From PDM to Process Design

How to involve the users in a redesign project

By J.J. Theelen

in partial fulfilment of the requirements for the degree of

Master of Science
in Business Information Systems

Supervisors:

Dr. Ir. H.A. Reijers (TU/e)
Drs. P. Eertink (Pallas Athena)
Dr. Ir. N. Sidorova (TU/e)
Dr. Ir. I.T.P. Vanderfeesten (TU/e)

Eindhoven/Apeldoorn, March 2010
Preface

Finally, I am writing the last words of this thesis. This will conclude my master Business Information Systems (BIS) at Eindhoven University of Technology (TU/e). It took me another 3,5 years to finish my master, after I finished my bachelor at Fontys University of professional education in 2006, but it was definitely worth it. I had a fabulous time, but most important, I learned a lot.

But I could not have done it without support. First of all, I would like to thank my supervisors of this master project. My special thanks go to Hajo Reijers, who offered me the possibility to conduct this master project at the Information Systems department of the TU/e. I also would like to thank Irene Vanderfeesten and Natalia Sidorova for their role as supervisor and for their support.

Furthermore, I would like to thank all my colleagues from Pallas Athena. I had a wonderful time in Apeldoorn. Special thanks go to Paul Eertink, for his excellent supervision and Mark van Roermund for proofreading my thesis.

Of course, I would also like to thank Jan Vogelaar from the province of North-Brabant. He made the evaluation of the developed method possible. Without his support and contribution, I would not have been able to deliver this work.

Finally, I would like to thank all my friends, family and my girlfriend for their support. I really enjoyed it.
Abstract

Business Process Management (BPM), as the name suggests, a management approach that is focused on the alignment of business processes. Over the previous years, the interest in BPM has grown, which resulted in different research areas. One of these areas is Product Based Workflow Design (PBWD). PBWD is an upcoming practice in the field of Business Process Redesign (BPR).

Since PBWD is a practice in the field of BPR, its focus lies on the redesign of business processes. In a PBWD project, the product structure serves as a starting point for the redesign process. This product structure is called a Product Data Model (PDM). A PDM is comparable with a Bill-of-Material, which is used in the manufacturing industry. A PDM explains how the product is composed out of several half-products.

Since the PDM describes how the product is composed, it fills a gap that is missing in most other BPR projects. PBWD is however not yet a widely accepted approach. A cause can be that the PDM explains how the product is composed, but it does not tell what the process to compose the product should look like. There is currently no user-friendly method that guides the conversion of a PDM into a process design. The goal of this research project is therefore to develop a user-friendly method to convert a PDM into a process design. A user-friendly conversion method should encourage the use of PBWD.

This research project resulted in a conversion method that is based on the direct execution of a PDM. The method treats the PDM as an implicit process model. Such a model offers more flexibility than desired in a final process design. Therefore the method aims at limiting this flexibility in a workshop setting. In these workshops, the dynamics of historic cases are examined and processed by the tooling developed for this method. This helps the participants in deciding where the flexibility should be limited, to offer better support.

The evaluation shows that the method offers better support to convert the PDM into a process design. Some points of the method can be improved, to make the method even more user-friendly. However, the method already offers better support compared to the old situation, especially in comparison with a manual conversion. It does however also show that more research in the direction of PBWD is desirable.
## Contents

1 Introduction.......................................................................................................................... 8
  1.1 Background assignment ................................................................................................. 8
  1.2 Objectives ....................................................................................................................... 8
    1.2.1 Problem description ................................................................................................. 8
    1.2.2 Goal and research questions ................................................................................... 9
  1.3 Approach ......................................................................................................................... 10
  1.4 Outline ........................................................................................................................... 11
2 Preliminaries .......................................................................................................................... 12
  2.1 BPM and BPMS ............................................................................................................. 12
  2.2 Process models .............................................................................................................. 12
  2.3 Petri nets ......................................................................................................................... 13
  2.4 PBWD and PDM ............................................................................................................ 13
  2.5 Case handling systems .................................................................................................. 14
3 Background .......................................................................................................................... 16
  3.1 From PDM to process design ......................................................................................... 16
  3.2 Visualizing dynamic behavior of (process) models ....................................................... 17
  3.3 PBWD support in case handling systems .................................................................... 19
  3.4 Communication in a BPM project .................................................................................. 19
  3.5 Summary ....................................................................................................................... 20
4 Design .................................................................................................................................. 22
  4.1 The approaches ............................................................................................................. 22
    4.1.1 The first approach: Visual support for selecting the best process design .............. 22
    4.1.2 The second approach: Using workshops to obtain the best process design ......... 24
    4.1.3 The third approach: Using interviews to obtain the best process design .......... 26
    4.1.4 Selected approach ................................................................................................. 27
  4.2 Detailed design of the method ...................................................................................... 28
    4.2.1 Steps of the method ............................................................................................... 28
    4.2.2 Tool-support ......................................................................................................... 29
    4.2.3 Communication platform of the method ............................................................... 32
    4.2.4 Maintainability ...................................................................................................... 34
  4.3 Example ......................................................................................................................... 35
    4.3.1 Preparation .............................................................................................................. 36
    4.3.2 The first workshop .................................................................................................. 38
    4.3.3 Modeling phase ...................................................................................................... 39
    4.3.4 Approval workshop ............................................................................................... 39
    4.3.5 Maintainability ...................................................................................................... 39
5 Evaluation ............................................................................................................................ 41
  5.1 Goal ............................................................................................................................... 41
  5.2 PDM ............................................................................................................................... 42
  5.3 Evaluation from a process-designer’s point of view ..................................................... 43
    5.3.1 Tool-support ......................................................................................................... 43
    5.3.2 Communication platform of the method ............................................................... 48
    5.3.3 Main findings ......................................................................................................... 50
5.4 Evaluation with the participants ................................................................. 51
5.5 Results ......................................................................................................... 53
6 Conclusion and future work ........................................................................ 55
  6.1 The project ................................................................................................ 55
  6.2 Comparison with related work .................................................................. 56
  6.3 Future work .............................................................................................. 56
References.......................................................................................................... 58
Appendix A: Method description ...................................................................... 61
Appendix B: Template for change requests .................................................... 67
Appendix C: Tool-support ................................................................................ 68
Appendix D: Cases used in the mined model .................................................. 70
Appendix E: Process design sent for approval (in Dutch) ................................. 72
Appendix F: Evaluation form (in Dutch) .......................................................... 79
1 Introduction

1.1 Background assignment

This thesis is a part of my master project conducted at the Information Systems department at Eindhoven University of Technology (TU/e) in the period between September 2009 and March 2010. It focuses on the area of Product Based Workflow Design (PBWD).

The research is carried out in cooperation with Pallas Athena as an industrial partner. Pallas Athena was selected as research partner because they offered an environment in which research can be combined with a practical business environment. Pallas Athena\(^1\) develops software for Business Process Management (BPM) and offers personalized solutions for its customers. With circa 150 employees it is a medium sized company. Besides Pallas Athena, the province of North-Brabant\(^2\) was involved in this research project. They offered us the opportunity to evaluate our research in a practical environment.

This chapter provides background information related to the project. The problem is described, together with our research goal. Next, we describe our research method and finally the outline of this thesis is presented.

1.2 Objectives

This research project is conducted in the field of PBWD. Understanding the problem description and goal of the assignment described in this section requires a basic knowledge of BPM in general and PBWD in particular. BPM is concerned with supporting business processes. Support can be given by methods, techniques and software to design and control business processes.

Business processes supported by BPM require a regular update (redesign), since rules, regulations and technologies will change over time. Redesigning business processes is known as Business Process Redesign (BPR). PBWD is an upcoming BPR method. It is innovative in the fact that it focuses on the product instead of the process. Most administrative end-products consist of a number of administrative half-products, like an approval. This is comparable with manufacturing environments, in which an end-product is produced out of a number of different physical half-products. Given the number of similarities a comparable way of visualizing the product structure is possible. In a production environment the product structure is called a “Bill-of-Material”, whereas it is called Product Data Model (PDM) in the field of PBWD.

If you are currently unfamiliar with BPM, BPR or PBWD, we suggest to read chapter 2 first. Chapter 2 will give a more complete overview of important definitions and concepts that are used throughout this thesis. If basic knowledge is already available, chapter 2 may be skipped.

1.2.1 Problem description

When the PDM is constructed, it is clear what the product is, and how it is composed out of half-products. This fills a gap that appears in ‘normal’ BPR methods, in which it is not always exactly known how the product is composed out of half-products. However, despite this benefit, PBWD is not yet widely accepted outside the academic world.

\(^1\) www.pallas-athena.com
\(^2\) www.brabant.nl
A possible cause can be the limited support offered to convert a PDM into a process design. There is currently no user-friendly method that guides the end-users and designer in the conversion from a PDM into a process design, although research has been performed in this direction.

Current research has led to two main directions to support the conversion of a PDM into a process design. The first direction focuses on the automatic conversion of a DPM into a process model. There are seven conversion algorithms available that automatically convert a PDM into a process model, but all these algorithms have their limitations. Selecting the best process model is not an easy task. Even if we succeed in selecting the best process model, this model cannot be seen as a final process design. Reasons are that e.g. clustering of data elements is not taken into account by any of the algorithms, the selected process model may have undesired behavior and information related to roles is not available.

The second direction is the direct execution of a PDM. Instead of the conversion of a PDM into a process model, it uses the PDM itself as an implicit process model. In this research project the main focus was on the selection of the most efficient enabled operation. In a (large) PDM, various operations will be enabled simultaneously, so it is important to select an operation that can result in the end-product in an efficient way (so the case should be processed in a near optimal way). To guide the end-users in this process, a recommendation system is used. However, none of the recommendation engines available today performs well in any given situation. The result is either based on a local calculation, or the required calculation can become too time-consuming, especially for larger PDMs. A complete description of these research directions can be found in section 3.1. For now, it is enough to understand that none of these directions is yet completely satisfying.

We expect that the lack of an easy, understandable method to convert a PDM into a process design is the main cause for the limited acceptance of PBWD outside the academic world. The availability of a user-friendly method can result in a wider acceptance of PBWD.

1.2.2 Goal and research questions

The problem description makes clear that PBWD is not yet widely accepted. This research project should lead to better support for the conversion of a PDM into a process design. This should lead to a better acceptance of PBWD outside the academic world.

This resulted in the project goal:

*Develop a user-friendly method that guides the conversion of a PDM into an accepted process design and therefore encourages the use of PBWD outside the academic world.*

In the literature one can find various descriptions of a process design. Our definition of a process design is based on the description of van der Aalst et al. [Aal01]:

*A process design is a combination of a process model, together with a description of the required data for any activity and the roles to process this activity.*

A process model in its turn is a graphical representation of the business process in a business process modeling language.

To guide the conversion in a user-friendly way, better tool-support may be a key to success. A method without tool-support is unlikely to become user-friendly for larger BPM projects. Such tool-support can e.g.
automate the steps required to convert the PDM into a process design, or it can be used to communicate about the (dis)advantages of the flexibility offered by the PDM. Based on the preceding, we formulate the following research question:

**RQ1: How can tooling support the usability of a method that guides the conversion of a PDM into a process design?**

Answering this research question should result in different ways in which tooling can support the steps required to convert a PDM into a process design. This should on its turn result in tooling (a prototype) that supports the method under development.

Furthermore, we believe that the method should support qualitative communication between the end-users and the designer to become user-friendly. Communication is important since the end-users have to accept the process design at the end of the project. Various decisions that have to be made are also depending on the experiences of the end-users that are involved in the process on a daily basis. Moreover, involving them in an early stage creates extra support for the redesign project. This resulted in the second research question:

**RQ2: How can we efficiently support the communication between the process designer and the end-users in the process of converting a PDM into a process design?**

Answering these research questions should lead to a method that improves the usability of PBWD. By improving the usability of PBWD, we significantly increase the change that PBWD will become wider accepted.

### 1.3 Approach

This project was started with a literature review to serve as a profound foundation for our work. No research project can exist without looking at previous work in its area. Based on the knowledge we gathered during this literature review, we developed three possible approaches that could be used to accomplish the goal. The approach that answers the research questions the best was selected to continue with.

After the selection of one of the approaches, the real development of the method was started. The tool-support has been developed in parallel with the development of the method, to make sure that they strengthen each other. After the method was finished, we conducted an evaluation project to examine if the developed method behaves as expected. The conclusions that are presented in this thesis are based on this evaluation project. This research method is also graphically depicted in figure 1.1.
1.4 Outline
In this chapter the setting of this project is described. This includes the goal of the project, the corresponding research questions and the research approach. In the next chapter the preliminaries that are required to understand the rest of this thesis are presented. Chapter 3 presents the outcomes of the literature review and additional background that serves as a foundation for this project. In section 4.1 three alternative approaches to meet the research goal are given, together with the corresponding selection process. The design decisions made during the development of the method are described in detail in section 4.2. The method itself is described in detail in appendix A. Chapter 4 is concluded with an example of a project conducted with the method. Chapter 5 describes the outcomes of the evaluation project conducted at the province of North-Brabant. Finally, conclusions and directions for future work are presented in chapter 6.
2 Preliminaries

In this section the most important concepts of BPM are explained. These concepts are required to understand the rest of the work presented in this thesis. Readers that are already familiar with these concepts can decide to skip (parts of) this chapter.

2.1 BPM and BPMS

As described, BPM stands for Business Process Management. The goal of BPM is the continuous improvement of business processes. According to Davenport [Dav93]:

A business process is a structured and measured, managed and controlled set of interrelated and interacting activities that uses resources to transform inputs into specified outputs (goods or services) for a particular customer or market.

Alternative definitions of a business process exist, but in this thesis we will use the definition of Davenport.

Most BPM projects starts as a redesign project, since its goal is to improve the current situation (as-is situation). BPM is supported by the BPM lifecycle [Aal01]. BPM should include methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. Various papers have various definitions for BPM. In this thesis we will use the BPM definition of van der Aalst et al. [Aal01]:

Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information.

BPM can be supported by Business Process Management Systems (BPMS). A BPMS supports the business processes in an operational way. Also here, various papers present various definitions for BPMSs. In this thesis we use the BPMS definition of van der Aalst et al. [Aal01]:

A generic software system that is driven by explicit process designs to enact and manage operational business processes.

2.2 Process models

BPM aims at the improvement of business processes. Process models are used to describe the behavior of these business processes. A process model is a graphical representation of a business process. This representation can be used in discussions about how a (future) business process should look like. Also in this case, various definitions of a process model appear in the literature. In this thesis we will used the definition of Kalpic et al. [Kal06]:

A process model is the set of relevant characteristics about a business process captured in a structured and documented form. A model is an abstraction process of the real world into a formal representation, where the relevant facts are expressed in terms of some formalism called a modeling language.

A process model is thus an abstraction from the reality, in which a business process is represented in a graphical manner. Various notations exist nowadays to represent a business process. Petri nets (see section 2.3) are an example of such a notation, but also BPMN, Protos Analysis Language, EPCs and many others can be used to model a business process. These notations are called modeling languages.
2.3 Petri nets

Petri nets are used to model business processes. This modeling language is often used, since Petri nets have a clear structure with a formal definition [Mur89]. A Petri net is a directed graph with places and transitions. A transition represents a task that has to be performed. A place basically represents a status. These places and transitions are connected with each other with arcs. Finally, tokens are used to represent the state of a case. A transition is enabled if every incoming arc of a transition is connected with a place that holds enough (at least one) tokens. An enabled transition means that the task that it represents can be performed.

Petri nets are often represented in a graphical way in which:

- A transition is represented by a rectangle
- A place is represented by a circle
- An arc is represented by an arrow
- A token is presented by a dot in a place

In figure 2.1 an example of a Petri net is given. In this example a token is placed in the place before the transition ‘start’. Arcs connect the places with transitions and vice versa. Petri nets do not allow arcs between two places or two transitions.

Figure 2.1: Example of a Petri net

2.4 PBWD and PDM

The goal of BPM is the continuous improvement of business processes as described in section 2.1. Currently, it is common practice to start with the current process in order to locate possible improvements. Such a redesign approach is called evolutionary, since the current process serves as an input for the development of a new process design. A disadvantage of this approach is that it is sometimes hard to locate inefficiencies. Especially after various rounds of the BPM lifecycle it is not always clear anymore why the process is as it is. Asking the people involved in the process can (or even will) result in subjective answers, since they take the current process as a starting point for their reasoning.

PBWD is a revolutionary approach that is focused on the end-product instead of the process. This means that the process is designed from scratch. The most important advantage of PBWD is that the process design will become objective [Van06]. Vanderfeesten gives two reasons for this argument. The first is that the product specification is taken as the basis for the process design. This makes it possible to link every task in the process design with a data element or operation in the product specification. The second is that the ordering of tasks will be based on the product structure. Therefore the process design will become objective, since it does not contain unnecessary tasks [Van06]. A task is unnecessary if it does not contribute to the end-product.
As described, PBWD uses a product structure for the development of a process design. This project structure is called a Product Data Model (PDM). A PDM consists of operation and data elements with arcs between them. These operations and data elements describe how the product is built up of different half-products. A PDM can be compared with a “Bill-of-Materials” (BoM) used in a manufacturing environment. However, a BoM represents a physical product, whereas a PDM represents an administrative product (none physical product).

A good example is an insurance claim. Possible half-products (data elements) of an insurance claim are: name, address, claimed amount, date, insurance number etcetera. The insurance company will use formulae to calculate the amount that has to be paid. The paid amount (end-product) will depend on these half-products (data elements).

The end-product of a PDM is called the root element and the data elements without inputs are called leaf elements. An example of a PDM can be found in figure 2.2 (original source [Kam08]).

![Figure 2.2: Example of a PDM](image)

### 2.5 Case handling systems

A disadvantage of most BPMSs is that they are inflexible, since they are process driven. This process driven structure is called activity-centric [Dum05]. In such BPMS only data that is directly related to the current activity is visible for the user.

Case Handling systems are data driven instead of process driven. They focus on what can be done, instead of what must be done. The case is the central concept in such a BPMS. The core features of case handling are [Van09a]:

- Present the whole case at all times, instead of just some pieces
- Activities are enabled based on the available data, not just on the control flow
- Different role types exist in a case handling system. Besides the execute role, case handling systems have a redo and skip role (names might differ per case handling system, this are the terms used in BPM|one).
- Information can be entered at the moment it becomes available, even if the activity to which this data belongs is not yet enabled.
To control the offered freedom, authorization is an important aspect of case handling systems, to prevent undesired behavior. An arbitrary processing order is standard allowed by a case handling system, but it is possible to limit this flexibility if desired (in most case handling systems).

Our research partner Pallas Athena develops a case handling system, called BPM|one. Previous versions of this software product are known as FLOWer. BPM|one differs from FLOWer in the sense that it has become web based. BPM|one offers various levels of authorization, so the system can become (almost) as flexible as the process designer desires. Case handling systems (BPM|one) solve for this reason the main limitation of an ordinary BPMS.

However, case handling systems, like BPM|one, are especially interesting since they are data driven. This data driven architecture offers potentially good support for PBWD. Research in this direction showed that FLOWer is already capable to handle most constructs of PBWD [Van06]. In the next chapter a more detailed description of case handling systems in relation with PBWD is given.
3 Background
In this chapter the background knowledge gathered during the initial phase of this project is presented. To develop a method that makes PBWD more user-friendly, we needed a decent amount of background knowledge. As preparation for this master project a literature review was conducted. This literature review was focused on earlier PBWD research in general, and on the visualization of dynamic behavior in particular. The choice to focus on visualization was made because visualization offers potentially good support for the comparison of process models. A comparison between different (generated) process models may help to meet this goal. Besides focusing on visualization, we examined which ways to convert a PDM into a process design already exist and how well PBWD is supported by current BPMSs. Furthermore, we investigated what kind of communication can be used to support the method. In the remainder of this chapter we present the background knowledge that influenced the development of the method.

3.1 From PDM to process design
Since the goal of this project requires that the usability of PBWD is improved, we started by focusing on its weak point. The weak point of PBWD is the support that can be given after the PDM has been developed. This PDM should finally result in a process design. It is of course possible to derive the process design manually, but this can be a time consuming and difficult task. It is also possible to derive a process model automatically with a conversion algorithm, or to make the PDM itself executable (in this case the process model is the PDM itself, so it is an implicit process model). None of these techniques is however completely satisfying. In this section we present these techniques and describe the advantages and disadvantages.

In the early days of PBWD, the conversion of a PDM into a process design was a manual and time-consuming task. Examples of projects that used a manual translation can be found in [Rei03]. Since this required quite some time of an experienced process designer, it was desired to automate this process where possible.

Research in the direction of automatically generated process models is described in the paper of Kamphuis et al. [Kam08] and the PhD thesis of Vanderfeesten [Van09b]. In total seven conversion algorithms that automatically convert a PDM into a process model exist. These algorithms are implemented in ProM [Pro10]. Six of them produce a Petri net and the other produces a YAWL (Yet Another Workflow Language) model, which is basically an extended Petri net. However, none of these algorithms is superior in any given situation, because all algorithms have their own advantages and disadvantages.

Based on the strong and weak points of these algorithms, the favorable process model should be selected out of the set of available process models. The framework presented in [Kam08, Van09b] is the best available guiding for this job so far. This framework also reveals that other conversion algorithms can be developed, since only 7 conversion algorithms are developed, where the framework is able to support 144 different conversion algorithms. The framework can be used to discover what kind of disadvantages can appear, since it classifies the algorithms. Examples of disadvantages that can occur in the corresponding process models are that: soundness cannot be guaranteed [Kam08], double tasks may occur, data elements can only be consumed once, and the size of the model is disproportional with its process (too large process model). It is difficult to understand the impact of these limitations for the process designer and end-users. Selecting the best process model is for this reason a difficult task.

Furthermore, none of the generated process models can be seen as a final process design. No clustering is conducted by any of the algorithms and all resulting process models might contain undesired limitations. Clustering means that the operations and data elements of the PDM are grouped in logical units of work.
Without (qualitative) clusters, work may not be processed efficiently. Besides the lack of clustering, the process model does not contain a description of the required data for the activities and the required role to execute the activities. For this reason the automatically generated process models require a manual update afterwards to become an acceptable process design.

Besides the automatic conversion into a process model, research has been conducted on the direct execution of a PDM. This means that the PDM itself is used for the execution of a case, instead of a (generated) process model. Direct execution of a PDM is more dynamic and flexible, since there are no restrictions on the execution [Van09b]. If a data element can be calculated based on the available data, direct execution supports the calculation of this data element. Based on the available information for a case, the ‘best’ next step can be calculated. What is best will depend on the chosen strategy, like lowest cost or shortest processing time [Van08b].

Unfortunately, calculating the next best step is not always a trivial task. Vanderfeesten describes a recommendation system in [Van08b] that can be used to calculate the next best step with various strategies. These recommendations are however based on local decisions. E.g. if the chosen strategy is lowest cost and two operations are enabled, of which the first will cost 1 euro and the second will cost 2 euro, the recommendation system suggest to select the first option. It is however possible that the second option will directly result in the end-product, where the first requires an additional step with a cost of 5 euro. In this case the local calculation does not result in the global best step.

A global calculation can overcome this limitation. Global calculations are possible with Markov Chains [Van09b]. A drawback of a global calculation with a Markov Chain is that it can result in a state explosion. A global calculation is therefore not suitable for larger PDMs. Since a PDM with more than 500 data elements is no exception, this is a serious limitation of direct execution.

In our opinion, the main limitation of an automatically generated process model can be compared with the main limitation of a direct execution, since it is in both cases hard to offer the right level of flexibility, without the chance to result in inefficient executions. In case of a generated process model the level of flexibility is set at design time, where direct execution ‘postpones this choice’ to runtime.

Overall, all support offered to convert a PDM into a process design, can result in restrictions, which prevents PBWD from becoming an accepted approach outside of the academic world. A manual translation is time-consuming. The conversion algorithms do not offer enough support to select the best alternative and even if a generated process models is selected, a manual update is still required. Direct execution solves these limitations, since a conversion is no longer required. Direct execution can be supported by a recommendation systems, that suggest what the next best step is based on strategies like lowest cost or shortest processing time. A local calculation for the next best step can be done without any problem, but this does not necessarily result in the best global decision. The calculation of a global best step is not always possible, since the state space becomes too large.

3.2 Visualizing dynamic behavior of (process) models

During the literature review, we investigated how dynamic behavior of a process model can be visualized. Different process models will offer different dynamic behavior. A full understanding of the dynamic behavior of a process model is essential for making the right choices [Kam98]. Visualization can be used to improve understandability, so an investigation into this direction seemed desirable [Str05]. Understandability proved
to be complex and sometimes hard to measure or predict. However, research in the previous years has resulted in a number of metrics and guidelines to improve understandability with visual support.

Understandability is hard to predict, since two (or more) models may look the same, but still have completely different behavior. An example can be found in the paper of Mendling et al. [Men07]. This paper shows a comparison between 3 different models that look the same, except for one connector. These models can be found in figure 3.1 (original source [Men07]).

![Figure 3.1: Three models that look the same, but with different dynamic behavior.](image)

The connector represents the routing constraints. AND means that all outgoing arcs are processed, XOR means that exactly one of the outgoing arcs is processed and OR means that one or more of the outgoing arcs are processed.

Research conducted with these models proved that there was a great difference in understandability. Especially inexperienced users have difficulties with reading and understanding graphics, like process models, because of their limited cognitive capabilities [Pet06, Gre06]. The more different symbols or constructs, the harder it becomes to understand the meaning of a graphical representation. Experienced users have often learned to link the information, so they are better able to understand complex figures. A simple and standard layout is therefore desirable when it comes to understandability of figures, especially for less experienced users.

A PDM has a simple and standard layout, comparable with a well known Bill-of-Material used in the manufacturing industry. A process model can have more difficult constructs, like a milestone [Wor10]. It may therefore be possible to visualize the dynamic behavior of a process model efficiently in the PDM. Such an approach can be used to show the dynamics of the automatically generated process models. A possible visualization method is described in the literature study belonging to this project [The09].

However, visualization of the dynamic behavior might also have some possible disadvantages. Understandability has a clear relation with the size of the model [Men08, Str05]. A PDM can easily grow to over 500 data elements. It is unclear if (or even unlikely that) a visual comparison of process models can work for larger PDMs. Furthermore, some limitations of the conversion algorithms, like the appearances of double activities are not shown when the visualization is done on the basis of the PDM.
Overall, it seems possible to use visualization to improve the understandability of the dynamic behavior of (generated) process models. By improving the understandability, we can improve the usability of PBWD. It can support the selection process of the best generated process model and it can also show the dynamics of a direct execution. However, such an approach might be less effective for larger PDMs. One can conclude that visualization can be very supportive, but that visualization alone might not be enough to create full understanding in all situations.

3.3 PBWD support in case handling systems
Since the goal of this project is to create acceptance of PBWD outside the academic world, there is a clear need for BPMs that supports PBWD. In [Van06] case handling systems are proposed as BPMs that offer potentially good support for PBWD. To gather knowledge about how well case handling systems support PBWD, we investigated case handling systems in general and BPM|one in particular. The last received extra attention since this BPMS is often referred in literature and it is also a product of our research partner.

Case handling systems offer more flexibility compared to a normal, event-driven, BPM. The whole case is visible from the start to the end in a case handling system. In a normal BPM the work is pushed to the user without the context, so only a part of the case is visible. Case handling systems are able to show the whole case at all times.

The literature shows that case handling systems already support PBWD in a certain degree. Especially BPM|one supports most concepts of PBWD already [Van06]. This is the result of the data driven structure. However, case handling systems are never developed to support PBWD, so PBWD is not supported in any graphical way. A PDM editor and viewer are thus missing. We can register the required data elements, group them on forms, structure the process flow, etcetera, but the case handling systems we investigated are not able to show the product structure in a graphical way.

Overall, we conclude that case handling systems, like BPM|one, already support the data driven thinking of PBWD to a great extent. Visual support is however still missing. Without visual support, case handling systems may not be user-friendly enough to be used in the field on a regular basis. Extending a case handling system with a PDM editor and viewer can be a worthwhile way to include the desired support.

3.4 Communication in a BPM project
Driven by the research questions, we also focused on the communication aspect of BPM projects. Communication between the end-users and the process designer is required, since certain decisions will depend on the experience of the end-users. The end-users understand the product and carry knowledge about the current process. For this reason the end-users should know why some choices are preferable and why others are undesired.

Furthermore, involving the end-users in a redesign-project will create user involvement (commitment). In [Sha01, Jes03] the importance of user involvement is described. Quite some BPM projects fail since end-user support is missing. Most redesign-projects result in the implementation of a BPMS (or at least in another, more optimal way, of working). A BPMS may be technically superior, but if the end-users and (lower) management refuse to work with the system, the project is likely to fail anyway. Reasons for not supporting the system are that they feel controlled or threatened by the system. Automation often results in a more optimal process execution, which may mean that fewer people are required to fulfill the same amount of work. This may frighten the users, since they see automation as a potential threat for their jobs. Of course,
user involvement will not remove this fear completely, but if the users are involved in the project, and their concerns and wishes are taken into account, most doubts can be refuted.

Communication with the end-users is therefore of great importance. Different forms of communication are possible. We decided to focus on a direct communication, since we believe that e-mails, newsletters, etcetera are not suitable in this case. Therefore, we investigated the advantages and disadvantages of workshops and interviews. These are the most common ways of direct communication.

Workshops are an effective means to create user involvement [Sha01, Jes06]. Important decisions should be taken during the workshops, to create the desired involvement and commitment. Workshops also offer a good platform for discussions [Jes06]. Different departments will have different wishes, which may be in contradiction with each other. In a workshop, these issues can be discussed, to come to a solution that is acceptable for all involved parties. However, planning a workshop can be difficult, especially if the workshop participants are not located at the same geographical location. An electronic meeting can be used in this case, according to [Sha01].

Also, interviews can be used to create user involvement. An interview is however focused on communication with a single person, instead of with a group. Interviews are often used in qualitative research [Ind07]. Compared to a workshop participant, an interviewee can feel freer to speak, since he is not limited by the opinion of other people involved. Furthermore, since the interviewee has to answer directly, the reaction is often more spontaneous than when an indirect form of communication is used [Opt06]. However, an interview does not offer a good platform to discuss issues that involve different departments. This is a major disadvantage compared to workshops. Planning an interview is however easier than planning a workshop. Finding an appropriate time and location will be less complicated, since fewer people are involved.

Overall, communication is an important factor in a BPM project. Indirect communication seems unsuitable for BPM projects. The most direct form of communication is an interview, in which an individual interviewee will answer the questions directly. In workshops more participants are involved, which opens the door for discussion. An interview does not offer the possibility to discuss issues that involve different departments. An interview is however easy to plan, which may be more difficult when workshops are used.

3.5 Summary
In this chapter the background knowledge is described that is required to answer the research questions formulated in section 1.2. It explains why the conversion from a PDM into a process design is currently a weak point of PBWD. The offered support is not enough to realize the acceptance of PBWD outside the academic world.

Visualizing the dynamic behavior of a generated process model can be used to create additional support, so this is an interesting direction for further research. Continuing with the research in the direction of direct execution can be another option to make the PBWD approach more user-friendly.

Furthermore, we learned that case handling systems offer potentially good support for PBWD. Compared to a normal process driven BPMS, a case handling system is more data driven. This data driven structure fits quite well with the data driven structure of PBWD, however visual support for PBWD is still missing in the case handling systems we examined. Without visual support, case handling systems are not user-friendly enough to support PBWD efficiently.
Communication is the last important factor we examined. Communication is required to discuss issues and to create user involvement. Workshops and interviews both offer an effective mean to create user involvement. Workshops also offer a good platform to discuss design issues that appear during the project. A workshop can however be difficult to plan. Interviews are easier to plan, but an interview does not offer a good platform to discuss issues. Especially if the issue relates to different departments, this may be an important limitation of interviews.

The background described in this chapter, will be used in the next chapter to serve as a foundation for the development of the method. We start with three possible approaches that can be used to accomplish our goal. The background described in this chapter can be used to select the best available approach.
4 Design

This chapter describes the development of the method, guided by the goal and corresponding research questions, which can be found in section 1.2.2. In section 4.1, three possible approaches that can be used to accomplish the goal are given. The way in which the goal will be achieved differs per approach. This section also describes the selection of one of these approaches. In section 4.2 the major choices made during the further development of the method are described. The method description itself can be found in Appendix A. Section 4.3 explains the resulting method based on an example.

4.1 The approaches

The goal of this project, a user-friendly method that guides the conversion of a PDM into an accepted process design, can be achieved in many different ways. According to the research questions, the focus should be on tool-support and communication between the process designer and the end-users. In order to achieve this goal, we formulated three possible approaches that can be used for this task. These approaches are a high level description of the method, but with enough detail to understand the focus of the approach. All approaches have the same goal (a user-friendly method), but since the way in which the goal will be achieved differs, all approaches have their own advantages and disadvantages. The approaches are based on our findings in the literature, which can be found in the previous chapter.

Section 4.1.1, 4.1.2 and 4.1.3 each describe an approach. In section 4.1.4 the selection of the best approach, based on the research questions, is described. The selected approach will be described in detail in the rest of this chapter.

4.1.1 The first approach: Visual support for selecting the best process design

The first approach aims at a method in which the dynamic behavior of a process model is visualized in the corresponding PDM. As described in section 3.1, seven conversion algorithms are developed that convert a PDM into a process model. None of these conversion algorithms always results in a process model with the desired dynamic behavior. Therefore, the process designer and end-users should select the process model that matches best with the desired dynamic behavior. A tool that visualizes the dynamic behavior of the process models can support the communication between the process designer and end-users.

This support is desired, since the selection of the process models that suits ‘best’ is not a trivial task. There are several reasons why none of the conversion algorithms always results in an acceptable process model. A disadvantage of the conversion algorithms is that they perform their task without clustering. Since some form of clustering is required, the process model has to be updated afterwards. The better the dynamic behavior matches with the desired behavior, the easier this task will be (at least this can be assumed). The tool CoCoFlow can be used to test if a developed cluster is a qualitative cluster [Rei04]. Unfortunately, the tool does not work in direct cooperation with the conversion algorithms.

Furthermore, the conversion algorithms limit the processing freedom, since they do not (necessarily) allow everything that is possible according to the PDM. It is doubtful whether these limitations are desired. Limiting the freedom offered by the PDM might be desirable, but only when the process designer and end-users believe that this is desired (it must be supportive). It is not sure if the limited freedom of the resulting
process models is desired in all cases. Nevertheless, since there are in total seven conversion algorithms, it is not unlikely that at least one of the algorithms limits the execution freedom in a (more or less) desirable way.

Despite these restrictions, a method based on the conversion algorithms can have quite some advantages. The main advantage is that we can use earlier work to a great extent. When the conversion algorithms are used it is not needed to develop yet another conversion (algorithm). Offering better support in the selection process seems a good way to accomplish our goal.

It is currently a difficult task to select the best generated process model, since it is not always easy to understand the limitations of a generated process model. Nevertheless, the process designer and end-users should figure out if a process model is more or less suitable. To make this possible they should understand the dynamic behavior of the model (in relation with the alternatives).

For this reason, we will develop a tool that enables the users to explore the dynamics offered by the various algorithms for this approach. The user interface should look like the one shown in figure 4.1. This interface allows to select one of the conversion algorithms (or direct execution of the PDM, as reference). Next, the process designer and end-users can walk through different possible executions to get familiar with the dynamics of a process model. Since all algorithms and direct execution are supported, this will support the end-users and process designer in the selection of the best available alternative.

![Figure 4.1: User interface of the first approach](image)

The dynamic behavior will be shown by using color, since the literature taught us that humans find it relatively easy to separate colors from each other [Gre09]. In figure 4.1 dark green means that the data element is already known, light green that the data element can be calculated and red that the algorithm does not allow the calculation, although this is supported based on the available data. This should result in a user-friendly tool that supports the communication. However, this method might become less user-friendly for larger PDMs. The literature [Men07] describes that the larger a model becomes, the more difficult it becomes to understand and thus compare the model. This is however a general problem, that is likely to occur in every possible method. Therefore we believe that the tool supports the processing of various cases, which enables the selection of the ‘best’ alternative. The tool can be used in a workshop setting, or with individual interviews with the end-users.
Selecting one of the process models does however not result in a final process design. The process model should be updated to become a complete and accepted process design. Activities should be clustered into logical units of work, undesired limitations in the execution freedom should be restored and missing information, like roles, should be added. The method should contain guidelines how this can be done, but this will always require a certain level of modeling experience.

The development of the tool can be an extension of the ProM framework [Pro10]. Although we are unfamiliar with the architecture of the ProM framework, we believe that this can be done in a reasonable amount of time. A stand-alone tool, or an extension to e.g. BPM|one, is likely to require more implementation time, since we also have to implement the conversion algorithms in this case.

Overall, this approach aims at improving the usability of PBWD by means of visualization. Visual support is used to guide the process designer and the end-users in the selection process of a generated process model. A clear user-interface, based on a scientific foundation, should enable a structured communication. After the selection, the process model has to be updated to become an accepted process design. Extending the ProM framework may be the best way to realize the tool-support.

4.1.2 The second approach: Using workshops to obtain the best process design

The second approach aims at a method in which direct execution of a PDM serves as a starting point. Vanderfeesten describes direct execution as more flexible and dynamic than the conversion algorithms [Van09b]. However, this flexibility should be controlled to process the cases in a (near) optimal way. Vanderfeesten describes various recommendation engines, but none of them offers the desired support in any given situation. A recommendation is either based on a local decision or on a global decision. However, in the last case it might be a time consuming task to calculate the next best step.

Another way to control the execution of cases is to limit the flexibility offered by the PDM, before the process is implemented. This can be done in a workshops setting, in which it is discussed what the process design should support, and what should not be supported. In this way it is possible to limit the flexibility already before the PDM is used in an operational setting. A limited flexibility will result in better support. A recommendation system can only limit the execution freedom in an operational setting.

The second approach aims therefore at a method which limits the flexibility offered by a PDM in a workshop setting. In the workshops, the process designers and end-users should decide which executions are desired and which are undesired. By means of a tool that supports the direct execution of a PDM, one can efficiently communicate about desired and undesired executions.

Because the tool should support everything that is possible according to the PDM, the tool should calculate a data element automatically if this is supported by the PDM. For example, if data element A can be calculated automatically after B and C become available, the tool should do this. However, currently a PDM does not store all required information needed for this task. For example, the data-type of a data element is not stored in a PDM (in the formal XML specification). To fill this gap, it may be necessary to make some extensions to the formal PDM structure.

We expect that a tool that supports the direct execution of a PDM, can contribute to the understandability of the dynamics, since it can be used to examine the advantages and disadvantages of certain decisions. This idea is already partly implemented as a prototype in DECLARE [Van08b], but the focus is in this case on a recommendation system. The method proposed in this section replaces the recommendation system by
workshops in which the execution freedom is discussed. The tool should therefore focus on the data elements and its dependencies. This should explain in a user-friendly way what the effect of a certain decision will be.

We propose to use BPM|one as a target system for the tool-support, since this is a product of Pallas Athena, but mainly because it is a data-driven case handling system. Section 3.3 describes that case handling systems offer potentially good support for PBWD. The user interface we propose is a form that is separated in three columns with data elements. The first one contains the leaf elements, the second one contains the automatically derived data elements and the last one contains the manually derived data elements as can be seen in figure 4.2. Left of the form all data elements of the PDM are shown, together with their status. The data elements that are on the vertical line are ready to be entered. The ones in front of this line cannot be entered yet, since the pre-conditions are not met. The data elements after the status line are already known.

![Figure 4.2: User interface of the second approach](image)

Since the user interface displays the progress of a case in an easily understandable way, it will be suitable to support the communication between the process designer and the end-users. By processing various cases it should become clear which executions are desired. Since the tool enables the users to enter values in edit-boxes, this tool should feel more like the actual processing of a case, compared to the first approach.

In the case of a large PDM, one can decide to split the PDM into smaller parts. If the dependencies between these parts are limited, this may have a minor effect on the quality of the decisions. At least it will be easier to split a PDM, than to split a process model, most of the times. This made us believe that this method is better able to handle larger PDMs compared to the first approach.

As described, this method will use workshops to discuss where and when flexibility in the process design is desired. Workshops are described in the literature [Sha01, Jes06] as a good way to create user involvement. Also the consultants of Pallas Athena use workshops in almost all their projects to communicate with the end-users. This communication is required, since the end-users exactly know how cases are processed at the moment (and why). The tool-support offered by this method enables the process designer and end-users to test possible executions of a PDM in an efficient way. When this is done in a workshop setting, this can result in a process design where undesired (or unnecessary) flexibility is excluded. To support the communication even better, it may be possible to combine the tooling of this approach with the visualization proposed in the first approach.
Overall, this method aims at an efficient way in which the process designer and the end-users can discuss which freedom should be supported by the process design. A PDM does not offer guidance about how the end-product can be produced in an efficient way. In a workshop setting, in which the process designer and end-users discuss when and where to limit the flexibility offered by the PDM, a process design will be developed that allows for the right level of freedom. A tool that supports direct execution is used to explain the effect of certain decisions.

4.1.3 The third approach: Using interviews to obtain the best process design
The third approach aims at a method in which interviews are used to discover the preferred way of working. A PDM offers in general more flexibility than desired, where the process models generated by the conversion algorithms may offer less flexibility than desired. In the second approach we constrain the flexibility offered by a PDM in workshops, in which we ask the end-users where and how they want to limit the flexibility. In this third method we will even go a bit further, since we interview individual end-users about how they prefer to execute a case, without any restriction. We use a tool to log how they prefer to process a case.

The third approach aims therefore at a method in which the flexibility of the process design is based on the preferences of the end-users. In interviews with various end-users, the process designer will ask how they prefer to process a couple of different cases. The number of cases may depend on the complexity of the process, but we use 5 as the lower bound to accomplish enough variability, without requiring too much time of the interviewees. The sequence in which they process these cases will be logged by a tool. Combining the executions of the various cases, from the different end-users will result in an initial process model. Discovering the initial model will be supported by a process mining tool.

For this method we will develop a tool that shows all data elements on a single form. If the PDM is large, it can be split in smaller parts, or different forms may be used. In contradiction with the second approach, we do not use the structure of the PDM. Only the data elements are taken into account. This prevents that the definition of a PDM, as composed by Vanderfeesten, has to be changed. The tool will be easy to realize, since we do not take the dependencies into account. We will again use BPM|one as target system, since it enables us to use the mining functionality offered by this BPMS. Another advantage is that it is a case handling system, which offers potentially good support for PBWD as described earlier. The user interface will look like the one in figure 4.3

![Figure 4.3: User interface of the third approach](image)

This method expects that the process designer interviews a representative group of end-users from different departments about how they prefer to process the cases. It is assumed that the end-users understand the dependencies that exist (as can be found in the PDM), since they are already involved in the process on a daily basis.
Interviewing a representative group of end-users can however be a difficult task. If the group of interviewees is either too small, or not representative for the organization, the developed process design will not match the desired dynamic behavior. Furthermore, the end-users that are interviewed should understand the whole process, not just the parts they are involved in. It is doubtful if this is the case in an average organization. When the interviewed end-users do not form a representative group, or if there are no users that oversee the whole process, one cannot expect that this approach will result in an accepted process design after the mining step.

We do not expect that these kind of limitations will exist for the second approach, since this approach uses workshops instead of interviews, it is possible to discuss about the desired flexibility. If certain flexibility is desired by one department, but not by the others, the workshops can be used to find a solution. Interviews with individual users do not offer a platform for discussion, so this can result in flexibility that is not desired by all end-users. It is in workshop also not required that the participants oversee the whole process, as long as all participants together oversee the whole process.

However, since this method uses process mining to discover the process model, the modeling task will become relatively easy. If one succeeds in interviewing a representative group of users, the generation of a process model will require only a couple of minutes. Extending the process model with missing information, like the role required to execute an activity, and clustering of data elements might be sufficient for acceptance of the process design. If the cases are highly variable, the process design will become highly flexible. This may reduce understandability of the process design, but the desired flexibility remains. The mining step of this method can also serve as an extension of the second approach, but the workshops will deliver fewer cases, so the level of detail might differ.

Overall, this method can result in a process design in a relatively easy way, but only if the process designer is able to interview a representative group of end-users that oversee the whole process. The desired tool will be easy to realize, since it only requires a form with all data elements. Also the interviews themselves do not require special knowledge, but the process designer should have a large understanding of the organization to select a representative group of end-users for the interviews.

### 4.1.4 Selected approach

All three approaches aim at accomplishing the same goal, but they reach the goal in a different way. Some parts of the different methods can strengthen another, but we based our decision on the best individual approach. The decision is mainly based on the research questions formulated in section 1.2.2, but also other factors that have a direct relation with the success of this project were taken into account. To support the selection process, table 4.1 presents the advantages and disadvantages of every approach.

<table>
<thead>
<tr>
<th></th>
<th>Approach 1</th>
<th>Approach 2</th>
<th>Approach 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1: Tooling supports method</td>
<td>++</td>
<td>++</td>
<td>+/-</td>
</tr>
<tr>
<td>RQ2: Support for communication</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Tool is realistic to realize</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Method is applicable in practice</td>
<td>+</td>
<td>++</td>
<td>+/-</td>
</tr>
<tr>
<td>Works with larger PDMs</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Overall rating</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 4.1: comparison of different approaches
Based on table 4.1, we decided to continue with the second approach. The communication of this method scores better than the other two approaches, since workshops offer a good platform for discussion. The tooling of this approach supports this discussion in an efficient way. The same is true for the first approach, but in this case we are limited by the flexibility offered by the conversion algorithms. This makes the first approach a little bit less applicable (and attractive) in practice, compared to the second approach. The third approach is the most easy to realize, but this approach does not outperform in quality. Since the second approach also works relatively well with larger PDMs, this approach has the most advantages.

Overall, the second approach offers the best opportunities to improve the usability of PBWD. In the next section a detailed description of the design of the method is given. It explains how the method should make PBWD better usable and why certain choices are made.

4.2 Detailed design of the method

This section can be seen as the most important section of this thesis, since it presents a detailed description of the design decisions that are made during the development of the method. The previous section already presents the decision to develop a method that is based on direct execution and a workshops setting.

In section 4.2.1 the main steps of the method are described. A more detailed description of the method can be found in Appendix A. Section 4.2.2 describes how the tool should support the method. In section 4.2.3, the communication platform is presented. Since the workshops perform a central role in the method, this section explains how the participants should be involved in the method. Finally, section 4.2.4 describes how a project can be maintained. This is however an optional step of the method.

4.2.1 Steps of the method

This section will explain the main steps of the method. It cannot be seen as a replacement of the method description (appendix A), since many of the details are left out.

However, first we will position the method in the larger setting of a PBWD project. The method focuses on the conversion of a PDM into a process design, since this is currently the weak point of PBWD. As can be seen in figure 4.4, this step is called *identify process design*. Though the focus is on the conversion of a PDM into a process design, we believe that PBWD can only become widely accepted if the resulting process design can be maintained, even if the product changes slightly. For this reason we extended the method with an additional step *maintain process design*. The implementation phase in between is not taken into account, since it has no direct relation with the development of a process design.

![Figure 4.4: Positioning of the method](image)

With the position of the method, within a PBWD project in mind, the major steps of the method can be explained. These steps can be found in figure 4.5. The method can start after the PDM is constructed. From this moment on, the first workshop should be prepared. In this step, *prepare first workshop*, the PDM is converted into an executable model (BPM|one model). This model makes the PDM available in BPM|one. This BPM|one model will be used during the workshops that follow.
In the step, *first workshop*, the process designer should discuss with the end-users what the process design should look like. The BPM|one model, that represents the PDM, is not suited as a process design, since it does not offer the desired support. By processing various cases with the BPM|one model, one can discuss when and where extra guiding is desired. Extra guiding means in general, limiting the dynamic possibilities of the model. Figure 3.1 explains why this can be desired. At the end of the workshop the process designer can automatically generate a process model, based on the desired executions. This automatically generated model will be discovered by an adapted version of process mining.

The mined process model may only be treated as an initial version, since not all proposed changes will be captured by this model. Changes proposed during the workshop have to be logged on change request forms. These change request forms basically describe how and where to guide the users through the process, by limiting the dynamic behavior. In the step *model process design* these change requests should be processed.

After the process designer believes that the process design complies to all change request, he can plan a new workshop. In the step *approval workshop* he should ask the workshop participants to approve the process design. This can either result in an approved process design or in new change requests. In the first case we are done. In the second case, the steps repeat themselves from *model process design*. This means that the method will end from the moment the process design is approved.

However, as described before, we included an additional step *maintain process design*. In this step small changes in the product, and thus the PDM, are applied. With basic knowledge of the method it should be possible to understand the rest of this chapter. The rest of this chapter describes the most important decisions that are made during the development of the method. It starts with a section that explains the goal and composition of the tool-support.

### 4.2.2 Tool-support

In section 1.2.2 we described that tool-support can be the key to success. Without tool-support it will not be possible to develop a user-friendly method. This section explains how tooling supports the method. Since the method is as strong as the weakest link, we embedded tooling in all main steps of the method. A more detailed description of decisions made during the development of the method can be found in Appendix C.

**Converting the PDM into BPM|one model**

Since direct execution is the basis of this method, we required a tool that makes the PDM available in an executable form. This idea is already implemented in DECLARE, but this declarative workflow system is in an early stage of development. To our best knowledge, this workflow system is only used within the academic world. Therefore we do not believe that this experimental workflow system is a good platform to promote PBWD. Based on the work of Vanderfeesten [Van08b], we selected BPM|one as target system.

Therefore, we developed a tool (a prototype) that can translate an arbitrary PDM into an executable BPM|one model. This translation is largely automated. However, since not all required information is stored
in the PDM, the resulting BPM|one model is not yet complete. To automate the conversion as much as possible, we extended the PDM definition with the data-type of the elements and with the operation type (leaf, automatically derived, manually derived). We decided however not to include the formulae that are required to calculate automatically derived data elements. This decision was taken, since a minor mistake in the formulae (e.g. type error) will result in a model that is not accepted by BPM|one.

This means that the formulae have to be entered manually, before the model becomes executable. Although this step is relatively easy, the conversion will become more user-friendly if the formulae are also automatically included in the BPM|one model. This will however require an addition to the PDM, what basically means that the work is moved to the step where the PDM is identified. This seems only desirable in combination with a PDM editor that is able to check if the required formulae are correct.

The conversion should result in a form with three columns as can be seen in figure 4.2. One column is for the leaf elements, one for automatically derived data elements and one for manually derived data elements. All data elements of the PDM are placed in the corresponding column. The underlying BPM|one model offers the same dynamics as the corresponding PDM. This should improve the communication, since the tooling can be used to show the effect of a process execution efficiently.

**Using the BPM|one model during the workshops**

After the BPM|one model is completed and tested it can be used. The goal of the model is to support the workshops. During the workshops the participants have to decide what the process design should look like. What level of flexibility is desired, and what the effect is of this flexibility, are issues that should be discussed during the workshops.

It is however not easy to understand the effect of a decision on the product structure alone. The tooling belonging to this method makes it possible to ‘execute’ the PDM. This makes it possible to ‘visualize’ the effect of a decision. If a data element is entered, the model automatically ‘calculates’ the new state. In this way the participants see which new data elements become available in the PDM. This should support the discussion, since the tooling can be used to discover which information is missing, which information is available but not required, if the data elements are entered in an efficient way, and so on. Although PBWD always supports the discussion based on the product structure, without tool-support, understanding all these dependencies can be a difficult task. By processing (historic) cases, the tooling can be used to test and communicate the effect of a decision.

To support the workshops in the best possible way, we choose to aim at a simple user interface. The developed interface shows the three columns of a BPM|one form as can be seen in figure 4.2. By separating the leaf elements, automatically derived elements and manually derived elements, we hope to achieve the best possible understanding. In this way the ‘type of work’ to process a case is more or less separated. The status bar (called Wavefront in BPM|one) left of the form shows the progress of the case. Since we used existing parts of BPM|one when possible, current users of BPM|one will have no problem to understand the tooling. According to Green et al. users tend toward the habitual [Gre09]. Since the tooling should feel as habitual as possible for current BPM|one users, we hope to create the best possible understanding.

By processing cases with the BPM|one model, we can log data related to the cases used during the workshops. This can help the process designer during the modeling phase, but it is especially useful for the automatic generation of a process model.
Mining a process model

BPMone offers the option to log information related to the case executions. This information can be used for e.g. process mining. Process mining is meant to discover how cases are processed, instead of how they should have been processed [Aal06]. In this method we use process mining to discover the paths that resulted in a desired case execution. Since we are only interested in desired case executions, we have to delete the undesired case executions to prevent them from influencing the mined model.

The mining step was originally not planned to be a part of this method, but it was proposed as a part of the third approach described in section 4.1.3. We decided to include the mining step during the development of the method, since an intermediate result can keep the participants involved. In [Din04a] it is described that it is important to communicate the draft process for communication purposes and for further development. Paper [Din04b] even states “It was therefore important to give the workshop participants feedback through a running system even if it was not complete. Waiting for the perfect and complete process guide would take too long and could kill the enthusiasm”.

The mined model can be used to communicate the intermediate result, but it can definitely not be seen as a suitable process model. The process mining algorithm does not take knock-outs and clustering into account. Knock-outs can be used to reach the end-product by e.g. a rejection [Rei03]. This can prevent the need to process the rest of the case. If there are different knock-outs possible, it is preferable to select the one with the highest chance of a knock-out. The mining algorithm does however not take this into account. Also clustering, as described in section 3.1, is not yet taken into account by the mining algorithm.

The mined model may even allow executions that are not supported by any of the processed cases. The reason is that most mining algorithms strive for a compact process model (with a minimal number of tasks). This makes a model often better readable, but it can also result in paths that are not intended to be executed [Aal07]. In figure 4.6 this limitation of process mining is explained.

Figure 4.6: Possible results of a mining algorithm.

Figure 4.6 presents two simple process models that both allow the execution A – B – C – D and A – C – B – D (all splits and joins are of the type XOR). Model 4.6.A has in total 6 tasks, with B and C duplicated, and only supports these 2 paths. Model 4.6.B has only 4 tasks, without duplicates, but this model also supports the execution of additional paths. For example the paths A – B – D and A – C – B – C – D are possible according to model 4.6.B. Therefore, it can be concluded that process mining can support the method (especially for communication purposes), but that the mined model should be inspected carefully, before it can become a complete process design.

Modeling the process design

The mined model can be used to create user involvement by showing the intermediate result. However, the
model will also be used during the modeling phase. Especially if the desired modeling language is BPM|one, the mined model can be updated relatively easily. We do not want to restrict the method to one modeling language, but we had to select a modeling language for the tool-support. If another language than BPM|one is used, the model can be converted into the language of choice.

During the modeling phase, the process designer should combine the mined model with the changes proposed during the workshops. Proposed changes have to be logged on a change request form, as will be explained in section 4.2.3. A part of the change requests is possibly already captured by the mined process model, but as explained clustering and knock-outs are definitely not taken into account. During the modeling step, the mined model should be adapted and information relating to roles and activities has to be added. How this has to be done will depend on the modeling language. In case of BPM|one we refer to [Pal09].

Since the modeling step will depend on the chosen method, we do not go into detail on this point. If BPM|one is chosen as modeling language, there is no effort required to convert the model. The less effort required, the better the chance that PBWD becomes widely accepted. Therefore we also investigated if the clustering of data elements and operations can be automated. To the author’s best knowledge, there is no automatic clustering algorithm. The quality of clusters can be tested by applying the coupling-cohesion metric as described in [Rei04, Van08a]. This metric does however not support the composition of clusters. Developing our own clustering algorithm did unfortunately not fit within the time of this project. For now, we suggest to use common practice, e.g. by clustering data elements and operations that sequentially follow each other and that are executed by the same role. Furthermore, it should be possible to cluster leaf elements and leaf operations. Further research in this direction is however desirable.

4.2.3 Communication platform of the method
The previous section of this thesis describes how tooling should support the method. Some parts of the tooling are especially meant to support the communication between the process designer and the end-users. This section describes how qualitative communication should be reached in all steps of the method. Since we chose to select a workshop approach, this starts with the preparation of the first workshop and ends after the process design is approved.

The workshop approach was chosen since workshops offer a good platform for discussion. The workshops have to be used to discuss different insights and opinions, which should finally result in an agreement about how the process design can support all end-users (and thus departments). The main alternative was to use an interview structure, but interviews do not offer the possibility to discuss issues between end-users. Therefore decisions could be made that are disliked by some of the participants, which could finally result in a failed project. Involving all parties in a discussion can seriously limit this risk, so a workshop approach fits our goal the best [Sha01].

Preparing a workshop
Organizing an efficient workshop is however not always easy. A good and structured preparation helps to achieve an effective workshop. This starts with planning the workshop on time. According to Jeston et al. [Jes06], “Planning workshops should be on time, since it requires some of the best resources out of the business for a significant amount of time to attend the workshops. It requires the best people, not just some people available”. The right moment to plan a workshop seems to be just after the conversion of the PDM into the BPM|one model. The duration of the workshop should depend on the project and on the number of participants. The process designer should decide what a suitable duration of the workshop will be.
According to Sharp et al. a workshop should be ‘developed’ [Sha01]. The process designer should therefore prepare the workshop. He should think in advance what the process design should look like. To be able to think about what the process design could look like, the process designer should use (historic) cases. These cases can be used to examine what the process should support. We believe that the process designer should collect these cases from various departments, since this brings him (again) in contact with the participants. This brings the project under attention of the participants, what may help to involve the participants. The cases he should collect, have to be realistic (representative for the process under consideration), and as diverse as possible. In this way we will assure the cases represent the process under consideration. A combination of diverse historic cases is therefore seen as suitable.

The workshops
The workshops should have a clear structure, in order to achieve the best results. People tend toward the habitual according to Green et al. [Gre09]. Therefore we believe that a clear structure will help to achieve the goal of the workshop. The workshops should start with an introductory presentation, followed by the core of the workshop, and at the end the process designer should give a summary of the decisions made during the workshop.

The process designer should explain the goal of the workshop, in an introductory presentation. During the first workshop the process designer should also explain the goal of the whole project. In [Jes06, Din04a] an introductory presentation is described as a good way to start a workshop. An introduction presentation is also common practice within Pallas Athena. The process designer should make sure that all participants understand the goal of the workshop. If the goal is not completely clear, or if the participants do not have the same expectations, there is a likely chance that the workshop will not lead to the desired result. According to Jeston et al. “If you not set the expectations for the participants they will set their own” [Jes06]. It is therefore important to make sure that all participants have the same expectation of the project.

The rest of the workshop should be devoted to the core of the workshop. This means that the participants should decide what the process design should support and what not. The tooling described in the previous section has to be used to support the discussions. These discussions are required, since different departments (represented by the participants), will have different wishes. Certain decisions may be preferable for the one department, but work counterproductive for the others. The PDM offers already some support, since it visualizes the product structure. However, as section 4.2.2 explains, it is not easy to understand the effect of a decision on the product structure alone. Therefore we use tool-support during the workshops, to visualize the effect of a decision.

To make decisions traceable they should be logged onto a change requests form. This change request form should hold e.g. a change request ID, a description of the desired change and the date of the change request. We chose this approach to make sure that decisions and changes are traceable. Also in Jeston et al. the importance of logging changes is described [Jes06]. Traceability is not the only advantage of logging the change requests, since they can also be used to implement these change requests and to test if the resulted model complies with all requests. This ensures that decisions that are made during the workshops are indeed processed. Since we require that a change request is accepted by all participants, it will reduce the chance that change requests are in contradiction with each other.

Change request are however mainly concerned with undesired flexibility (dynamic behavior). Logging all desired executions may be desirable, but this is difficult to realize in practice since the number of desired executions may grow incredibly fast. Therefore this method does not require that all desired executions are
logged. It is however desirable to log executions that led to discussion. In this way all major decisions are logged, without requiring too much time.

To prevent that the participants feel like they did not spend their time efficiently, an intermediate result at the end of the workshop is desired. The most basic form is to give a summary of the decisions made during the workshops and how they will influence or relate to the rest of the project. However, it is hard to show the progress to the participants with only a summary. Therefore a more concrete approach is desirable. To accomplish this, we extended the method with process mining step as described in section 4.2.2.

Process mining is usually used to discover how cases are really executed, instead of just looking at how they should be executed. This gives information related to the quality of the process. We will use process mining in a comparable way, but we are only interested in the desired executions, since we want to show the participants an intermediate model. This model should allow all desired executions that are discussed during the workshop. By showing them an intermediate process model, participants should feel like their opinion is really taken into account. The main goal of the process mining step is thus to communicate the progress to the participants, although process mining has also other advantages as described in 4.2.2.

The communication should not stop after the first workshop. The research goal states that the method should lead to an accepted process design. The participants should therefore approve the process design developed during the modeling phase. They should judge if the model developed by the process designer meets all change requests. It can also happen that some points are not discussed earlier and that the process design therefore still contains undesired behavior (flexibility). For this reason the process design should be accepted by the participants in another workshop. In this workshop the process design will either be accepted or rejected. In case the participants reject the process design, this should result in new change requests, which means that the last steps should be repeated. This method ends after process design is accepted in a workshop. Asking for approval by e.g. e-mail may be cheaper and easier to realize (no workshop has to be planned), but this will not offer the possibility to discuss the process design, and can therefore not replace a workshop.

4.2.4 Maintainability
After the process design is accepted it can be implemented within the organization. Implementation is outside the scope of this method, since it will rarely relate to the acceptance of PBWD. Maintainability of the process design has however stronger relations with the acceptance of PBWD, so we included an optional step in the method that relates to the maintainability of a PBWD project. Maintainability is important since it is likely that the product, and thus the PDM will change over time. Changes in the product can be the result of innovation, legal changes and so on. If the PDM changes, the corresponding process design should also change [Vog09].

It is possible to start from scratch if the PDM changes, but the method aims at updating the current process design. This option was chosen since starting from scratch almost certainly results in higher costs compared to adapting the current process design. This will be the cases since choices that are made in the past, have to be made again, while there is no real reason for this.

Adapting the current process design should be done by focusing on the dependencies between the new PDM and the current process design. In the current process design the new operations and data elements are missing (or deleted operations and data elements are still included). If one can locate to which regions the new (or deleted) data elements and operations belong, it is possible to adapt these regions. A region
consists of one or more transitions that are logically related. Unaffected regions remain therefore untouched. This approach is based on the work of Reijers et al. that describes how small changes in the PDM should be handled [Rei09]. The method should fully support this approach, since this method requires that change requests of earlier workshops are logged. Therefore it should be possible to focus on the dependencies between the PDM and the process design.

Since the structure of an ‘evolutionary PBWD project’ (redesign of an existing process model) is comparable with a ‘revolutionary PBWD project’ (redesign with a Greenfield approach) the workshop settings and responsibilities remain the same. Also the tool-support can be used to a large extent. Process mining may however not be effective, but this is depending on the changes that are made. If possible, the same participants should be invited for the workshop as for the original project. It is however not likely that this is possible in all cases, so the group of participants should be comparable. The next section will give an example that explains what a project should look like.

4.3 Example
The previous section explains the main decisions that are made during the development of the method. This section gives an example of a project to explain the method. The example describes the project in such a way that it should be possible to replay this project. This example can also take away possible ambiguities in the method description. The project is conducted at Pallas Athena internally, so it will not be implemented in reality.

This example will use a simplified version of a mortgage calculation. The PDM contains ten data elements and twelve operations. This PDM is originally introduced by Vanderfeesten in [Van09b], but we added the data elements I and J and operation Op10 and Op12 to include a manual operation. In figure 4.7 the PDM is shown together with a description of the data elements.

Since this PDM is a simplified version of reality, the PDM cannot be used for a real mortgage application. However, it is perfectly suitable to explain the steps of the method. The goal of this PDM is to calculate the maximal mortgage that a potential customer can borrow from a bank. The root element can be reached in 3 ways: with Op01, Op02 or Op03. Op01 requires B, C and D as input. Op02 requires E as input and Op03 requires H as input. Since E represents a previous offer, it is not clear if this data exist. Op03 can only be taken if the applicant has a credit registration. Furthermore, to calculate data element C, data element F, G, J and H should be available and to calculate J data element I is required. The data elements that are not yet mentioned are leaf elements. Op10 is a manual operation. Op01, Op02, Op03 and Op05 are automatic operations and the rest of the operations are leaf operations.
In order to replay the project, it is required to know how the operations calculate the data elements. Op10 is the only manual operation. This operation is used to judge the employer of the applicant. If the employer is not trustworthy the maximal amount that can be borrowed will be lower. Since it is not possible to store all employers with a ranking, this will be a manual task. Op01 will be used to calculate the maximal amount that can be borrowed by the formulae $C \times D + C \times D \times (100 / B)$. Op02 will simply take the value of $E$ as soon as it becomes available. Op03 will result in 0 if the applicant has a registration at BKR and will not be executed otherwise. Finally, Op05 result in $G \times (100 / F)$ if $J$ is TRUE and $H$ is FALSE or in $G \times (100 / F) \times 0.5$ if $J$ is FALSE and $H$ is FALSE. In all other situations Op05 will not be executed. These formulae are required during the preparation phase, but since the PDM will depend on these formulae we introduced them here.

4.3.1 Preparation
The method starts by converting a PDM into a BPM|one model. Any XSLT engine will do, but in this example we use XMLSpy\(^3\) (see Appendix C for more details). The translation can be done by opening the PDM (stored in the XML format), press the XSL button and select the XSLT file belonging to this method as can be seen in figure 4.8. The resulted file has to be saved with the extension chp.

The resulted model has to be loaded into BPM|one afterwards. Details about how to load a model can be found in the BPM|one user manual [Pal09]. Afterwards, the formulae given in the introduction of this section have to be entered in the corresponding fields in the format given in figure 4.9 (for operation Op01).

\(^3\) http://www.altova.com/xmlspy.html
After the model is completed it should be tested. If no errors are found, the model is ready to be used in the workshops, so the first workshop can be planned. The form belonging to this model has been shown already in figure 4.2. Since the PDM is not based on a real process (at least not directly), we could not collect historic cases for this process. Therefore we have devised a number of examples as can be found in table 4.2. The table shows 9 possible cases, together with the opinion of the process designer. The last serves as a part of the preparation. Furthermore, the process designer should make an introduction presentation, to explain the goal and to introduce the PDM to the participants.

<table>
<thead>
<tr>
<th>Case</th>
<th>Desired by process designer?</th>
</tr>
</thead>
<tbody>
<tr>
<td>H (registration) – A (use to explain tooling)</td>
<td>Yes</td>
</tr>
<tr>
<td>I - F - G - H (registration) - A</td>
<td>No</td>
</tr>
<tr>
<td>E - A</td>
<td>Yes</td>
</tr>
<tr>
<td>B - D - H (no registration) - I - J - G - F - C - A</td>
<td>No</td>
</tr>
<tr>
<td>H (no registration) – B – D – F – G – I – J – C - A</td>
<td>Yes</td>
</tr>
<tr>
<td>H (no registration) - E - A</td>
<td>No</td>
</tr>
<tr>
<td>I - E - A</td>
<td>No</td>
</tr>
<tr>
<td>H (registration) - A</td>
<td>Yes</td>
</tr>
<tr>
<td>H (no registration) - I - J - B – D – F – G – C - A</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4.2: Example cases
4.3.2  The first workshop

To execute the workshop we used an internal meeting at Pallas Athena in which we explained our research. We used this example to explain and test our results. After a presentation about PBWD we showed the participants the PDM together with the tooling and the 9 example cases of figure 4.1 (without the opinion of the process designer). We asked them what kinds of freedom they would desire in the given setting. The participants were no real end-users, but they were able to put themselves in such a position.

The opinion of the process designer, as can be found in table 4.2, matched with the flexibility desired by the participants. It proved preferable to test first whether data element E is available. If this is not the case, the next best step is to collect data element H according to the participants. This order was chosen since both data elements could result in a knock-out, however data element E can be collected internally and data element H requires an external partner (BKR in this case). These two knock-outs were captured in a change request. Furthermore, the assumption was made that all data elements can be processed by the same person or role, which led to another change request.

For the 5 desired cases we mined the model of figure 4.10 (two cases follow the path H – A). The model starts with data element E or H, however no difference in priority is made. This is the result of the fact that knock-outs are not taken into account. Since the model is based on only 5 cases, some tasks appear in sequential order in the model, where they should be in parallel (and clustered) in the final process design (at least this can logically be assumed). After showing and explaining the model to the participants, the workshop ended.

![Figure 4.10: Mined model of the mortgage example](image-url)
4.3.3 **Modeling phase**

After the first workshop, the process designer should combine the mined model with the change requests. Furthermore the activities should be clustered where possible. The resulted model is shown in figure 4.11. The model starts with a check if E is available. This knock-out was required by one of the change requests. If E is unavailable one can only enter H as a result of the same change request.

![Final model of the mortgage example](image)

If there is no credit registration, data elements B, D, I, F and G have to be entered within the same activity. This is the result of the change request that describes that all data can be entered by the same role, in combination with the wish to cluster data where possible. Since C is and A are automated task, the case should finish automatically after J is entered. Since A can be derived in different ways, these tasks should not be combined. If H is positive or if E is entered the case will also end automatically.

In this example we use BPM|one as modeling language. Another language can be used, but this requires that the model of figure 4.10 is first converted to the desired modeling language. We did not choose this option, since it will result in additional overhead.

4.3.4 **Approval workshop**

The method expects that the process design will be accepted by all participants in a new workshop. Since this project was only meant to explain the method, a second workshop to ask for approval was not desired. This would be too expensive and since the participants are not really involved in the process on a daily basis, we did not expect comments on the details of the process design. Instead, we asked 2 of the participants if they would accept the process design and this proved to be the case. After this (replacement of an) approval, the required part of the method ends.

4.3.5 **Maintainability**

As addition to the method we added an optional step that can be used when the product changes slightly. In this example we present a legal change of the PDM. In the original PDM the maximal mortgage depends on
his employer. In this section we assume that a new law forbids basing the decision on the employer of the applicant. It is left to the imagination of the reader why this can be the case.

This new law means that data element I and J together with operations Op10 and Op12 should be removed from the PDM. The method requires to locate the dependencies between the new PDM and the current process design related to these data elements and operations. In a workshop setting one should discuss how the corresponding regions should be adapted. In this example two activities use either data element I or J. It is decided that the activity with data element I should be adapted, to remove this data element. The activity with data element J could be removed completely. This resulted in the model of figure 4.12. After an approval of the new process design, it can be implemented. This imaginary approval is the end of this example.

![Process model after the changes are processed](image)

Figure 4.12: Process model after the changes are processed
5 Evaluation
The province of North-Brabant offered us the opportunity to test and evaluate the method in a real project. This government agency is currently embracing PBWD for some of their projects. At the time of our research project, a PBWD project for the reservation of meeting rooms was conducted.

The project was started since the province has to reduce the number of mistakes made during the reservation of meeting rooms. The province of North-Brabant rarely has enough meeting rooms and meetings are not always scheduled in an optimal way. Communication problems between departments result in a relatively high number of mistakes. Over the first half of 2009, in total 6272 meetings were requested. From these requests 303 meetings could not be planned, 1082 meetings were canceled after they were planned and 16 meetings were not cancelled, but did not take place either. The goal is to improve this situation, but improvements were not quantified beforehand, since improving the quality (reducing the number of mistakes) of the process is the most important factor for the province.

The process for the reservation of meeting rooms covers different departments. Since every department should be involved, the project group had over ten members. The PDM was also composed in a workshop setting, which resulted in a large overlap between the participants of both workshop sessions. This could possibly streamline the integration with our method.

In the rest of this chapter we evaluate our method based on this project. This evaluation is both based on our own experiences and on the experiences of the participants. In section 5.1 the goal and structure of the evaluation project is described in more detail. In section 5.2 the PDM that serves as input for the method is described. Section 5.3 describes how we and the process designer of the province of North-Brabant experienced the method. Section 5.4 describes how the participants experienced the method. We distinguish these in this evaluation, since the evaluation with the participants requires a different level of detail. In section 5.5 an overall result is given, that combines these opinions.

5.1 Goal
We participated in this project to evaluate the method in a practical setting. Until this project, the method was only ‘tested’ in fictitious projects, like the one described in section 4.3. A fictitious project may indicate how the method can behave, but the practice may be different. By evaluating the method in a real project, we can confirm the usefulness of the method and locate its weak points.

The evaluation was divided into two parts. The first part was conducted throughout the duration of the project. In this part we evaluate the project from the point of view of the process designer. By using the method in this project, we could locate where undesired side effects occur, or notify that the tooling is not self-explanatory enough. By participating in the workshops ourselves, we could easily spot its weak points and that allows us to adapt the method already during the project. Other problems or limitations may possibly result in future work. The results of this part of the evaluation are described in section 5.3.

The second part of the evaluation took place after the process design was accepted. In this phase, the project was evaluated with the participants, by asking them to fill in an evaluation form. The questions on the form related to the workshop setting and the tool-support. The experiences of the participants are described in section 5.4.
5.2 PDM
The PDM is seen as input for the method. However, the PDM can influence the project to a great extent. Especially since only one evaluation project is conducted, the quality of the PDM may influence the quality of the evaluation of the method. The PDM used during this project can be found in figure 5.1. As can be seen, the PDM is relatively small. It has 26 data elements and 25 operations. Some data elements are shown in red, since their meaning is discussed in this section. Nine of the operations are either automatically or manually derived. The remaining 16 operations are leaf operations. Most leaf elements and data elements that have a list as predecessor are required for the reservation of a room. This results in a relatively flat PDM.

Figure 5.1: PDM for the planning of a meeting

A couple of decisions were made that may influence the outcomes of the evaluation. It is for example decided to see a meeting as the end product, instead of a reservation for a room. This makes it difficult to calculate e.g. the throughput time of a case. However, since the goal is to improve the quality, this should not have an influence on the measurability.

Furthermore, the PDM contains static data elements. With static data elements, we mean data elements that have the same value for all cases over a longer period. Examples are the data elements ‘ICT list’, ‘AV list’ and ‘Catering list’. These kinds of data elements will contain a static list like: PCs, beamers, and laptops for data element ‘ICT list’. A comparable construct is used for the data element ‘Earlier Bookings’. This data element represents all earlier made reservations. These reservations are needed to book a room, since the selected room has to be available. However, every new or changed reservation should result in a change in this data element. PBWD offers limited support to indicate dependencies between cases; therefore the decision to capture it by a data element was made. The decision to include this type of data elements can
lead to discussion, but since the method should support every possible PDM, it can be seen as a good test for our method.

It should however be noticed that the data elements described above, are system dependant data elements, since their value depends on internal information systems. For this reason we will call this kind of data elements **system data elements**. To be more specific, we formulated the following definition of a system data element:

> A system data element is a data element that can be retrieved from an information system in an automated way with zero costs.

Of course, the actual development of a connection between different information systems might be a costly affair, but we only focus on the costs to retrieve the data element. We assume that there will be no costs involved if the data element is retrieved from an organization’s own information systems, but that costs will be involved if the information system is external. More research is however needed to verify this definition.

Finally, in this PDM it is assumed that a data element also contains all data of data elements lower in the chain. This may deviate from the definition of a PDM, since the dependencies become less visible. Nevertheless, the province of North-Brabant chose this approach, since it prevents the PDM from being flooded with arrows. E.g. the data element ‘Room’ also holds data regarding the date and time of the meeting. This means that the operation that produces the confirmation has only a direct link with data element ‘Room’ and not with all data elements lower in the chain. The province believes that this improves the readability of the PDM.

### 5.3 Evaluation from a process-designer's point of view

The previous chapter describes the choices that are made to make the method as user-friendly as possible. In this evaluation section, we describe how well we succeeded in this goal. This section is based on our own opinion and on the opinion of the involved process designer from the province of North-Brabant. The experiences of the participants are described in the next section.

This section is split into several paragraphs to answer the research questions in a structured way. Section 5.3.1 describes to which degree the tooling supports the method. Section 5.3.2 describes our experiences with the communication platform. Since tooling plays such an important part in the method, major issues are related to the use of this tooling to support the communication. Section 5.3.3 describes our main findings, which is a combination of the experiences of all involved parties.

#### 5.3.1 Tool-support

This section evaluates the tool-support described in the previous chapter. This section is built up in the same way as section 4.2.2.

**Converting the PDM into a BPM|one model**

The tool-support starts with the conversion of a PDM into a BPM|one model. Converting the PDM of figure 5.1 into an executable BPM|one model did not result in any deviation from the method description. Only two formulae for automatically derived data elements were required. Data element ‘Confirmation’ is a confirmation of the room reservation, therefore the formula just copies the value of ‘Room’ directly after the data element ‘Room’ is filled. In any other case the operation does not produce any output. Data element ‘AV/ICT delivery’ is calculated after data element ‘Room’ is filled, by concatenating ‘AV equipment’ with ‘ICT equipment’. This resulted in the BPM|one form of figure 5.2.
The model was tested by comparing the flexibility of the PDM with the flexibility of the BPM|one model. The BPM|one model should offer exactly the same dynamics as the PDM. E.g. in figure 5.2 no data element is entered, so all leaf elements should be enabled and none of the other data elements should be enabled or calculated yet. As can be seen this proves to be the case. All other dependencies also matched, so the model passed the test.

![Figure 5.2: The user interface of the BPM|one form](image)

Using the BPM|one model during the workshops
The first workshop was the first time that the end-users were confronted with the tool-support offered by the method. We asked the participants already before this workshop to deliver a couple of historic cases that could be processed by the tooling. The composition of these cases is described in the next section.

Although the tooling itself worked fine, the tooling caused some problems related to the understandability. This was the result of the combination of two factors. The first factor relates to the system data elements (e.g. the list with available ICT equipment) that are included in the PDM. Since the PDM contains various system data elements (which are leaf elements), the tooling shows these data elements in the left column. Since this is as intended, it should not have caused any problem. However, some participants compared the tooling with the current application form. This caused some problems related to understandability, since the users were not confronted with these data elements before. The second factor is thus that some participants did not understand the goal of the tool-support (and the revolutionary design method) completely. If the current application form had shown these system data elements, these problems may not have occurred. However, in this case the goal of the tooling (and the PBWD approach) had to be explained again during the processing of the first case. It proved to be difficult for the participants not to think in an evolutionary way, since they constantly tried to link the tooling with the current process.

Unfortunately, even without the delay caused by the tooling, processing the first case required more time as expected. This was the result of discussions related to alternatives (4 in total) during the processing of this case. We did not want to process various cases with our tool simultaneously, so we wrote down these alternatives. We could finish this case, but since the limited time available for the workshop we could not process another case. The case that we processed, together with a description of the 4 alternatives can be found in Appendix D.
In the second workshop we processed 3 additional cases. This was to validate the assumptions we made in the modeling phase, to help us in formulating the task descriptions, and to discover the roles that are needed to process these tasks. However, from an evaluation point of view, we were mainly interested in how the participants would react this time on the tooling. Most participants should have been familiar with the tooling by now, so we were interested how this influenced the workshop. Processing the cases led to some new insights, but it mainly confirmed weak points of the method. Understanding the tooling in relation with the system data elements still proved to be difficult. Also the goal of the tooling seemed to be forgotten by some participants.

Removing the system data elements from the tooling may improve its usability. However, just removing the data elements that meet the definition given in section 5.2 will not be enough, since the classification of the data elements (leaf, manually derived, automatically derived) should be adapted accordingly. A system data element is almost certainly a leaf element. When a leaf data element is deleted, another data element (or even more than 1) should become a leaf element instead. When the classification is not adapted, some data elements will be placed in the wrong column of the tooling, which in its turn can cause troubles in understandability.

Another way to improve the understandability of the tooling seems to offer more (or better) visual support. The tooling only shows a form with the data elements. The structure of the PDM is not visually embedded in the tooling. Already during the development of the method, a PDM viewer was seen as a desired extension. However, based on the expected development time, we decided not to include this in the tooling. Instead we chose to use a printed PDM during the workshops. Although this gave some support, this was not enough to make the link between the PDM and the tooling clear for all participants in every situation. We expect that a visualization of the (state of the) PDM would drastically improve the understandability. Such visualization may look like the one we proposed in our first approach (see section 3.1).

Mining a process model
Since we were unable to process more than one case during the first workshop, mining a process model was of limited use. A mined model based on only one case is simply the sequence of the executed steps. The mining functionality was therefore not used during the workshop. As a result, we could not verify if a mined model improves the communication between the process designer and the end-users.

However, since the mined model is also used during the modeling phase, we decided to process the four alternatives discussed during the first workshop afterwards. Details about these alternatives can be found in appendix D. We processed these alternatives in the way as we expected that the participants would have processed these cases. Considering the relatively small number of data elements, the number of decisions we had to make was limited. We could however not verify if the participants accepted the order in which these cases are processed. Therefore, it is impossible to draw any conclusion based on the resulting model alone.

Mining the model itself did not cause any problem. The resulted model is shown in figure 5.3. Since most joins in the mined model are of the type AND, routing problems cannot occur at these places. Activity ‘Cost centre’ is the only XOR join. Executing this activity twice (both incoming arcs will be enabled at some point in time) will result in a deadlock situation at activity ‘Room’ [Aal01]. It is up to the process designer to prevent this from happening in the final process design. A disadvantage of BPM|one that there is no visual difference between an AND or XOR splits (and joins). This makes it difficult to get a quick overview of the process, as can be concluded based on figure 5.3. A deeper analysis during the workshop seems for this reason not
desirable. Although this is also not the goal of the mined model, the process designer should actively prevent this from happening.

**Figure 5.3: Mined model after adding 4 alternative cases**

**Modeling the process design**

Since the mined model was only indirectly based on real historical cases, we chose not to use the mined model as starting point for the development of the process design. This decision was however also based on the fact that the process model should be delivered in BPMN, instead of BPM|one. Converting the mined model to BPMN and continuing from there was not seen as the best possible decision, since every conversion step means a loss in quality. This does however definitely not mean that the mined model is useless. The mined model can be used to ‘test’ the BPMN model. By analyzing the two models, we can verify if executions that are supported by the mined model, are also supported by the BPMN model. This also made it possible to separate responsibilities. The process designer of the province of North-Brabant was mainly responsible for the BPMN model, since he is not directly involved in the research project. Our role will be to compare the resulting model with the mined model and support the process designer where needed.

During the modeling phase, we tried to match input received during the workshops with constructs in the process model under construction. Examples of such input are comments, questions and remarks. In this
manner we could develop a process design that supports all requirements we were aware of. Since the input of the first workshop was limited, we decided not to describe the tasks and roles already after the first workshop. The resulted model can be found in figure 5.4.

![Figure 5.4: Process model at the end of the modeling phase (rotated)](image)

After completing the model of figure 5.4, it could be compared with the mined model of figure 5.3. Under the assumption that we processed the 4 alternative cases in an acceptable way, we checked if the flexibility offered by the mined model is also supported by the model displayed in figure 5.4. Of course, we should also check if undesired behavior of the mined model is impossible in the model of figure 5.4. The chance of e.g. deadlock is however quite limited in the BPMN model, since this model has mainly sequential tasks.
The BPMN model is mainly sequential, where the mined model has a lot of parallelism. The reason is that in the mined model no clustering was performed, where the manually developed BPMN model is largely clustered. Clustering is however mainly used to group the leaf data elements as can be seen in figure 5.4. This made it possible to compare these two models.

This comparison did not result in the discovery of errors, but some parts are worth mentioning. The manually developed BPMN model assumes that the end-users are identified on the network, so their personal data is filled in automatically. Also other system related data, like the list, becomes automatically available when required. This may prevent the kind of misinterpretations that appeared during the first (and the case processing part of the second) workshop.

The BPMN model also separates ‘primary’ data from the data that depends on the lists. Primary data is all data required to make a reservation for a room, except for the data that depends on the lists. Furthermore, in the BPMN model, the confirmation also depends on the reservation of equipment and services. This is not the case in the mined model (or in the PDM). Although this dependency does not exist in the PDM, this is included to check if the communication between the departments went as planned. Finally, the BPMN model contains ‘swim lanes’ that separate the departments. In this case the process design is mixed with the organizational structure. This can improve the communication, but also limits the freedom.

Since no errors were discovered, the model was ready to be used during the next workshop(s). However, as described earlier in this section, we processed only one case during the first workshop, so it was desirable to process some extra cases (during the second workshop). We therefore decided to start with processing some other cases, before the process model was presented. Showing the model could have influenced the cases that we still liked to process.

At the end of the second workshop we showed the process model to the participants. It proved that the model was understood and accepted on the major points by all participants. Only two small changes were requested. The first one is that the model contains the option to request a meeting per phone. However, this option was seen as an exception that should not be modeled explicitly. The second remark was that the messenger service can be removed from the process model, since this is only applicable to events which require a customized approach. Although the messenger service is part of the PDM, the involved department decided that it is not desired to include it in the final process design.

The proposed changes were modeled and missing information was added. The last refers to the task descriptions and role information, which was not included earlier in the project. This update resulted in the process design described in Appendix E. This process design was finally accepted. Details about the workshop setting that led to approval of the process are described in the next sub-section.

5.3.2 Communication platform of the method

In the previous sub-section we already referred a couple of times to this section, since the tooling is used to support the communication. Problems with the tooling will probably also result in problems in the communication and vice versa. This section evaluates the communication during the project at the province of North-Brabant.

Preparation of the workshops

Since the workshops have to fulfill a central role, they have to be prepared in advance. From a
communication point of view, this includes the preparation of an introductory presentation and the collection of historic cases.

The representative historic cases are important, since they are used to prepare the workshop and to support the discussion (with the tooling) during the workshops. We requested these cases from a number of participants by e-mail. We asked them to deliver realistic and diverse historical cases that the new process should be able to support. In order to discuss efficiently what level of flexibility the new process should support, an overview of what was required in the past was helpful. Historic cases are therefore a good means to support the workshops and its preparation. The e-mail we sent resulted in more than enough different cases to get the desired overview. Therefore the preparation of the workshops went as planned, which means that the workshop approach seemed to deliver the desired results until then.

The workshops
The goal of the workshops is to discuss what flexibility the process design should support. In this section we describe how well the workshops indeed supported this. Therefore, this section does not give details about the outcome of the project, but how the outcome was reached.

As the method describes, we preferred to plan the workshop well in advance. Planning a workshop proved however more difficult as expected. Since different departments were involved, it was hard to find a timeslot in which all participants could participate. As a result we mainly planned workshops of 1.5 hours, instead of the desired 2.5 hours. We also planned the workshops further in the future than desired. Planning the workshops that far in advance, resulted in other things that were given priority by the participants, which led to cancelation of a couple of workshops. From the 5 workshops that were planned, only 2 really took place with a total duration of less than 4 hours. The last workshop in which the process design should be approved was even canceled without being replaced.

To replace the missing approval workshop, we e-mailed the participants the process design (see appendix E). We asked all participants to react on the process design. This reaction could either contain an approval of the process design or comments that would possibly lead to changes in the process design. In total, this resulted in five (out of 13) responses, meaning that not all participants responded to the e-mail and the corresponding reminder. However, the participants who did react on the request gave some useful comments. Compared to an approval workshop, this comments seemed to be more personal.

Besides the fact that 3 workshops were canceled, also the composition of the participants changed over time. If only a limited number of participants could not participate, they were either replaced by a colleague of the same department or the workshop was held without a delegated from that department. This caused some communication problems. Participants who missed the first workshop had difficulties in understanding the goal of the workshops and the tooling. We assumed that the goal of the project in general and the tooling in particular was clear after the first workshop. Since the group of participants changed, this proved not to be the case for some participants.

That the goal of the project was not completely clear was also the result of unclear communication. The goal of the project is to reduce the number of errors made. However, during the second workshop the project leader communicated that the new process design should make it easier for the customers of the process (the secretaries who plan a meeting). To some extent this is the case, since less mistakes means less troubleshooting, what is also in their advantage. However, to streamline the process, it is expected that the secretaries will lose some freedom, for which they get better support in return. Currently they can request a
room by e-mail, phone, intranet, a document generated with Smart Documents [Sma10], etcetera. Some of these triggers are processed inefficiently. E.g. a document generated with Smart Documents is sent to the booking department, where it is printed and entered into another system. It is therefore expected that a couple of triggers will be removed. Removing these triggers does however not result in an easier process for the secretaries. From that we can conclude that the improvement goal was not clear enough.

Also the purpose of the tooling was not completely clear for all participants, since they tried to link the tooling with their current work (see previous section). Explaining the purpose better and more often might have helped, however adapting the tooling in the way as described in the previous section seems to be more effective since the misinterpretations appeared in all workshops. Constantly explaining the goal of the tooling seems undesirable, since it will require too much time. Also planning the workshops sooner after each other might help to improve understandability. In this case explaining the goal of the tooling in every workshop may not be required. The 3 weeks between both workshops were at least too long to remember the goal exactly. Further research is however required to validate this.

5.3.3 Main findings
The previous two sub-sections describe how well the tool-support and communication resulted in the desired outcomes. This section combines the results and presents the main findings.

We believe that the evaluation shows that the foundation of the method works, but that certain parts of the method and tooling were not completely understood by all participants. However, small changes in the method and tooling may drastically improve the user friendliness (and understandability) of the method. Since we evaluated the method only during one project, we have insufficient proof to draw solid conclusions. However, it points us in the direction of future work and some assumptions.

Although some parts of the method did not always bring what we hoped for, we were able to deliver a process design that was accepted by the participants of the workshops. Already during the second workshop we were able to present an initial process model that proved to be quite complete (only 2 small changes were proposed). Besides our method, we believe that the PBWD approach in general contributed to this success. The PDM enabled us to develop the process model with limited input from the participants, since we could combine this input with the structure of the PDM. Normally a workshop tends to result in a relatively high level process model. Because of the PBWD approach, we had almost all information that was needed to develop a more detailed process model after the first workshop. This had not been possible when a process based approach had been chosen. The second workshop was already enough to complete the process design (without the approval).

The main reason why workshops play a central role in the method is to support (or enable) the required communication. In our opinion we accomplished this, but we also learned that workshops have some disadvantages. In order to be really effective, a fixed group of participants is desired and the workshops should take place in a limited time span. In such a setting the best results can be accomplished. However, we did not succeed to create such a setting. Planning a workshop can be difficult for a larger group of participants, due to conflicts in agendas. The more people that are involved in the project, the higher the chance that not everyone can attend to the workshop. This should be taken into account already early in the project. Better project sponsorship from senior staff members may have prevented this limitation; however the project leader did not offer extra support. The project leader agreed that the workshops were not planned in an accurate way, but she believed that she could do nothing about it.
Furthermore, some of the workshops were canceled. This happened 3 (out of 5) times during this project. Most of them were replaced, but the last one was not replaced at all. This caused some difficulties. It might be the case that workshops do not perform well under some conditions. We still believe that workshops are required in this method, but maybe we can limit the number of workshops or the number of participants to streamline the process. Also extra project sponsorship from senior staff members is desired.

Related to the tooling of the method, we have to conclude that the system data elements in combination with the way in which some participants ‘treated’ the tooling was not a lucky combination. This caused discussions related to the goal of the tooling and project, instead of discussions related to the desired dynamics of the process design. This may be solved by a better explanation of the tooling. However since these discussion appeared in both workshops it seems better to prevent this from happening in a different way.

The easiest solution is to remove the system data elements. This can be done by only accepting PDMs without system data elements. However, we believe that excluding certain PDMs is not desired, since this will make the method less user-friendly. Removing the system data elements from the tooling seems therefore a better option. Also including a PDM viewer, as proposed in section 5.3.1, can be used to improve the tooling. The last may require more implementation time, but we expect the best results from this approach. The combination of the BPM|one form with a PDM that visualizes the current state may definitely improve the method, since it helps to relate the BPM|one form with the PDM instead of with the current process.

Overall, we believe that the method has the potential to result in the desired process design, but that some changes can improve the usability of the method significantly. In this section, suggestions for improvement are based on our own opinion. Section 5.4 describes how the users experienced the method and how they believe that the method can be improved. In section 5.5 we combined our vision with the vision of the participants to result in an overall evaluation of the project.

5.4 Evaluation with the participants

After the process design was accepted by the participants, we could evaluate the project (and thus the method) with them. We developed an evaluation form, composed out of 11 questions. This form can be found in Appendix F. This evaluation form was sent to a selected group of participants. Four evaluation forms were returned, of which the outcomes are described in this section. The evaluation form has two kinds of questions. The first group relates to how the workshop participants experienced the method and the second group of question relates to ways in which the tooling can be improved. The last was included since we learned from the project that the tooling was not always completely understood (user-friendly).

We received two comments related to the composition of the workshop participants. Since the number of participants was relatively high, with a relatively high number of inexperienced workshop participants, the workshops were not always as effective as they could have been. The changing composition of the participants also contributed to less effective workshops according to three of the respondents. Two respondents wrote down that it had been better if we had not invited everyone for every workshop. A smaller group of participants would have resulted in more efficient workshops, with fewer discussions that are not related to the goal.

The workshop setting itself was desired by three of the respondents. One of them especially mentioned that she was very happy with her involvement, although we did not ask this question (directly). The fourth
respondent agreed with the workshop structure of the project, but thought that the workshops were not efficient enough (for the reasons described above) and therefore not completely suitable.

Three of the participants were satisfied with the PBWD approach (or at least the focus on the product). The last had expected more of the lean method used. Although PBWD should result in a lean model, the approach is not based on the lean systematic. Except for one of the respondents, all participants agreed that the tooling contributed to the workshops, but that it took too much time to understand the goal of the tooling. The system data elements were identified as a possible cause by three of the participants. One of them suggested showing these data elements in gray. Two of these respondents think that the system data elements contributed to the limited speed in which the cases were processed. The tooling was however still experienced as a good way to test the reality. One of the participants stated: “It is a good method to test the practice, because many things where discovered that have to be taken into account”.

As expected based on the evaluation project, all participants had comments related to the usability of the tooling. Since we already experienced this during the workshops, we developed 4 alternative user interfaces. We focused only on a different form layout, since we thought that it was too hard to measure the effects of an extension to the tooling with e.g. a PDM viewer.

Another participant suggested using no tool-support at all, so she rejected all alternatives. Her opinion is that the tooling is too technical to use during the workshops. All other participants who contributed to the evaluation see version 1 and 3 as an improvement (see appendix F). Version 1 is a user interface in which all system data elements are removed. A screenshot of this user interface can be found in figure 5.5. Version 3 has a comparable user interface, but in this case also a number of the edit boxes is replaced by dropdown boxes or selection boxes. E.g. the ICT equipment can be chosen in such a way. In this version we combined a system data element with a selection data element (see appendix F). The interface of figure 5.5 can be reached by applying the formula of section 5.2 and afterwards restore the dependencies (leaf, automatically derived, manually derived). This can be seen as a relatively easy task.

![Figure 5.5: Improved user interface](image)

Overall, we received some valuable input from the respondents. The comments we received were largely consistent with each other. The workshop setting was desired by most respondents, although it was
suggested to be (more) careful about who to invite for the workshops. The respondents saw the (potential) power of the tool-support, but agreed that the tool can (or has to) be improved. Deleting the system data elements from the user interface may already deliver the desired user friendliness. This evaluation gave us some nice directions for further work. In the next section we combine our own findings during the workshops with the experiences of the participants that helped us to evaluate the project.

5.5 Results
Throughout this section, the main findings of the evaluation are described. Since this evaluation is based on only one project, one has to be careful with interpreting the results. Some of the outcomes may be caused by the setting in which the evaluation project is conducted. Also the number of returned (4) evaluation forms is limited. A more decent evaluation is therefore desirable, to verify these results.

Tool-support
The evaluation showed that the tooling can be used in the way as intended. However, the evaluation also showed that it can be hard to use the tooling efficiently during the workshops. We believe that there are two main reasons, which strengthened each other. In the first place, some participants found it hard to ‘forget’ about the current process. They tried to link the tool-support directly with their current work. They also saw the tooling as a replacement of the current system. Although treating the tooling as a system that will be replace the current system in the near future is not desired at all, it does not have to influence the outcomes. However, since the tooling was not optimized for this (second reason), some participants had the feeling that the new process would lead to a decline in quality compared to the current process

The system data elements were mainly responsible for the non-optimal tool-support. The tool requires that a textbox is filled to represent a list. This is not the case in the current process and neither will it be the case in the new process. A printed version of the PDM could not prevent this from happening, since switching between the tooling and the PDM on paper did not seem to work.

Preventing the users from thinking in an evolutionary way without additional support seems to be impossible. Therefore, we believe that the tooling should be improved to become more user-friendly. We believe that this can be achieved by deleting the system data elements from the tooling. Also the evaluation with the participants suggests that removing the system data elements will improve the usability of the tooling. Removing the system data elements from the tooling seems to be easy to realize. In that case we do not have to put additional restrictions on the PDM that serves as an input for the method.

However, we believe that visualization of the link between the PDM and the tooling will be even better. Based on the literature review, we learned that visualization is a strong mean to improve understandability [The09]. Also the process designer of the province of North-Brabant believes that visualization can be used to improve the tooling. Not all PDMs will contain system data elements, so improving the tooling on this point may have a limited effect. We expect that a clear link between the PDM and the tooling can limit the ‘risk’ that participants think in an evolutionary way. This link should look like the PDM viewer as proposed in section 4.1.1. The combination of the current tooling and a PDM viewer may drastically improve the user friendliness of the tooling, since this combines the case driven approach of the tooling, with the visualization power of the PDM. Unfortunately we have no feedback from the participants related to a PDM viewer.

Unfortunately, we did not completely succeed in testing the process mining part of the method. Due to a lack of time during the workshops, we did not process enough cases to mine a model during the workshop. Processing additional cases afterwards, supported us during the modeling phase, but since this was a
deviation from the method we will not draw any conclusion related to the process mining step of the method.

Communication platform
The second pillar of the method was the workshop approach. The workshops were used to communicate with the end-users what the process design should look like. However, the workshops were not always as efficient as they could have been. We identified a couple of possible causes. The most important one of these concerns the understandability of the tooling as described in section 5.3.1. This was however not the only cause. Since it was difficult to plan the workshops, the time between them was too long. This resulted in participants that ‘forgot’ the goal of the project. We did not communicate the goal clearly enough at the start of every workshop. The project leader communicated the goal again during the second workshop, but formulated it in a slightly different way.

Besides the limitations during the workshops we also had some difficulties with planning the workshops. Since the group of participants was relatively large, it was difficult to find a free timeslot to plan the workshop. Some of the workshops were canceled, of which the last was not replaced. This meant that we had to ask for approval of the process design by e-mail. Also the duration of the workshops was therefore sometimes limited to 1.5 hours, instead of the desired 2.5 hours.

Nevertheless, the participants that took place in the evaluation process were pleased with their involvement in the project. Furthermore, it was possible to deliver an approved process design, after only two workshops with a total duration of less than 4 hours. We therefore believe that the choice to use workshops for communication purposes was a good choice. However, the way in which the workshops are organized can be improved. Limiting the number of participants in some of the workshops is one of the options. Instead of one workshop with all participants, it is possible to organize two smaller workshops with a limited number of participants. Less workshop participants could result in more effective workshops that are also easier to plan. This may however largely depend on the organization under consideration. If the organization has more sponsorship for the project, the results may (or will) improve.

Conclusion
Overall, we believe that the method can improve the usability of PBWD, and that it therefore can contribute to a better acceptance of PBWD. However, some parts of the method did not completely behave as expected. With both the tooling and the workshop approach we experienced some limitations. Changes to the method are therefore desired. All these findings are however based on only one evaluation project. An evaluation within another sector, or even within the same sector, could have resulted in completely different outcome. Since the limited time available for this project, we did not have the opportunity to evaluate (an adapted version of) the method at other place, with another PDM. Therefore, one has to be cautious with drawing conclusions.
6 Conclusion and future work

This final chapter concludes the thesis at hand. It presents the outcomes of the project. Section 6.1 elaborates on answering the research questions postulated in section 1.2.2. By answering the research questions we can conclude whether the goal the project is met. In section 6.2 we compare our work with other work in this area. This thesis will finally be completed with possible directions for future work in section 6.3.

6.1 The project

In this part of the thesis we will conclude if the goal of the project is met, by answering the research questions. The first research question deals with ways in which tooling can support the usability of the method. We developed 3 possible approaches, all with their own tool-support, from which one approach was selected for further development. Based on the evaluation project we believe that the developed tooling supports the usability of the method. The tooling offers an efficient way examine the effect of a decision, since the tooling ‘makes’ the PDM executable. Therefore, it becomes easier to examine and discuss which level of flexibility is desired by the process design.

However, the tooling as-is has some disadvantages that reduced the level of understandability. The interface of tooling looks like an application form according to some of the participants. Therefore they treated the tooling according to that feeling. This caused some misinterpretations, since various data elements do not belong on an application form. This reduced the effectiveness of the tooling. Adapting the tooling is therefore desired, since the tooling should be understandable (self explaining) for inexperienced users. Section 5.5 already describes what can be done to improve the tooling. In section 6.3 this is translated to directions for future work. Without these adaptations the tooling can be used, but in that case the goal of the tool-support should be explained very carefully.

The second research question was used to examine what the communication platform should look like. In general, a PBWD project requires communication between process designer and the end-users. This is also the case in the method we developed. We chose to use workshops for communication purposes, since workshops offer a good platform for discussion. We believe that discussions are required to ‘discover’ the right level of flexibility in the process design. Since most process designers will not have a detailed knowledge about the process, they need the end-users to determine which dynamic behavior should be offered by the process design. Involving the end-users in the project will also create user commitment. Based on the evaluation we conducted, we believe that the choice for a workshop approach fits our goal quite well, since the workshops resulted in discussions about what should be supported by the process design.

Nevertheless, the workshops were not always an easy instrument. Planning a workshop proved to be a difficult task, due to conflicts in agendas. A number of workshops were even canceled for this reason. Also the workshops itself were not always as efficient as they could have been. The relatively large number of participants resulted sometimes in unstructured discussions, or discussions that not contributed to the goal of the workshop. Some participants suggested that not all participants should be invited for every workshop. Despite these undesired side effects, we were able to deliver a process design that was approved by the participants in less than 4 hours of workshops. Therefore, the workshop approach is seen as a good communication platform for this method.

With the research questions answered, we should be able to conclude if we succeeded in the goal of this project:
Develop a user-friendly method that guides the conversion of a PDM into an accepted process design and therefore encourages the use of PBWD outside the academic world.

During this project, we developed tool-support and a communication platform that should be combined into a user-friendly method. Based on the answers on the research questions, we can conclude that the combination of the tooling and the workshop approach resulted in a (potentially) user-friendly method. We included the word potentially between brackets, since the method can definitely be improved on some points. Most of the proposed changes are easy to realize and even without these improvements the method can be used efficiently, as long as one is aware of its limitations.

As stated in the goal, the method should finally encourage the use of PBWD outside the academic world. We believe that this goal is definitely achieved. Our research partner, Pallas Athena, is satisfied with the result. They are currently investigating what role PBWD can play within their organization and products.

6.2 Comparison with related work

In this section we will relate our work with earlier research performed in this direction. Most important research in this area is conducted by Vanderfeesten [Van10]. Together with Kamphuis and Reijers, she developed most of the conversion algorithms described in [Kam08]. Direct execution of a PDM is also based on her work [Van09b]. We will compare our work with these two projects.

The conversion algorithms offered the first support for the automatic conversion of a PDM into a process model. Before the development of the conversion algorithms, converting a PDM into a process model was a manual and time consuming task. Compared to our method, the conversion algorithms directly result in a process model. Our method requires at least one workshop before the first results are visible. However, the conversion algorithms only produce an initial process model, which needs to be adapted afterwards. It is also not clear which algorithm to use when. Our method aims at a complete process design. The choice between the conversion algorithms and the method described in this thesis will depend on the desired deliverables of the project.

Direct execution of a PDM by means of a recommendation system may be better comparable with our method. Both techniques aim at controlling the execution freedom of the PDM, by involving the end-users. None of these approaches is always better. Both techniques are in an early stage of development. The major difference is that a recommendation system tries to limit the execution freedom at runtime and the workshop approach tries to limit the freedom at design time. Which approach is better might depend on the end-users. A recommendation system might be overruled, where an implemented process design cannot be overruled at runtime anymore. A completely satisfying recommendation system is still missing, but also the tooling developed for the method needs to be improved.

6.3 Future work

In the final section of this thesis we propose several directions for future work. We will mainly focus on extensions of our method. The most desired extension of the tooling is a PDM viewer that shows the actual state of a case. We assume that a PDM viewer can drastically improve the usability of the prototype. Based on the literature review, we believe that a layout as proposed in section 4.1.1 will do the job. A PDM viewer is also a first step to a PDM editor. We expect that both a PDM viewer and editor are required to make PBWD really user-friendly.

How to handle dependencies between cases is another direction that is worth a closer investigation. PBWD does not offer (good) support to show dependencies between cases. However, such dependencies can (or
even will) exist in practice. In the evaluation project, a data element called ‘Earlier bookings’ was added in the PDM in order to indicate the relation with other cases. This solution may work, but a more standard solution is desired. Also a standard solution for other kinds of system data elements is desired. This can prevent that extra restriction on the input PDM are required, or that the tooling has to show a model that deviates from the PDM. Standardization can prevent the need to develop individual solutions. A standard solution will at the end result in a better understandability, since users better know what they can expect.

Also the workshop setting may require an update. The workshop approach seems to support the method, but planning and conducting an efficient workshop is not an easy task. The books of Jeston et al. [Jes06] and Sharp et al. [Sha01] describe the structure of a good workshop. Nevertheless, we experienced some difficulties with the workshops during the evaluation project. Since these difficulties are largely depending on the organization, most literature is a bit vague about major questions, like who should be invited and how far in advance a workshop should be planned. These factors are however of great influence on the project outcome, so further research in this direction is desired. This can result in e.g. better guidelines that support the process designer (or project leader) in selecting the right participants.

Furthermore, since the method has been evaluated during only one PBWD project, it is desirable to perform a larger evaluation. Where possible after the method is improved at one or more of the points described above. In this case the results of various projects can be compared to improve the method even further.

Finally, it is important to spread the news of PBWD. Besides the developed method, we made a start with several presentations related to PBWD within Pallas Athena. If the larger BPM suites and consultancy firms adapt PBWD, it can become an adapted approach in the coming years. This also requires however that the development of PBWD continues.
References


Appendix A: Method description

Introduction
This document describes a method to translate a Product Data Model (PDM) into an efficient process design. It is intended to support Product Based Workflow Design (PBWD) projects. Without this method, the conversion of a PDM into a process design is either a manual job without (enough) guiding or does not necessarily result in a (near) optimal process design. This document is mainly intended for the process designer. The process designer can be an external or internal consultant. However, if the company under consideration is new to PBWD (or even to BPM) an external consultant with experience in the field of PBWD might be preferable. The process designer is primarily responsible for the steps described in this document. He also decides (together with the management) who should be involved in the project.

Figure A.1: A PBWD project structure

In figure A.1 the main steps of a PBWD project are shown. After the decision is made to start a Business Process Redesign (BPR) project, a project team and a clear goal should be formulated. The goal depends on what the organization wants to improve. Examples of goals are cost saving, reducing the processing time, improving customer satisfaction, etcetera. This goal should be formulated in cooperation with the management. Management support is essential when it comes to a successful project execution. If the management does not see the added value of the project it is doomed to fail.

Besides management support, the end-users should support the final implementation. To make this happen it is desired to involve end-users as early as possible. In most cases this can start during the PDM identification phase. The PDM will serve as an input for the phase in which the process design should be discovered. Finally the process design should be implemented within the organization.

The method described in this document is focused on the phase in which the process design should be discovered. The steps in this phase are shown in figure 2. In the preparation step the process designer creates an executable model that allows for all execution paths that are possible to produce the end-product according to the PDM. The method offers tool-support to accomplish this. Furthermore the process designer should prepare the workshop of the next step.

Figure 2: The steps required to translate a PDM into a process design

Goal of the workshops is to limit the execution freedom in an efficient and structured way (e.g. to minimize costs or processing time). Complete execution freedom is undesired in most situations. The goal of the first workshop is to come up with change requests that should limit the execution freedom.
After this workshop the process designer should process these change requests. This should result in a process design that meets all requirements according to the process designer. However, this method does not accept that the process designer approves the process design himself. This should be done by the end-users in another workshop. This can lead to acceptance of the model or new change requests. At the end a process design with the right level of execution freedom should be accepted. In the remainder of this document the steps of the method are described in more detail.

**Prepare first workshop**

As described, this method starts when the PDM is finished. This section explains the steps that have to be performed during the preparation step. Workshops fulfill an important part of this method. Goal of the workshops is to discover and approve the final process design. In the preparation step the 1st workshop is prepared. This includes:

- Automatically translate the PDM (with any available XSLT parser together with the XSLT file belonging to this method)
- Update the chp file in the following way:
  - Import chp file
  - Order static plan (top – down, thus as an inverted PDM) to make all nodes visible
  - Fill in expressions for derived variables (the comment should explain how the variable can be derived)
  - Optionally group data elements in a logical order on the form
  - Release plan (and perform some testing in the runtime environment)
- Plan workshop (invite people, arrange necessary equipment, etc)
- Prepare workshop
  - Make an introductory presentation
  - Think about possible improvements that limit the execution freedom in an effective way and work out these alternatives (use the modeling guidelines described in the next section)
  - Test equipment used during the workshop
  - Gather real and realistic cases that can be used during the workshop

After this step the model that allows for direct execution of the PDM is ready to be used. Furthermore, the process designer has thought of possible ways to limit the execution freedom and prepared the workshop of the next step.

**The first workshop**

In this section the steps that have to be taken during the first workshop are explained. A possible agenda for the first workshop could look like this:

- Opening presentation
- Show PDM
- Start Prototype
- Let workshop participants experience the freedom of direct execution and discuss where and how to limit the execution freedom (under supervision of the process designer)
- The process designer should fill in change request forms (during the discussion) when all decision makers (often a combination of end-users and their managers) desire a change. This form can be found in Appendix B.
Based on all ‘allowed’ execution paths an initial process design should be mined at the end of the workshop.

Finally, the process designer should give a summary (based on the mined model) and thank the participants.

In the remainder of this section these agenda points are explained in more detail. The workshop should start with a (PowerPoint) presentation given by the process designer. He should explain how the workshop will run, who is participating in the workshop and why (their roles), the method that is used and the relation between the prototype and the end-product.

Next, the PDM should be shown to the workshop participants. If the participants are already familiar with the PDM, this will bring the PDM under the attention again. The process designer should explain that the goal is to produce a process design that is an improvement in comparison with the current process. This should be done by discussing which execution paths to produce the end-product are desired. If the current process is (digitally) available (preferably, in the used modeling language), this process model can also be presented (e.g. to indicate its weak points).

Before the discussion starts, the process designer should make sure that all participants understand the goal of the workshop and expect the same from the workshop. The discussion starts with an introduction of the prototype. We propose to use a beamer, for which we split the projection area in two. The first part should show the executable model (the BPM|one form with the data elements and its progress indicator) and the second part to show the PDM. If splitting the screen is unpractical, one can also choose to switch between the screens or to use a second beamer.

The prototype will be used to support the discussion about which execution paths are desirable. This should be done by letting the end-users and their managers “play” (under supervision of the process designer) with the prototype and let them experience the freedom that is possible according to the PDM. This can be done as follows: the process designer should create a case instance (based on a real and realistic case), decide together with the participants how to process the case instance and at the end decide if the chosen execution path should be allowed in the final process design. Not all execution paths chosen in the workshop should result in a desirable execution path. The prototype should also be used to show the disadvantages of particular executions. Also paths that seem to be desired at a first glance might be shown to be less optimal later on.

The decision if an execution path is desired can be supported by the guidelines given below. Besides changes that directly lead to a limitation with the goal in mind, one can also propose changes that simply feel more “natural”. For example it might be logical to first enter the personal data of a claimant in case of an insurance claim and thereafter the claimed amount. This is however not necessary according to the PDM and might not directly result in a faster or cheaper execution. The discussion can be supported with e.g. a whiteboard or flip over. It can be used to write down opinions and comments, possible reasons for a knock-out, etcetera.

Undesired execution paths should lead to change requests. A change request in its turn should result in a change in the process. A change request should only be made if everyone (with decision power) agrees with it. This prevents unwanted change requests or change requests that are in contradiction with each other. The process designer is responsible to log them on the template belonging to this method. This template is a formal part of the method and can be found in Appendix B. Also the guidelines (see below) are quite formal.
However, the process designer should explain these guidelines in terms (a language) that the participants understand, to make the workshops efficient and attractive.

This method does not require logging all desired paths and the reasons why they are desired. However, we advise to log decisions about desired paths that have lead to discussion and/or paths that are important for most (standard) process executions.

The decisions if a path is desired can be based on the following guidelines:

- Make use of best practices (as described in: Best practices in business process redesign) [Lim07]
- Look at industry standards, benchmarking, etcetera where possible
- Look at possible performance indicators (both quantity and quality)
- Start with focusing on the normal process flow (applicable to most case instances) as long as this does not limit possible exceptions too much. Next, the less often executed paths should be considered
- Resulting tasks should be performed by one role (tasks should be atomic)
- Minimize the number of tasks since this will result in the best understandable process design
- If there are known bottlenecks in the current process think about possible ways to solve them (by looking at the cause of the bottleneck)
- Desired execution paths of the current process should remain possible in the new process design
- Automated tasks should remain in parallel (assuming zero cost and time)
- Don’t model the universe (too much detail will only result in extra costs)
- Think about the scope. What is out of scope?

Some of the guidelines have their origin as modeling guideline. However, it is not the task of the participants to model during the workshop. This should be done by the process designer afterwards. Nevertheless, these guidelines are already important in this stage, since they can support the discussion and since the modeling task will be easier if some modeling principles are already taken into account by the process designer. Since the workshop participants are not (necessary) familiar with modeling, the process designer should use these guidelines in the background. If necessary he should translate them to a language that the participants understand. In this case he can use the modeling guidelines without required modeling knowledge of the participants.

At the end of the workshop the process designer can automatically generate a process design (BPM|one model) based on the desired execution paths. This model cannot be seen as a final process design, since e.g. clustering and knock-outs are not taken into account. Also, the task descriptions and role information is missing. However, the process design can serve as an input for the next phase, in which the designer should process the change requests. Furthermore, the mined model should be shown to the participants in order to indicate how much progress has been made during the workshop. This keeps participants involved. The mined model may be left behind at the customer (user) site, for further internal discussions.

**Model process design**

In this step of the method, the mined process model should be combined with the change requests. The modeling task can be performed in any modeling language. However, the process mining step performed during the first workshop will result in a BPM|one model. If another modeling language than BPM|one is chosen, the mined model should be converted.
The mined model should be combined with the change requests captured in the previous phase. It is advisable to use the process modeling guidelines described in the previous section as support. However, since the modeling task will largely depend on the chosen modeling language, this method description does not go into detail about the modeling step. We therefore refer to the language description of the chosen modeling language. In the case BPM|one is selected, the language description can be found in [Pal09]. To make the process design complete, descriptions of the tasks and role information has to be added. After the process design is completed, it should be tested to confirm if the resulting process design conforms to all change requests.

**Approval workshop**

The process design that is modeled during the previous phase should be discussed during an approval workshop. This discussion should result in a process design that is either approved or rejected by the participants. If the process design is rejected, a new version should be made and the rest of the steps should repeat itself. These steps will repeat itself until the participants accept the process design.

The structure of the approval workshops should be comparable with structure of the first workshop. However, some parts can be left out, like the process mining step. Most important is to plan the workshop on time and to explain that the goal of this workshop is to ask for approval of the process design. It should be clear that after the approval of the process design, no changes will be made anymore.

After the process design is accepted, it can be implemented. The implementation itself is not a part of this method. However, since a PDM can change over time we included a part about maintainability in the next section in order to handle with these changes.

**Maintain process design**

It can (or even will) happen that the product and thus the PDM changes after a process design is implemented. Possible reasons are legal changes, new company rules, innovation, etcetera. A change in the PDM will definitely require a change in the corresponding process design.

After a change in the PDM, one can discover the new process design by starting all over again. However, this will cost too much effort, since decisions made in the past are not taken into account anymore. Especially for large PDMs that change periodically, the time required to agree on a new process design should be limited where possible.

Limiting the required time is possible when the relation between the PDM and the process design is understood in enough detail. When a small change is made in the PDM, it is possible to locate the region(s) of the process design to which this change has effect. In this case the regions of the process design that are not affected by the change can remain untouched, where regions that are involved should be reconsidered.

For the regions that are affected, earlier made change request have to be reevaluated. One should investigate if they still hold. In this manner it is not necessary to reevaluate the whole process design, but only the parts that are affected.

The evaluation should be performed in a similar way as the initial process design is discovered. The process designer should think about the impact of the changes and organize a workshop. In this workshop the effect of the changes should be discussed and new change request should be made.
In order to be able to do the above in an efficient way, change request (or notes) of earlier workshops should be available at all times. This means that decisions made during the workshop should be logged on a change request form. We propose to do this in a digital format where possible. This should not cause any problem since the change request template and the tooling is digitally available. In this way PBWD can be an effective approach in any BPR project.
Appendix B: Template for change requests
This document serves as a template for change request. If a change request is desired it should be logged on
this template. The template should be filled by the process designer, if a change request is made during the
workshops. This helps (the process designer) to keep track on changes. This is needed to implement the
change request, but also to track the changes made if the product (and PDM) changes.

Project name:

Project number:

Date workshop:

Change request name:

Change request number (decide on format per project/organization):

Change request description:

Status of the change request (update after changes):
Appendix C: Tool-support

In this appendix we describe the most important decisions that are made during the development of the tooling. In order to achieve the best possible results with the tooling, we had to make some design decisions during the development. The first section describes the selection of the development platform. In the section thereafter, the required mapping to convert a PDM into an executable model is described and finally the decisions made during the development of the mining functionality are described. This appendix is therefore be quite technical.

Choice for a development platform

The tooling should support the translation of a PDM into a BPM|one model. Using the resulting model itself should not cause any trouble, since BPM|one is already widely used. The desired layout of the form was already set, so we knew exactly what we had to develop.

We examined three ‘development’ platforms: the BPM|one environment, ProM and XSLT. Extending BPM|one with a PDM converter would integrate all tool-support into one software suite. However, we quickly learned that developing an extension to BPM|one (or at least this extension) requires quite some knowledge of the BPM|one structure. Our research partner has this knowledge, but the developers could only offer us a limited amount of their time for support.

The second platform we examined is ProM. ProM has already a large selection of conversion algorithms, so we wondered if we could easily realize our conversion algorithm in ProM. Another advantage is that ProM can already visualize PDMs. ProM definitely offers the possibility to convert the PDM into a BPM|one model, but also here a certain knowledge of the architecture is required.

The last possibility we examined was XSLT. Since a PDM is stored in a XML format, a XML conversion language, like XSLT, could do the job. We quickly learned that XSLT offers good support to translate a XML file into another text format. Since a BPM|one model is stored as a text file, we chose to continue with XSLT. An advantage of XSLT is that it does not require any knowledge about the architecture of BPM|one or ProM, as long as we know what the output should look like. We chose XMLSpy as XSLT parser, since this package is advices in various literature. However, every XSLT parser will do the job.

Conversion decisions

After selecting a development platform, the development of a conversion algorithm could start. We had to decide how the mapping between a PDM and a BPM|one model should look like. All constructs of a PDM should be mapped on a corresponding construct in BPM|one.

One of the most important elements of a PDM, the data elements, were mapped on data elements in BPM|one. The only difference between these two data elements was that a PDM does not store the data type of its element. In BPM|one a data type is required for every data element. We therefore decided to extend the XML format of a PDM with a data type. As long as this is an optional field, it will not harm other applications that use the PDM data structure. Furthermore, an extension with a data type seems to be a logical extension.

Another important construct of a PDM is the operation. Since we wanted to make the full dynamics of a PDM available in BPM|one, there was no need to cluster the operations. We therefore decided to map a single operation in a PDM with a single activity in BPM|one. In order to separate the kind of operations, we extended the PDM structure with the kind of the operation (leaf, automatic, manual). However, as explained
in chapter 4, we decided not to fill in the formulae for automatic operations, since this can easily result in errors in the model.

The most difficult and final construct we had to map, was the XOR join of a PDM. A PDM allows that a data element can be calculated in different ways. As an example we refer to section 4.3. The root element of the PDM discussed in this section can be calculated in 3 different ways. We believe that the case should finish after the root element has been calculated. However, we could not map the XOR construct to a direct construct in BPM\textsuperscript{one} (at least not in the way as we desired it). To develop a decent solution, we had to split the problem. By treating a data element with a XOR differently, we could map these data elements with more than one (the number of incoming arcs + one) data element in the BPM\textsuperscript{one} model. In this way we could develop a model, in which every arc of the XOR is connected with its own data element. The remaining data element is used to connect the extra data elements with the rest of the model.

This did not yet completely solve the problem, but it helped us in developing conditions, that can be used to finish the case after the end-product has been reached. This result is acceptable for us (especially for a proof-of-concept), but these conditions do not work for XOR joins in the middle of the model. If a sub-product should also mean that its predecessors cannot be calculated anymore, a construct like “conditional mandatory” is desired. Such construct does however not yet exist in BPM\textsuperscript{one}. It is also possible to use the dependencies of the PDM for this task, but this means that the mapping between constructs will become more complex.

**Mining a model**

BPM\textsuperscript{one} already offered process mining functionality, which we could use for this project. However, we quickly learned that we could not use the tooling to extract the cases from the database. We need to extract these cases, since they serve as input for the process mining step. The available tooling is able to extract information related to executed activities from the database. However, the conditions described in the previous section, result in the execution of all activities. This does necessary means that the corresponding data element is always changed. The already available tooling could not separate if the execution of the activity resulted in an adaptation of the data elements (we are only interested in these activities).

Therefore, we had to develop our own queries, which focus on the data instead of the activities. Since all data is stored in a SQL database, we were able to develop queries that could be used to extract the right data from the database. To make the use of these queries as user-friendly as possible, we put the queries in a scripting file. Running the script automatically results a file that can be imported in BPM\textsuperscript{one}. 
Appendix D: Cases used in the mined model
This appendix describes the cases that are processed during the first workshop and directly thereafter. As described in section 5.3.1, only one case was completely processed during the workshop itself, so it was not useful to mine a model on that case alone. However, various alternatives were discussed during that workshop. We processed these alternatives after the workshop in order to be able to mine a useful process model. Both the case processed during the workshop and the 4 alternatives are described in this appendix.

Case processed during the workshop
As can be seen in figure 5.1, the PDM contains a couple of system dependent data elements. Examples are: ICT available, AV available and Earlier bookings. Furthermore, the process designer told that it is advisable to start with the ‘Applicant’, which could be derived from the information system. Also the lists can derived automatically (in the final system), when the user is logged in to an information system. The rest of the input data elements can be filled in an arbitrary order. In a certain sense the participants were therefore triggered to think in terms of an implementation. Nevertheless, it seems desirable to start with the applicant and the system data elements. Processing this case resulted in the following sequence:

Applicant -> ICT available -> AV available -> Catering list -> Earlier bookings -> Possible budget numbers -> Meeting name -> Data -> Start time -> End time -> # participants -> GS/CvK present -> Catering wishes -> Catering time -> Budget number -> Cost centre -> ICT equipment -> AV equipment -> Desk location -> Messenger service wish -> Room -> AV and ICT delivery -> Confirmation -> Catering -> Messenger service -> Meeting

The processed case shows a particular order. First, the system data elements were entered and directly thereafter the other leaf elements. It seems that the most ‘important’ leaf elements were entered first. After all leaf elements were entered the room reservation was made and the rest of the case was processed.

Alternative cases discussed during the workshop
As described, various alternatives on the case execution were discussed during the first workshop. These variants mainly related to the order in which the leaf elements have to be entered. Although it was explicitly mentioned that the tooling supports an arbitrary order, we received questions related to the flexibility of the tooling (or the PDM).

The first relates to the static data element. The participants agreed that the system data elements have to be entered at the start of a case (or better they should appear automatically, but the tooling does not support this). We therefore processed these data elements in an arbitrary order after data element ‘Applicant’ was entered. One of the participants asked why we chose this order, instead of another one. We explained that we processed them in an arbitrary order and that we could also start with, for example, the catering list. Although we expected that it will be desirable to cluster the system data elements in the final process design, we wrote down this alternative. This resulted in the following sequence:

Applicant -> Catering list -> Earlier bookings -> ICT available -> AV available -> Possible budget numbers -> Meeting name -> Data -> Start time -> End time -> # participants -> GS/CvK present -> Catering wishes -> Catering time -> Budget number -> Cost centre -> ICT equipment -> AV equipment -> Desk location -> Messenger service wish -> Room -> AV and ICT delivery -> Confirmation -> Catering -> Messenger service -> Meeting
Another applicant asked if it is possible to start the processing of the cases with the financial related data elements. We do not know why this is desired, but locating the right budget number is sometimes a ‘difficult’ task. It may be desired to start with these data elements, to finish this part as early as possible. We however believe that this question was asked to test the flexibility of the tooling. This case resulted in the sequence:

Applicant -> ICT available -> AV available -> Catering list -> Earlier bookings -> Possible budget numbers -> Budget number -> Cost centre -> Meeting name -> Catering time -> ICT equipment -> Data -> Start time -> End time -> Catering wishes -> # participants -> GS/CvK present -> AV equipment -> Desk location -> Messenger service wish -> Room -> AV and ICT delivery -> Confirmation -> Messenger service -> Catering -> Meeting

The third alternative relates to the data elements that depend on a list. They appear in the right column (manually derived data elements) of the tooling, since they are manually derived. However, the participants wondered why they were not located between the ‘other’ input data elements. We explained that this is caused by the lists, but that these data elements can be entered as soon as the corresponding list is entered. This kind of execution can look like:

Applicant -> ICT available -> AV available -> Catering list -> Earlier bookings -> Possible budget numbers -> ICT equipment -> AV equipment -> Catering wishes -> Meeting name -> Data -> Start time -> End time -> # participants -> GS/CvK present -> Catering time -> Budget number -> Cost centre -> Desk location -> Messenger service wish -> Room -> AV and ICT delivery -> Confirmation -> Catering -> Messenger service -> Meeting

Finally we received the question if it is possible to start with the data elements that do not relate to the reservation of a room. In all other alternatives discussed, the reservation of a room seemed most important. In a certain sense this is true, but it is also possible to start with the catering time or the messenger service wish. The corresponding sequence looks as follows:

Applicant -> Catering list -> Earlier bookings -> ICT available -> AV available -> Possible budget numbers -> Catering wishes -> Catering time -> Messenger service wish -> Meeting name -> Data -> Start time -> End time -> # participants -> GS/CvK present -> Budget number -> Cost centre -> ICT equipment -> AV equipment -> Desk location -> Room -> AV and ICT delivery -> Confirmation -> Catering -> Messenger service -> Meeting

As described we have processed a case for every alternative to show the possibilities. We assumed that it is desired to start with the applicant and the system data elements in all cases. Therefore, some alternatives were interpreted as: first the applicant and system data elements and directly thereafter the alternative under consideration. Since we had only the information described above, we could not check this assumption and we had to guess how the rest of the data elements should be processed. Since the process seems quite sequential, we believe that this could be done in an acceptable way. Processing these alternatives resulted in the mined model of figure 5.3.
Appendix E: Process design sent for approval (in Dutch)

In deze sectie wordt elke processtap uit het procesmodel in Figuur 1 beschreven. Van elke stap beschrijven we wie hem uit dient te voeren en wat er moet gebeuren.

Figuur 1 – Procesmodel

1. **Gebruik login gegevens als contactgegevens**
   
   **Wie:** Aanvrager/systeem.
   
   **Wanneer:** Eerste stap in het proces.
   
   **Wat:** De aanvrager vraagt het aanvraagformulier op (opent het online). Hierbij worden automatisch de gebruikersgegevens opgehaald, en als eerste gegevens automatisch ingevuld op het formulier.

2. **Voor een ander persoon?**

   **Wie:** Aanvrager.
   
   **Wanneer:** Meteen na het openen van het formulier.
   
   **Wat:** De aanvrager wordt geconfronteerd met zijn of haar gebruikersgegevens, en voor de optie gesteld om de aanvraag namens een ander persoon te doen, en daarom dus de automatisch ingevulde contactgegevens, te wijzigen.
3. **Wijzigen contactgegevens**

Wie: Aanvrager.
Wanneer: Indien ervoor gekozen wordt niet zelf als contactpersoon op te treden.
Wat: De aanvrager wordt geconfronteerd met zijn of haar gebruikersgegevens, en voor de optie gesteld om de aanvraag namens een ander persoon te doen, en daarom dus de automatisch ingevulde contactgegevens, te wijzigen.

4. **Invoeren primaire gegevens**

Wie: Aanvrager.
Wanneer: Na het eens zijn met de (al dan niet zelf) ingevulde contactgegevens.
Wat: Allereerst maakt de aanvrager hier een keuze voor een “type aanvraag”. Er kan een vergadering (standaard) gekozen worden, of een vergadering met lunch, maar eventueel ook een lunch, terwijl de vergaderruimte (kamer van een directeur) al op eigen gelegenheid is geregeld. Tenslotte is er nog de keuze om een evenement aan te vragen.

De aanvrager vult hier alle “primaire” gegevens in, zijnde:
- Naam van de bijeenkomst
- Datum van de bijeenkomst
- Begintijd
- Eindtijd
- Aantal deelnemers
- De aanwezigheid van een lid van GS, of de algemeen directeur (met naam)

5. **Invoeren wensen op basis van lijsten**

Wie: Aanvrager.
Wanneer: Na of tegelijk met het invullen van de primaire gegevens (afhankelijk van techniek).
Wat: Naar aanleiding van de ingevulde gegevens, en faciliteiten die gekozen worden (ICT/AV-middelen), kan hier een aantal gegevens aangevuld worden:
- Tijd catering, indien van toepassing
- Keuze type catering, uit een (zeer beperkte) lijst met mogelijkheden. Daarnaast een optie voor overige en een invulveld, met een verwijzing naar de assortimentsgids.
- Keuze uit een aantal ICT en audiovisuele middelen (een beperkte lijst met de veel voorkomende middelen). Andere zaken zijn nog aan te vragen in een invulveld, met een verwijzing naar de PNB catalogus.
- De kostenplaats. Deze is al ingevuld op basis van de gegevens van de aanvrager, maar kan eventueel ook gewijzigd worden (het is mogelijk een verwijzing naar de mogelijke kostenplaatsen te volgen, ter ondersteuning van het achterhalen van de juiste)

Keuze begrotingspost, uit een beperkte lijst met mogelijkheden. Daarnaast is een optie voor een post buiten deze lijst, en een invulveld om deze dan in te kunnen voeren. Hiervoor kan gebruik worden gemaakt van een overzicht van bekende begrotingsposten.
6. **Dient DP te behandelen?**

**Wie:** Medewerker dienstenplein.

**Wanneer:** Het binnenkomen van de aanvraag of na het bepalen van de consequenties van een wijziging.

**Wat:** Op basis van een aantal richtlijnen, wordt bepaald of de aanvraag door dienstenplein, danwel door het evenementenbureau dient te worden behandeld. Om dit te beoordelen, worden alle belangrijke gegevens voor deze keuze bekeken, zoals aantal deelnemers, aangevraagde faciliteiten, etc. De precieze beslisregels worden vastgelegd in document B. Richtlijnen.

7. **Bied maatwerk oplossing**

**Wie:** Medewerker dienstenplein.

**Wanneer:** Het binnenkomen van de aanvraag of na het bepalen van de consequenties van een wijziging.

**Wat:** Op basis van een aantal richtlijnen, wordt bepaald of de aanvraag door dienstenplein, danwel door het evenementenbureau dient te worden behandeld. Om dit te beoordelen, worden alle belangrijke gegevens voor deze keuze bekeken, zoals aantal deelnemers, aangevraagde faciliteiten, etc. De precieze beslisregels worden vastgelegd in document B. Richtlijnen.

8. **Bekijk mogelijkheden**

**Wie:** Medewerker dienstenplein.

**Wanneer:** Nadat bepaald is dat dienstenplein deze aanvraag behandelt.

**Wat:** Wederom op basis van een aantal richtlijnen, wordt bekeken welke ruimtes vrij zijn. In document B. Richtlijnen staat onder andere een aantal regels van welke ruimtes voor welk aantal deelnemers geschikt zijn, en welke beperkingen er zijn wat betreft faciliteiten in bepaalde ruimten. Voor deze specifieke aanvraag, wordt bekeken welke ruimtes opties zouden zijn, en welke van deze vrij zijn. In het geval van een meervoudige aanvraag, worden voor elke datum afzonderlijk de mogelijkheden bekeken. Daarnaast zijn ook de al gemaakte boekingen van belang om te bepalen welke ruimtes nog vrij zijn. Deze worden geregistreerd door document C. Reeds gemaakte boekingen.

9. **Mogelijkheid vrij?**

**Wie:** Medewerker dienstenplein.

**Wanneer:** Op het moment dat de mogelijkheden zijn bekeken.

**Wat:** Er wordt hier enkel een afweging gemaakt of een geschikte ruimte vrij is of niet. Vervolgens wordt de juiste vervolgactie genomen.

10. **Leg ruimte vast**

**Wie:** Medewerker dienstenplein.

**Wanneer:** Als duidelijk is dat een geschikte ruimte vrij is.

**Wat:** Er zijn één of meer geschikte ruimten gevonden die nog vrij zijn. Er wordt een keus gemaakt, en deze ruimte wordt vastgelegd in het systeem, als zijnde bezet.

11. **Schuiven mogelijk?**

**Wie:** Medewerker dienstenplein.
Wanneer: Als er geen geschikte ruimte vrij is gebleken voor de aanvraag.
Wat: Er wordt bekeken of er mogelijkheid is om te schuiven in eerdere boekingen en eventueel in deze aanvraag zelf, om toch een mogelijkheid te creëren. Dit uiteraard in samenspraak met de aanvrager van de te verschuiven boekingen.

12. **Stel melder op hoogte van de onmogelijkheid**

Wie: Medewerker dienstenplein.
Wanneer: Als blijkt dat er geen mogelijkheid is om te schuiven.
Wat: Er wordt via een mail terugkoppeling gegeven aan de aanvrager dat er geen mogelijkheid is om aan de aanvraag te voldoen.

13. **Leg ruimte vast**

Wie: Medewerker dienstenplein.
Wanneer: Als een andere boeking verschoven is en dus nu een mogelijkheid vrij is om deze aanvraag in te passen.
Wat: Er is een mogelijkheid vrijgemaakt in het systeem, deze ruimte wordt vastgelegd in het systeem, als zijnde bezet, voor deze aanvraag.

14. **Verwerk reservering**

Wie: Medewerker dienstenplein.
Wanneer: Op het moment dat een ruimte is vastgelegd.
Wat: De verdere gegevensverwerking van de reservering wordt hier geregeld. Dit houdt in het vullen van de verschillende systemen, zowel als het bepalen welke vervolgstappen nodig zijn.

15. **AV middelen in aanvraag/gewijzigd?**

Wie: Medewerker dienstenplein.
Wanneer: De aanvraag is vooralsnog vastgelegd in het systeem, en de verdere gegevens zijn ook verwerkt (of worden dat op dit moment).
Wat: Er wordt gecontroleerd of en welke (standaard) AV middelen zijn aangevraagd of in de wijziging zijn toegevoegd of verwijderd aan de aanvraag.

16. **ICT in aanvraag/gewijzigd?**

Wie: Medewerker dienstenplein.
Wanneer: De aanvraag is vooralsnog vastgelegd in het systeem, en de verdere gegevens zijn ook verwerkt (of worden dat op dit moment).
Wat: Er wordt gecontroleerd of en welke (standaard) ICT middelen zijn aangevraagd of in de wijziging zijn toegevoegd of verwijderd aan de aanvraag.

17. **Catering in aanvraag/gewijzigd?**

Wie: Medewerker dienstenplein.
Wanneer: De aanvraag is vooralsnog vastgelegd in het systeem, en de verdere gegevens zijn ook verwerkt (of worden dat op dit moment).
Wat: Er wordt gecontroleerd of en welke (standaard) catering producten zijn aangevraagd of in de wijziging zijn toegevoegd of verwijderd aan de aanvraag.

18. Faciliteiten in aanvraag/gewijzigd?

Wie: Medewerker dienstenplein.
Wanneer: De aanvraag is vooralsnog vastgelegd in het systeem, en de verdere gegevens zijn ook verwerkt (of worden dat op dit moment).
Wat: Er wordt gecontroleerd of er in het algemeen extra faciliteiten zijn aangevraagd. Als dit het geval is, dan is er in één of meer van de drie eerdere stappen een positief antwoord gebleken, en het proces met de backoffice in gang gezet voor die toepasselijke gevallen, daar loopt het proces dus nog steeds.

Als blijkt dat geen extra faciliteiten zijn aangevraagd, dan kan direct tot terugkoppeling worden overgegaan door het dienstenplein.

Deze stap is procestechnisch nodig om het geheel sluitend te laten zijn. Als er namelijk geen van de extra faciliteiten is gevraagd, wordt de reservering teruggekoppeld zonder de verdere back-office erbij te betrekken.

19. Aanvragen/wijzigen AV

Wie: Medewerker dienstenplein.
Wanneer: Als de aanvraag is vastgelegd en bepaald is dat er vervolgactie nodig is om audiovisuele middelen aan te vragen of een eerdere aanvraag te wijzigen op dit vlak.
Wat: Er wordt een document opgesteld van de vorm zoals beschreven in D.Verzoek. Hierin staat welke middelen aangevraagd worden, of in het geval van een wijziging, welke middelen niet meer aangevraagd worden, en op welke tijd dit gewenst is. Bij deze tijd wordt rekening gehouden met het ophalen van materiaal, voordat de bijeenkomst plaatsvindt. Vandaar dat dit materiaal een half uur eerder beschikbaar wordt gesteld, dan de bijeenkomst zelf wordt geboekt (indien mogelijk). Dit document wordt als verzoek doorgestuurd naar de beheerder van de audiovisuele middelen.

20. Aanvragen/wijzigen ICT

Wie: Medewerker dienstenplein.
Wanneer: Als de aanvraag is vastgelegd en bepaald is dat er vervolgactie nodig is om ICT middelen aan te vragen of een eerdere aanvraag te wijzigen op dit vlak.
Wat: Er wordt een document opgesteld van de vorm zoals beschreven in D.Verzoek. Hierin staat welke middelen aangevraagd worden, of in het geval van een wijziging, welke middelen niet meer aangevraagd worden, en op welke tijd dit gewenst is. Bij deze tijd wordt rekening gehouden met het ophalen van materiaal, voordat de bijeenkomst plaatsvindt. Vandaar dat dit materiaal een half uur eerder beschikbaar wordt gesteld, dan de bijeenkomst zelf wordt geboekt (indien mogelijk). Dit document wordt als verzoek doorgestuurd naar de beheerder van de ICT middelen.

21. Aanvragen/wijzigen catering

Wie: Medewerker dienstenplein.
Wanneer: Als de aanvraag is vastgelegd en bepaald is dat er vervolgactie nodig is om catering producten aan te vragen of een eerdere aanvraag te wijzigen op dit vlak.
Wat: Er wordt een document opgesteld van de vorm zoals beschreven in F.Verzoek. Hierin staat welke producten aangevraagd worden, of in het geval van een wijziging, ook welke producten niet meer aangevraagd worden, en op welke tijd, op welke locatie dit gewenst is. Dit document wordt als verzoek doorgestuurd naar de cateringdienst.

22. Reserveren materiaal indien mogelijk

Wie: Beheerder ICT of beheerder AV middelen.
Wanneer: Op het moment dat er een verzoek van het dienstenplein is ontvangen in de vorm D.Verzoek.
Wat: Als het mogelijk is om de verzochte middelen te reserveren op de gewenste tijd, dan wordt dit hier gedaan, en verwerkt in eigen administratie (E.Reservering).

23. Terugkoppelen mogelijkheid tot reserveren

Wie: Beheerder ICT of beheerder AV middelen.
Wanneer: De reservering van de gewenste middelen heeft plaats gevonden of er is geconstateerd dat dit niet mogelijk is.
Wat: Een document in de vorm van H.Geboekte middelen/service wordt opgesteld. Hierin staat welke middelen gereserveerd zijn, op welke tijd, en welke vergadering aanvraag dit betreft. Indien het niet mogelijk is deze middelen te reserveren, wordt dit hier aangegeven.

24. Plannen dienst indien mogelijk

Wie: Dienstplanner catering.
Wanneer: Op het moment dat er een verzoek van het dienstenplein is ontvangen in de vorm F.Verzoek.
Wat: Als het mogelijk is om de verzochte service te plannen op de gewenste tijd, dan wordt dit hier gedaan, en verwerkt in eigen administratie (G.Planning).

25. Terugkoppelen mogelijkheid tot service

Wie: Dienstplanner catering.
Wanneer: De planning van de gewenste service heeft plaats gevonden of er is geconstateerd dat dit niet mogelijk is.
Wat: Een document in de vorm van H.Geboekte middelen/service wordt opgesteld. Hierin staat welke producten leverd worden, op welke tijd, en welke vergadering aanvraag dit betreft. Indien het niet mogelijk is deze producten te leveren, wordt dit hier aangegeven.

26. Terugkoppelen gemaakte reservering

Wie: Medewerker dienstenplein.
Wanneer: Op het moment dat terugkoppeling is ontvangen van alle back-office partijen die betrokken zijn bij deze aanvraag; of na stap 18, indien er geen verdere faciliteiten bijgeboekt hoeven worden naast de ruimte.
Wat: Een document in de vorm van I.Aanvraag bevestiging wordt opgesteld. Hierin staat welke ruimte toegekend is, welke middelen en producten leverd worden, op welke tijd, en welke vergadering aanvraag dit betreft. Bovendien wordt gemeld dat de materialen zelf dienen te worden opgehaald bij het dienstenplein,
vanaf een half uur voor de bijeenkomst plaatsvindt. Indien het niet mogelijk is aan (een deel van) de aanvraag te voldoen, wordt dit hier aangegeven.

In het geval van een annulering, wordt hier teruggekoppeld dat de annulering is verwerkt.

27. Aanvragen wijziging/annulering

Wie: Aanvrager.
Wanneer: Als de aanvrager besluit iets aan zijn aanvraag te wijzigen, toe te voegen, of de aanvraag in haar geheel te annuleren.
Wat: De aanvrager vult een wijzigingsformulier in waarbij het meldingsnummer van de originele aanvraag wordt meegegeven. Daarnaast hoeven alleen die zaken ingevuld te worden, die moeten worden gewijzigd.

28. Bepaal consequenties wijziging

Wie: Medewerker dienstenplein.
Wanneer: Op het moment dat een aanvraag tot wijzigen of annuleren is binnengekomen.
Wat: Er wordt bekeken welke delen van de aanvraag daadwerkelijk gewijzigd zijn. Hierbij wordt document B.Richtlijnen gevolgd, om te bepalen wat de consequenties zijn, en of de gewijzigde aanvraag door dienstenplein, danwel evenementenbureau behandeld dient te worden. Vervolgens kan het normale traject gevolgd worden voor de behandeling van de aanvraag, waar die delen die ongewijzigd blijven, kunnen worden overgeslagen.

29. Leveren materiaal

Wie: Beheerder ICT of beheerder AV middelen.
Wanneer: Als de tijd aangebroken is waarop de reservering is geplaatst.
Wat: De beheerder ziet in de administratie dat bepaalde middelen gereserveerd zijn op een tijdstip, en zorgt ervoor dat ze op dat moment beschikbaar worden gesteld bij het dienstenplein. Nu is het aan de aanvrager om de materialen op te halen.

30. Leveren diensten

Wie: Medewerker catering.
Wanneer: Als de tijd aangebroken is waarop in de planning staat dat de service geleverd moet worden.
Wat: De medewerkers van de catering worden aangestuurd op basis van de planning. Op het moment dat de datum en het tijdstip aangebroken zijn, waarop de service is geboekt, worden de juiste producten naar de vergaderruimte gebracht.
Appendix F: Evaluation form (in Dutch)

Voor je ligt het evaluatie formulier waarmee wij de afgelopen workshops willen evalueren. Doel van de workshops was het opleveren van een proces model voor het plannen van bijeenkomsten. Middels deze evaluatie willen we de ervaringen van de deelnemers in kaart brengen om verbeterpunten in de methode boven water te halen.

De evaluatie bestaat uit 2 delen. We beginnen met een aantal algemene vragen over de gekozen aanpak en vervolgens behandelen we het prototype dat is gebruikt tijdens de laatste twee workshops. In totaal bestaat deze evaluatie uit 11 vragen, die in 10-15 minuten beantwoord kunnen worden.

Alvast bedankt voor de gedane moeite! Jullie bijdrage zorgt er voor dat ik mijn afstudeeropdracht kan afronden.

Algemene evaluatie vragen
Voor dit project hebben we gekozen voor een project aanpak die is gebaseerd op de Product Based Workflow Design (PBWD) methode. Bij deze ontwerp methode staat het product centraal (i.p.v. het proces). Verder is er gekozen voor een aanpak waarin workshops centraal staan.

Een dergelijke combinatie moet zorgen voor een goede interactie tussen de gebruikers en de proces ontwerpers. Over het algemeen resulteert dit in een kwalitatief goed proces ontwerp waar beide partijen achter staan. Wij zijn dan ook benieuwd naar hoe jullie deze aanpak hebben ervaren. We hopen dit helder te krijgen middels de vragen in deze sectie.

1. Hoe heb je de aanpak d.m.v. workshops ervaren (en dus de directe betrokkenheid in het project)?

2. Hoe heb je de duur van de workshops ervaren?

3. Hoe heb je de keuze om te starten met een Product Data Model (PDM) ervaren?

4. Wat vind je van de samenstelling van de workshopdeelnemers?

Het gebruikte prototype
Als proces ontwerpers hadden we jullie input hard nodig. Gedurende de eerste workshops hebben we dit gedaan door gezamenlijk het PDM op te stellen. Om vervolgens de afhankelijkheden (beter) inzichtelijk te krijgen hebben we gebruik gemaakt van een prototype. Dit prototype stelde ons in staat om jullie voorkeuren/wensen aan te hebben. Verder hebben we hierdoor geleerd hoe jullie omgaan met uitzonderingen en wanneer deze op kunnen treden. Voor ons was het gebruik van het prototype dan ook zeker zinvol. We zijn dan ook erg benieuwd hoe jullie het gebruik van een prototype hebben ervaren en of een eventuele andere indelingen het prototype begrijpelijk zou hebben gemaakt.
5. Het prototype is nooit bedoeld als een definitief systeem dat binnen het proces gebruikt zal gaan worden. Het doel was de afhankelijkheden en uitzonderingen duidelijk te krijgen. Was het doel van het prototype voldoende duidelijk?

6. In het prototype is er bewust voor gekozen alle mogelijk invulbare velden, invulbaar te maken. Is het wenselijk hier vooraf een volgorde van invullen in te bepalen?

7. We hebben gekozen voor korte namen in het prototype. Was de naamgeving van de data elementen voldoende duidelijk?

8. MogelijkePosten wordt (in het definitieve systeem) automatisch opgehaald. Dit betekent dat dit veld door de aanvrager niet ingevuld wordt. Toch was dit data element in het prototype opgenomen. Is het prototype beter te begrijpen als dergelijke velden weg worden gelaten?

9. Wat vond je van het tempo waarmee aanvragenscenario’s tijdens de workshop werden doorlopen?

10. Een van de mogelijke beperkingen van het prototype was de gekozen indeling. We hebben een viertal alternatieven ontworpen en zijn benieuwd wat jullie van deze alternatieven vinden. Welke zijn een verbetering ten opzichte van het huidige prototype en welke niet?
In dit prototype zijn de lijsten achterwege gelaten en is de melder al ingevuld op basis van zijn of haar inlog gegevens. Hierdoor staan velden die de aanvrager mogelijk moet invullen links.

Wel / geen verbetering

Motivatie:

**VERSIE 2**

In deze versie worden de lijsten automatisch als tekst klaar gezet door het systeem. Aangezien er in dit voorbeeld geen directeur of GS aanwezig is, staat een luxe lunch niet in de lijst. Rechts moet echter nog wel de gekozen lunch worden ingevoerd. Alle data elementen van het PDM zijn in deze versie aanwezig.

Wel / geen verbetering

Motivatie:
In deze versie zijn de lijsten verwerkt in een combo-box of selectiebox (kijk naar benodigde ICT en Begrotingspost). De gebruiker klikt aan wat hij nodig heeft. Standaard is er niets geselecteerd. Aangezien de Catering wensen (of mogelijkheden) afhankelijk zijn van de aanwezigheid van een directeur en/of GSer kan de aanvrager pas een catering keuze maken nadat de deze vragen beantwoord zijn. Een dergelijk prototype lijkt meer op een aanvraagformulier. Het proces kan nog steeds worden nagespeeld, maar de keuze vrijheid is enigszins beperkt (aan de invoerkaart).

Wel / geen verbetering

Motivatie:

**VERSIE 4**

Deze versie bestaat uit verschillende formulieren. De gebruiker wordt stap voor stap door verschillende formulieren geleid. Deze versie toont het minste data elementen. Dit komt het overzicht ten goede, echter uitzonderingen en/of fouten kunnen minder goed worden opgevangen. Dit kan een groot nadeel zijn om de juiste afhankelijkheden boven

Wel / geen verbetering

Motivatie:

11. Tot slot zijn wij benieuwd welk cijfer zou je de laatste 2 workshops zou willen geven (waarin we het PDM vertaald naar een proces ontwerp)?

Nogmaals bedankt voor je bijdrage. Jullie maken het mogelijk dat ik mijn afstudeertraject kan afronden!

Indien je nog vragen en/of opmerkingen hebt kun je die hieronder kwijt.