4 were selected for use: size, diameter, density and coefficient of connectivity.

All the metrics were calculated from the process models granted by the supervisors. A total of 3 different scenarios were included in this example of differences between the models. 2 of them has 2 models inside them – one As-Is model and one To-Be model.

The first, most basic measure for models is size. Size is regarded by many as a simple and effective method for model comparison and analysis. Size is simply the number of nodes in the model. These include events, gateways and tasks. The comparison of the sizes is listed in table 1.

Table 1. Comparison of size of models.

Size, Size	Novices	Experts
ETMproject	60	63
Pharmacy prescription as-is	52	68
Pharmacy prescription to-be	66	53
Project2011 as-is	73	73
Project2011 to-be	83	59
ProcurementProcess	74	119

The second measure chosen was diameter. Diameter is the length of the longest path from the start node to the end node in a business process model. In this case for every model the number of edges between the longest paths was counted Diameter is also quite a popular measuring tool, however it comes with some drawbacks. For example, consider two example models – the first only has one, long path whilst the second has multiple alternatives running along parallel paths. In this case the diameter would state that the first models is "bigger" than the second, although the second one is much bigger size-wise. The comparison of the diameters is displayed in table 2.

Table 2. Comparison of diameter of models.

Size, Diameter	Novices	Experts
ETMproject	26	40
Pharmacy prescription as-is	41	56
Pharmacy prescription to-be	47	48
Project2011 as-is	56	46
Project2011 to-be	64	40
ProcurementProcess	51	88

In addition to the two metrics above, which belong to the "Size" category, two metrics were also selected from the "Density" category. Density is a term that refers to metrics that relate numbers of nodes to numbers of edges.

The first metric in this category shares the name with the category – density. Density refers to the number of edges divided by the number of the maximum number of edges for the same number of nodes. Density figures are listed in table 3.

Table 3. Comparison of density of models.

Density, Density	Novices	Experts
ETMproject	61/60*59 (0.017)	70/63*62 (0.018)
Pharmacy prescription as-is	51/52*51 (0.019)	74/68*67 (0.016)
Pharmacy prescription to-be	69/66*65 (0.016)	62/53*52 (0.022)
Project2011 as-is	81/73*72 (0.015)	62/73*72 (0.012)
Project2011 to-be	96/83*82 (0.014)	56/59*58 (0.016)
ProcurementProcess	74/74*73 (0.014)	132/119*118 (0.009)

The second density metric is called the coefficient of connectivity (CoC). It is calculated simply as the ratio of edges to nodes. The comparison of coefficient of connectivity is displayed in table 4.

Table 4. Comparison of coefficient of connectivity of models.

Density, CoC	Novices	Experts
ETMproject	61/60 (1.017)	70/63 (1.111)
Pharmacy prescription as-is	51/52 (0.981)	74/68 (1.088)
Pharmacy prescription to-be	69/66 (1.045)	62/53 (1.170)
Project2011 as-is	81/73 (1.110)	62/73 (0.849)
Project2011 to-be	96/83 (1.157)	56/59 (0.949)
ProcurementProcess	74/74 (1.000)	132/119 (1.109)

In addition to analysing the differences between size and density of models, the groups were also compared regarding how they differ with regard to adhering to the rules of BPMN. For each rule the number of models that displayed errors were counted. The results of this are displayed in figure 4.

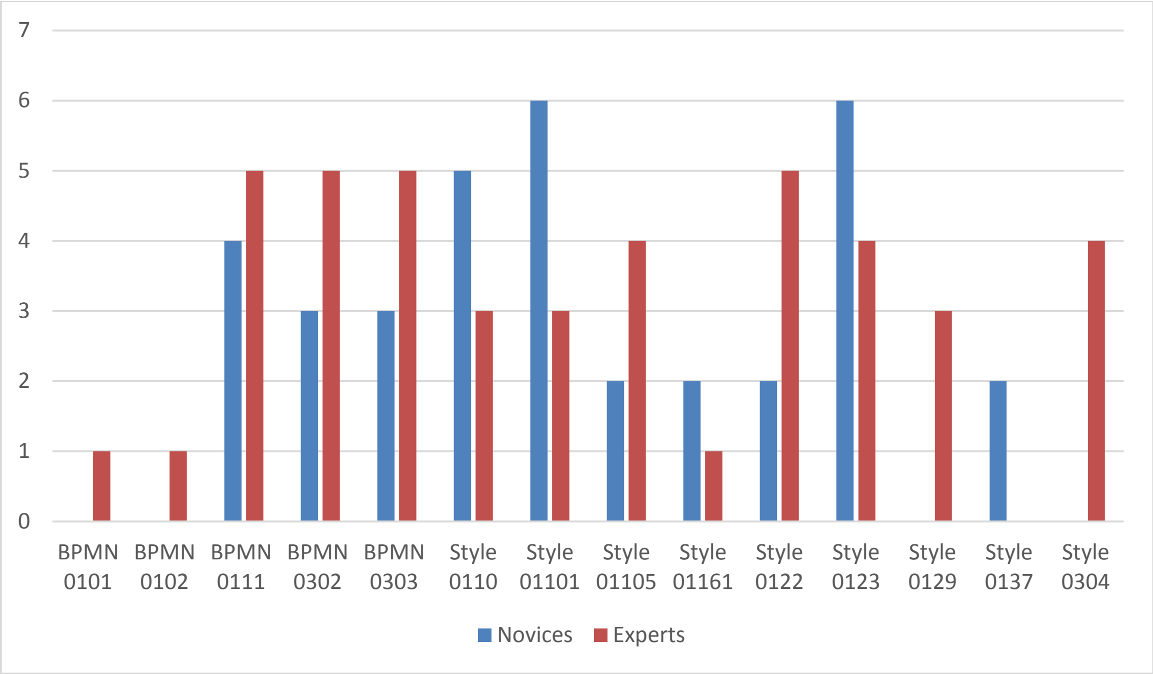

Figure 4. Comparison of number of times BPMN rules were broken.

4.3 Discussion

This chapter discusses the differences between models made by novices and experts. The models in each scenario are taken side-by-side and compared to discover how they differ. Only the most distinguishing facts are brought up in each case. Differences that were negligible are not included in the following analysis. These differences include aspects such as properly connecting all elements with sequence flows, usage of boundary events, different types of gateways and default sequence flows as well as lanes, milestones and groups.

4.3.1 ETMproject

The ETMproject model is a business process model about handling insurance claims.

In this pair of models the most intriguing difference is that experts have problems with using message and sequence flows correctly. The major issue is that the proficient modellers drew message flows originating from and leading to tasks that were not specialised as the task type was not specified. If the message flow is modelled to come from or go to tasks then the tasks must be specified as being a send, user or service task. The novices, on the other hand, avoided using message flows between task altogether and instead used message events.

Another interesting feature that became apparent was that message start events were not used properly. This time, the novices were the ones who made the mistakes. The message start events that they used did not have an incoming message flow, which is always a requirement. Additionally, the message start event did not embody a label as per the BPMN rules. The experts used start events flawlessly.

When observing how frequently sequence and message flows are crossed the experts crossed flows on four occasions. The novices, however, managed to build their model so that all flows successfully avoided crossing each other. Generally it is a good practice to avoid crossing flows, so it was quite surprising to see this phenomenon.

The final notable contrast between experts and novices is the way the process end events and tasks are labelled. Not having the top-level process end events labelled with the end state was frequent occurrence among all novices. The experts, on the other hand, used labels for all end events. Regarding tasks, the novices only labelled their tasks using the approved “verb + object” format in 23% of the cases. All those cases occurred in the top-most process chain which breaks the whole process down into many sub-processes. For example, instead of naming a sub-process “Handle claim” it was titled “Claim handling”. Every task in those sub-processes was labelled following the convention, though. In the model made by experts, all tasks followed this guideline appropriately.

When we take a look at the various elements the two groups use we can see some distinctions. The model made by novices included just one comment, whilst the one by skilled modellers contained as many as 18 comments. It should be noted, however, that the experts also used comments to label their start and events – these were the source of half of the comments.

In addition to comments, experts also used a whole lot more data objects. While novices did not use any, the experts used 20 instances of data objects. The reason why they are completely ignored by amateur modellers is presumably that they don’t know how to properly utilise them and as such choose to exclude them from the models.

4.3.2 Pharmacy prescription

This project is about improving the process of how a drugstore operates. The project encompasses both an As-Is as well as a To-Be scenario.

As this scenario included a large use of subprocesses, both experts and novices had to choose the proper start event as a starting point. In the majority of the cases, however, both experts and novices did not specify the event types. The only instances message events were used was when the pools had to exchange information between each other. In some cases, even timer events were used, which is also erroneous.

Similarly to the previous modelling scenario, experts yet again show problems in using message and sequence flows properly. If message events are used, they must have incoming and outgoing message flows. This was not a problem for novices, who correctly connected their message events. The experts, however, used sequence flows in more than one case instead where the message flows usually came from and lead to unspecified tasks. The novices used message events instead of tasks, most likely recognising that above everything else they should try to keep the model simple.

The two groups can also be clearly separated based on the number of crossings of message and sequence flows. Like in the previous scenario, there were no instances of novices crossing the flows, whilst there were ten instances where experts crossed flows.

Differences become truly apparent when we look at how the two groups label their elements. Style rules 01101, 01105, 01161 and 0304 stipulate which elements ought to be labelled and how. Style rule 01101 states that whenever message events are included in a model they must be labelled using the format “Receive [message name]“. Style rule 01105 says that the start events in the top-level process should be labelled. Style rule 01161 states that a catching intermediate event has to be labelled. Regarding all these scenarios, the novices never succeeded in following these rules as they did not label any of the elements mentioned. Regarding rule 0304 though, both groups were similar in that neither labelled their message flows with the name of the message being sent. Regarding labelling tasks, both groups properly labelled their tasks using object + verb.

The last phenomenon worth mentioning in this model pair is the difference of the usage of various modelling elements. Just like with the previous scenario, the number of comments used by experts was decidedly higher – 15 compared to just 1.

Another similarity to the previous modelling scenario is that the novices did not use any data objects, whilst experts used a total of 11 for both As-Is and To-Be scenarios.

When counting the number of task repetition symbols, novices appear to be much more geared towards using alternative methods for symbolising repetition as in this scenario they never used the marking meant for this purpose. In all 7 situations where repetition was needed, the experts did use the necessary symbol.

The last distinction that can be highlighted is the specialisation of events – the novices had a higher proportion of events that were distinguished using one of the event types. This held true for both the As-Is and the To-Be scenario.

4.3.3 Project2011

This project requires students to design and automate an As-Is and a To-Be business process model for a procure-to-pay process for hiring heavy equipment at a construction company.

In this model pair both experts and novices yet again fell victim to incorrect usage of message flows. In this sample, they used message flows to connect data objects instead of using association. Association differs from message flows in that message flows are directed, whilst associations are not. Furthermore, the majority of the message start events of both groups in the models did not have an incoming message flow. The reason so many errors occurred in this scenario is that many subprocesses began with message events and thus were unsuitable for incoming message flows. The experts do differ from novices in that the proportion of errors was somewhat higher compared to novices.

Surprisingly, the overall model structure is marginally better implemented by the experts this time as they had fewer total crossings – 11 compared to 14.

When inspecting the labelling of the elements, both groups failed to properly follow the labelling guidelines. Both failed to title their message start events, catching intermediate events etc. One eye-catching contrast is that novices properly labelled all their message flows with the message name as necessary, while the experts did not label any of the message flows in either of the scenarios. All of the tasks were labelled using the proper object + verb formula by both user groups.

Yet again, novices show discrepancies from experts when using various modelling elements. First off, the experts used an abundance of comments, however they inexplicably used the commenting system to label all their elements instead. Meanwhile, the novices did not use any comments at all.

The first dissimilarity between this and the previous scenarios is the usage of data objects. The novices modelled five data objects into their models, whilst experts used just one less data object. What's more, we see the first usage of data stores by experts as they included five of these elements in their To-Be model. The novices did not use any data stores.

A stark contrast between previous models and this one is task repetitions. Previously, novices had completely ignored task repetitions. This time, the situation was reversed and the novices were the ones who used the special repetition marking. The experts found alternative methods for implementing repetition.

Finally, the proportion of specific event types is quite similar to previous scenarios. The novices' events are more often designed with an event type, while experts mostly kept to using default events.

4.3.4 ProcurementProcess

This scenario is about modelling an As-Is procurement process for a large company that wants to conduct a review of its current procurement process.

When inspecting how experts and novices use message flows we see that somewhat like previous models, novices and experts fail to properly use message flows and events. Here the novices drew message flows from and lead them to elements that are inappropriate for this purpose. The experts, on the other hand, failed to properly connect all their message start events with the required incoming message flow. Furthermore, they did not attach catching and throwing message events with incoming and outgoing message flows.

In this scenario both the novices and experts failed to design the models in a way that would avoid crossing sequence and message flows. When compared to previous scenarios, though, the total number of crossings is much smaller. Experts crossed flows on three and novices on two occasions.

Labelling-wise, there was not much difference between novices and experts. The only contrast between them was that novices did not label the message start events using the recommended formula “Receive [message name]”. Additionally, all task elements were labelled according to the rules.

Just like in the previous scenarios, comments and data objects are much more common among proficient modellers as they used six comments and six data objects. The difference between the usage of task repetitions and specifying event types is negligible.

4.3.5 Conclusion

Based on the previous analysis it is possible to draw numerous conclusions.

It’s hard to make any conclusions based on the size and density measurements because the sample size is fairly low and the measurements are rather inconclusive.

Both novices and experts seriously struggle with using message flows and message events. When the inspection was initiated it seemed that experts have more problems with message events and flows than novices, however after going through all models it became apparent that novices, too, share this problem. Conclusively, though, in these models the experts still made more errors than novices.

An surprising discovery was that experts constantly cross message and sequence flows in their diagrams, doing it much more often than novices.

Looking at how novices and experts label modelling elements we find no noteworthy differences. Both groups regularly fail to follow the rules regulating proper labelling set by Bruce Silver.

The two groups differ significantly in using various modelling elements. Firstly, experts were much more eager to use comments, data stores and data objects. Secondly, novices used specific event types somewhat more frequently. The use of task repetitions, though, was not substantially different.

Having finished the analysis, we can compare the measurements found in this data analysis, such as size and diameter, to the findings in previous work by Matthias Kunze [4]. Because the dataset used by Kunze contained thousands of models that were hardly complete, the size of the models there was much smaller than of the models used here, just as expected. The same holds true for the diameter of models. As the size of the models is much greater in our dataset the density of the models is skewed towards very small figures. The only feasible measurement that can be compared is thus coefficient of connectivity. The average CoC value in our models was 1.051, while the average CoC of Kunze's dataset was just 0.79. With this we can say the models in our analysis were very dense.

Lastly, we can pronounce verdicts regarding out hypotheses formulated previously.

- H1. Professionals build their models with less nodes, as they know how to express what is needed with the least amount of clutter possible – Inconclusive. Our results are hardly conclusive as on occasion novices made bigger models than experts and vice-versa. This held true for both size and dimension of the models.

- H2. Models made by professionals are less dense as they tend to focus on one, primary scenario – Inconclusive. Again, the data we have is inconclusive as some novice models were less dense than experts' and some were more dense.
- H3. Professionals make less syntactic errors as they are familiar with the essential BPMN modelling syntax – False. Remarkably, in our case the experts actually made more syntactic errors than novices.
- H4. Professionals make less style errors because they've gathered plenty of experience and know to avoid these errors – Inconclusive. Due to the lack of conclusive data it is not possible to pronounce a verdict regarding this hypothesis.

5. CONCLUSION

This thesis dealt with two problems in the field of business process modelling – determining what errors novice modellers make most frequently and comparing how business process models made by novices and experts differ.

Research was done aimed at answering the question “What typical mistakes are made by novice process modellers?”, which was raised in the beginning. It was discovered that the novices primarily commit errors in the field of labelling modelling elements. For example, a message start event should be labelled using the formula “Receive [message name]“ and message flows should be labelled with the name of the message. In most cases, novices failed to adhere to the labelling principles. Errors were also made because novices do not know when to use message flows and when to use sequence flows very well.

Models by novices and experts were taken under inspection to answer the question “What distinguishes process models produced by novices with comparable process models produced by experts?”. Firstly it should be mentioned that this analysis bears limitations as the total number of models analysed is fairly low. The study showed that both groups produce similar models based on size and density. The groups do show some contrasts when comparing what elements they use, however. Experts tend to use comments, data objects and data stores much more frequently and novices tend to use specific event types instead of retaining the default event marker.

It is possible to build upon the work done by expanding the sample size and conducting measurements upon the new dataset to hopefully obtain noticeable differences between the modelling characteristics of novices and experts. Most importantly, it might give some real differences between the size and density of models.

REFERENCES

[1] Dinesh Batra, Joseph G. Davis: Conceptual Data Modelling in Database Design: Similarities and Differences between Expert and Novice Designers. International Journal of Man-Machine Studies 37(1): 83-101 (1992)

[2] Jan Recker, Norizan Safrudin, Michael Rosemann: How novices design business processes. Inf. Syst. 37(6): 557-573 (2012)

[3] Jan Mendling: Detection and prediction of errors in EPC business process models. Wirtschaftsuniversität Wien 2007, pp. I-XXVI, 1-491

[4] Matthias Kunze, Alexander Luebbe, Matthias Weidlich, Mathias Weske: Towards Understanding Process Modeling - The Case of the BPM Academic Initiative. BPMN 2011: 44-58

[5] Matthias Kunze, Philipp Berger, Mathias Weske. BPM Academic Initiative - Fostering Empirical Research. In Proc. of BPM Demos, CEUR Workshop Proceedings, Vol. 940, pp. 1-5, 2012

[6] Jan Mendling, Hajo A. Reijers, Wil M. P. van der Aalst: Seven process modeling guidelines (7PMG). Information & Software Technology 52(2): 127-136 (2010)

[7] Darius Silingas, Edita Mileviciene: Refactoring BPMN Models: From ‘Bad Smells’ to Best Practices and Patterns. BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Modeling Notation (BPMN): 125-134 (2011)

[8] Bruce Silver: BPMN method and style, Second Edition. 2011

[9] Bruce Silver: Business Process Watch. www.brsilver.com/, accessed on 21 May 2015

[10] Michael zur Muehlen, Jan Recker: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. CAiSE 2008: 465-479

[11] Michael zur Muehlen: “The Great BPMN Debate.” www.column2.com/2008/03/the-great-bpmn-debate/, accessed on 4 May 2015

[12] Remco M. Dijkman: A Classification of Differences between Similar BusinessProcesses. EDOC 2007: 37-50

[13] Chen Li, Manfred Reichert, Andreas Wombacher: On Measuring Process Model Similarity Based on High-Level Change Operations. ER 2008: 248-264

[14] Matthias Kunze: Searching Business Process Models by Example. 2013

[15] Michael Becker, Ralf Laue: A comparative survey of business process similarity measures. Computers in Industry 63(2): 148-167 (2012)

[16] Remco M. Dijkman, Marlon Dumas, Luciano García-Bañuelos: Graph Matching Algorithms for Business Process Model Similarity Search. BPM 2009: 48-63

[17] Juntao Gao, Li Zhang: On Measuring Semantic Similarity of Business Process Models. IESA 2009: 289-293

[18] Tao Jin, Jianmin Wang, Lijie Wen: Efficient Retrieval of Similar Business Process Models Based on Structure - (Short Paper). OTM Conferences (1) 2011: 56-63

[19] Cindy Fähnrich, Tobias Hoppe, Andrina Mascher: PromniCAT - Collecting and Analyzing. Heterogenic Business Process Models.

[20] Bruce Silver: “The Rules of BPMN.” www.brsilver.com/the-rules-of-bpmn/, accessed on 21 May 2015

[21] Bizagi: “Best practices in modeling.” help.bizagi.com/processmodeler/en/faq.htm, accessed on 21 May 2015

LICENCE

Non-exclusive licence to reproduce thesis and make thesis public

I, **Mihkel Alavere** (date of birth: 21.05.2015),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

On the Differences Between Process Models by Novice and Expert Modellers,

supervised by Marlon Dumas and Matthias Kunze,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **21.05.2015**