MASTER

Comparing BPM approaches in the healthcare domain
case handling vs. model driven engineering (BPMone vs. Mendix)

Muijres, G.J.M.

Award date:
2011

Eindhoven, August 2011

Comparing BPM approaches in the healthcare domain: Case handling vs. Model driven engineering (BPMone vs. Mendix)

by
G.J.M. Muijres

BSc Industrial Engineering and Management Science
Student identity number 0555168

in partial fulfilment of the requirements for the degree of

Master of Science
in Innovation Management

Supervisors:
dr. P.M.E. Van Gorp, TU/e, IS
dr.ir. H.A. Reijers, TU/e, IS

TUE. School of Industrial Engineering.
Series Master Theses Innovation Management

Subject headings: healthcare, data analysis, workflow, case handling, model driven engineering, technology evaluation

Abstract

In a world of continuously changing business environments, organizations are forced to deal with, and adjust to the demands for such level of change. To handle this level of change, many companies use different approaches and tools which include modeling and measuring business processes. Business Process Modeling provides support on business operations to improve business efficiency and to find new opportunities for the company. Not all organizations have recognized the added value of this approach. Especially healthcare organizations do not fully use the benefits of IT support. This thesis presents the pros and cons of two promising BPM approaches in the healthcare domain, with the Eye Care Network in Rotterdam as case study. The Eye Care Network is a collaborating healthcare business network of different players within the ophthalmology domain. There is often a network manager within a healthcare business network that wants to standardize the way the different players work. This can improve efficiency within the network and it can improve communication between the different players. The Eye Care Network, as a network manager, designed different treatment plans, as desired processes for treating patients in the hospitals connected to the network. One important aspect of these treatment plans is dealing with exceptions. Flexibility is required with this, to ensure the network members can continue providing their services, even when they want or need to deviate from the standardized process. This thesis presents the further investigated business process that describes the different steps a patient will endure during the treatment plan for the eye disease glaucoma. This thesis discusses how well two promising BPM approaches meet with the requirements from the Eye Care Network, based on the construction and evaluation of two prototypes. To be able to objectively compare the two approaches, the comparison framework of the thesis consists of seven principles divided over four different levels; case study level, design level, language level, and tool level. Model Driven Engineering (MDE) and Case Handling are the two approaches used to deal with the required flexibility by design. BPMone from Pallas Athena represents a specialized case handling tool, and Mendix represents a MDE tool. After creating a first version of both applications, BPMone could be seen as the best option out of the two tools. Mendix's primary weakness is the lower level of abstraction, compared to BPMone. This results in the need of a higher specification effort, which leads to the risk of making bad design decisions. Decisions that are already taken implicitly in BPMone. However, this limitation can be overcome by changing the Mendix model with a number of novel patterns, based on the apparently stronger BPMone principles. Similar options are not seen for BPMone (nor is there a need for it). An improved version of the Mendix model resulted in a comparing situation where BPMone was no longer the clear winner. This thesis not only gives a description of the abstraction patterns it also shows how they could be used on other cases as well. The suitability is not limited to this specific healthcare setting, since the generalizability of the thesis conclusions are demonstrated by considering a case study from a completely different domain as well.

The goal of this thesis is to examine which BPM approach is best suiting the requirements given by the Eye Care Network, for their business process. Furthermore, the possibility to generalize these findings outside of this case study will be checked.

Table of content

List of figures and tables

1. Introduction

Organizations in many domains are always looking for ways to improve the efficiency of their business processes in often rapidly and continuously changing environments. Business processes can be described as a set of activities with a common goal (Salimifard & Wright, 2001). All sorts of organizations find it difficult to deal with the demand for the continuous level of change. An industry that responds to this organizational issue consists of IT service providing companies. A main discipline in dealing with this matter is Business Process Modeling (BPM). The use of IT support for business processes has already been recognized as a tool for staying competitive in a market (Weber & Reichert, 2010). BPM tools provide the organization with valuable insight into their processes. It gives the opportunity to make well-structured decisions and, most importantly, it creates options to make the business processes more efficient (Aalst, Hofstede, & Weske, 2003).

Chapter 1.1 describes BPM in further detail. This is followed by chapter 1.2 where the different goals and uses of BPM are discussed. Then a number of different BPM approaches are mentioned in chapter 1.3, alongside with the chosen tools for each approach. Chapter 1.4 describes the environment in which this research is executed, and it explains the research question of this study.

The goal of this thesis is to examine which BPM approach is best suiting the requirements given by the Eye Care Network, for their business process. Furthermore, the possibility to generalize these findings outside of this case study will be checked.

Finally chapter 1.5 is a delineation of the different steps in this thesis.

1.1 What is BPM

Business process modeling is the systematic, structured method of representing enterprise processes in a model. This is done to analyze, improve, control, and manage the processes. There are a lot of definitions for a business process in the literature. One definition is given in the previous chapter, another description of a business process is: “a set of logically related tasks performed to achieve a defined business outcome” (Davenport & Short, 1990, page 12). Business processes have two important characteristics: (1) a process has customers – internal or external customers, and (2) a process crosses organizational boundaries – it can be across or between business units. According to Davenport & Short (1990), a process model shows which steps are required to execute the business process, in what order these steps need to be taken, and who needs to take these steps. More recently, process modeling approaches have emerged to paradigms where the order of execution is not specified explicitly. An important example of this kind of approaches is case handling, which will be discussed in more depth in chapter 1.3.
Business process modeling is used to trace and solve problems or to find opportunities to improve the business processes. There are actually two different kinds of models which can be used; simulation models and enactment models. To use a simulation model, detailed probabilities and time-related annotations are needed. For the implementation of the business processes (for workflow execution purposes), an enactment model can be used, for which operational details, like data sources and user interface interactions are needed. A definition of enactment modeling is: “*the use of software to support the execution of operational processes”* (Aalst, Hofstede, & Weske, 2003). The focus of this thesis will be on enactment models.

The field of Business Process Modeling exists for quite some time now, but the level of interest is still high. The interest in using information technology for BPM has grown not only in the business management community, but also in the computer science community (Badica, Badica, & Litoiu, 2003). These both communities have completely different backgrounds, but as will become clear, they are moving closer together. Since businesses are complex objects, precise models are needed for describing, analyzing and/or enacting its processes. Because of this need for carefully developing models, many different notations have been proposed for business process modeling activities. According to Badica et al. (2003), these notations can be broadly classified in two groups. First, the high-level visual notations which are more intuitive – because of the visual readability – and are mainly used by the business process management community. Second, the low-level foundational notations which are mainly used by the computer science community. The focus in this thesis will be on visual notations because of the readability advantage they have over non-visual notations. Visual notations make it possible for organizations to optimize their inherent business processes and to communicate them to partners to simplify business to business transactions (Vasko & Dustdar, 2006). A first observation of the converging action of the two different communities, can be linked to this classification in notations. The Model Driven Engineering (further explained in chapter 1.3) branch is part of the software engineering or computer science community, but uses increasingly high-level modeling notations (Hailpern & Tarr, 2006), just like the business process management community.

The second observation of the converging communities of BPM and software engineering is based on flexibility. Supporting the different communities are different process support paradigms like workflow management and case handling. Workflow management systems are most useful for business processes which are well structured and have a high degree of repetition (Weber & Reichert, 2010). Players in the healthcare domain often do not recognize their processes as having a high degree of repetition. This lack of flexibility is more dealt with in the second paradigm; case handling systems. Case handling is suited for more flexible process execution by avoiding restrictions which workflow management systems do have (Aalst, Weske, & Grünbauer, Case Handling: a new paradigm for business process support, 2005). However case handling systems are not perfect either. They are for example not suitable for fully automated business processes.
At the extreme end of the flexibility spectrum, you have the option to implement a new system from scratch, using a general purpose programming language. Writing everything from scratch is counter-productive. Therefore, in model driven engineering one is complementing general purpose programming languages with Domain-Specific Languages (DSLs) to improve development efficiency (France & Rumpe, 2007). Besides the rather big BPM field, the software engineering field is also a big research community, so a combination of both fields could lead to an optimal solution. Therefore, traditional engineering techniques and paradigms (like object-oriented engineering) should be combined with engineering principles to improve the support for business processes (like case handling) (Weber & Reichert, 2010). That is why in this thesis the generic object oriented software engineering approach was compared with the specialized case handling approach. To find out which approach gives the best results, to possibly boost the demand for IT support in healthcare as well.

1.2 Goals/uses of BPM

Business process models can be used for many different approaches, for example to support an organizational analysis, to derive requirements and specifications for building an information system, or to support automated execution of business processes (Paul, Giaglis, & Hlupic, 1999). Dependent on the requirements of the method of choice, some modeling techniques are more suitable than others. Table 1 shows five main business process modeling goals (based on Curtis, Kellner, & Over, 1992) together with their associated BPM requirements. Since these are main BPM goals, the goals are for simulation models as well as for enactment models. Since some overlap exists and ideally the created models serve both purposes, all goals are explained briefly. The first two goals are mainly focused on using BPM as part of organizational change management. So these are more a reflection

Business Process Modeling Goals	Business Process Modeling Requirements
- Facilitate human understanding and communication	comprehensibility, communicability, completeness
- Support process improvement	component identification, reusability, measurability, comparability, change impact assessment, decision support, evolution support
- Support process management	reasoning support, forecasting support, monitoring and coordination support
- Automated guidance in performing process	integration with development environments, documentability, reusability
- Automated execution support	cooperative work support, automated performance measurement support, process integrity check support

Table 1: Business Process Modeling Goals and Requirements from (Paul, Giaglis, & Hlupic, 1999)

of the goals of simulation models. The last goal is mainly aimed at enactment models. The first goal shown in Table 1 is to facilitate human understanding and communication. In this approach, the models are mainly used to support the communication between different stakeholders. To ensure all players understand the models, *comprehensibility* and *communicability* are part of the requirements. The models are used to visualize the processes. To guarantee correct communication between the different players, model *completeness* of the models is another requirement of this approach.

Using business processes as support for process improvement is the second possible goal. In this approach, the models are used to define and analyze a process. By means of this analysis, possible points of improvement can be found, for example, flow times or bottlenecks in the process. To make sure the complete process is modeled, *component identification* is a requirement. To be able to do calculations on the analysis, *measurability* and *comparability* are also important. *Decision support* and *evolution support* are needed to implement possible improvements, as well as figuring out how the process will react on these improvements. Therefore, *change impact assessment* is also one of the requirements. *Reusability* allows reusing parts of the models in the improved situation. Consequently, it is not necessary to start all over again.

The next goal is supporting process management. In this approach, the process models can be used as a reference point for management in planning, monitoring and coordination of the business process. With the requirement *reasoning support*, the models act as foundation for grounded decisions by the management. Referring to the requirements *forecasting support* and *monitoring and coordination support*, the models can be used to compare the current with the future situation. Automated performance process guidance is the fourth goal mentioned. In this approach, the model is used as assistance for a process, without actually implementing the model. Since it is running alongside of the actual process, *integration with development environments* is one of the requirements. Since the model is capturing and reusing the business process know-how, *reusability* and *documentability* are also requirements of this approach.

The final goal is automated execution support, which indicates that the models are implemented with support of an information system to control behavior in an automated environment. An example is a workflow management system. Since the environment is automated, the results can also be measured automatically by means of *automated performance measurement support*. Using the technology to support people in their work is described as *cooperative work support*. To make sure all files do what they are intended to do, a *process integrity check* can be performed.

1.3 Different approaches

With the increasing number of purposes for using business process models, the popularity and, consequently, the number and variety of users has increased as well. There are a number of different modeling approaches to handle the different modeling goals. Discussing all of these approaches would be outside of the scope of this research, therefore only the most important approaches are discussed in section 1.3.1.

1.3.1 Business process management

The first field of work which uses business process modeling is the business process management community. Over the years, many information systems have been built to support business processes. This continuously improving field of research is referred to as workflow management. Even though workflow management has been used for a number of decades now, the use of workflow management systems is often limited. Processes in which workflow management systems are used are mostly very simple. Workflow management systems are either not applicable for complex processes or they require considerable modeling and implementation efforts (Aalst & Berens, 2001).

1.3.1.1 Case handling

The goal of the case handling paradigm is to overcome some of the limitations of former existing workflow management systems (Reijers, Rigter, & van der Aalst, 2003). These improvements are generally visible in two principles; context tunneling and handling exceptions. Users of earlier workflow management systems might encounter context tunneling by only being able to see the work that they should do. It is not possible to access previously executed tasks or the yet to be performed tasks. The only visible data is data that is directly linked to the current activity, because most of these approaches are process driven. Case handling is data driven and claims to deal with this issue by creating awareness to the user. By showing the entire process at all times, (Vanderfeesten, Reijers, & van der Aalst, 2009) there is no longer a situation of context tunneling. The second principle – handling exceptions – is mainly a flexibility issue. Traditional workflow management systems are not good at dealing with deviations from the standardized routing of work. Case handling deals with this limitation by being data driven instead of process driven and by giving the user the opportunity to skip or redo a task (Aalst & Berens, 2001). In this way, the user can deviate from the standardized path. By creating the opportunity to choose the next task to perform, the case handling is more suitable for flexible process execution. Since it is avoiding restrictions which workflow management systems do have, by using predefined process control structures to determine what should be done (Aalst, Weske, & Grünbauer, 2005). Within the case handling paradigm, the focus is on what can be done to achieve a certain goal, instead of what should be done.

BPMone is a case handling tool from Pallas Athena, which is divided into two parts; BPMone Design and BPMone Control. BPMone Design is similar to Pallas Athena's prior product Protos (Pallas-Athena, 2011), whereas BPMone Control is comparable to the prior product FLOWer. BPMone is used as a workflow management system, but the BPMone Control part specifically supports the case handling paradigm. A paradigm that is used for supporting flexible and knowledge intensive business processes (Aalst, Weske, & Grünbauer, 2005). More about BPMone, and why it is chosen, in section 1.3.1.3.

1.3.1.2 Case handling RCM

To have a visual overview of the concepts explained in section 1.3.1.1, a Reference Concept Map (Rodríguez-Priego, García-Izquierdo, & Rubio, 2010) is used. This technique is based on the well-known Concept Map approach and shows the big picture of the main concepts and their relatedness via academic citations. The explanation of case handling mentioned in the previous section is made visual in Figure 1. The numbers in the figure represent the references, where 1 represents (Pallas-Athena, 2011), 2 represents (Aalst, Weske, & Grünbauer, 2005), 3 represents (Reijers, Rigter, & van der Aalst, 2003), 4 represents (Aalst & Berens, 2001), 5 represents (Vanderfeesten, Reijers, & van der Aalst, 2009), 6 represents (Vanderfeesten, Reijers, & van der Aalst, 2006), 7 represents (Aalst, Hofstede, & Weske, 2003), 8 represents (Aalst & Hee, 2002), 9 represents (Mutschler, Weber, & Reichert, 2008), and 10 represents (Guenther & Aalst, 2005).

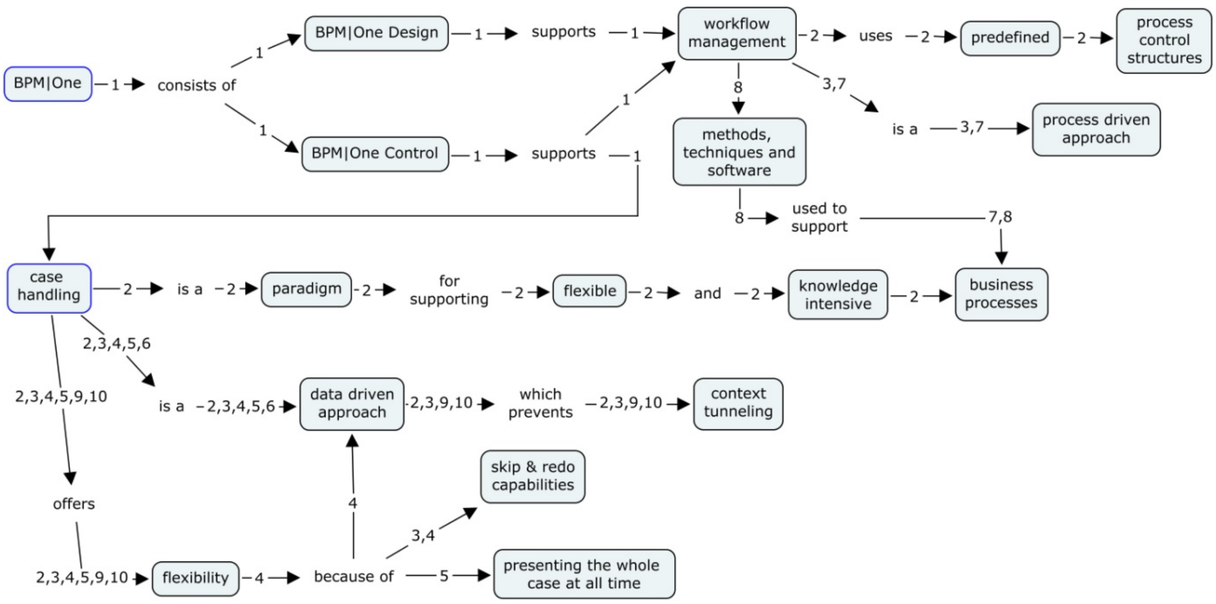


Figure 1: Reference Concept Map for Case Handling

1.3.1.3 BPMone for case handling

The tool chosen to suit the case handling paradigm is BPMone from Pallas Athena. This chapter describes the tool, some of its competitors and why it is chosen over other tools. "BPMone will discover, design, automate, and execute all your processes" (Pallas-Athena, 2011).

What is BPMone

As previously mentioned, BPMone is a software solution that is divided in two parts. BPMone Design is a modeling environment where you can model your processes. Once you have modeled your processes in BPMone Design, BPMone offers a simple way of seeing whether the processes are actually implemented in the way they were intended, by converting the model to a workflow model that can be accessed in BPMone Control. BPMone Control makes it possible to keep everything organized; a different 'worklist' (a sort of to-do list) can be created for each user, the level of access for each user can be defined, the specific forms the user will see can be adjusted and placed into the desired sequence, skip/redo options can be linked to the specific user who can apply them, and much more.

In 2010, Garnet made a Magic Quadrant which shows the top 25 vendors for Business Process Management Suites (BPMS) (Sinur & Hill, 2010). This list of vendors is an update for the research Garnet performed in 2009. The full Magic Quadrant can be seen in Appendix Figure 1. One of the vendors that were added in 2010 is Pallas Athena. The quadrant is divided in four sectors; challengers, leaders, niche players and visionaries. BPMone is situated in the sectors *niche players* which are classified as players that are generally new to BPMS market or new to Gartner's Magic Quadrant. They are players that have achieved significant-enough market awareness to be in the top 25 vendors. Pallas Athena is described by Gartner as a European vendor that has been growing quietly but largely, in the Netherlands. "It has the best technology and visualization we've seen for automated process discovery, and also handles the case management process style exceptionally well" (Sinur & Hill, 2010). Strengths and weaknesses described by Gartner can be seen in Appendix Table 1, alongside the strengths and weaknesses of a number of competitors. More features on BPMone can be seen in the description of the BPMone model in chapter 4, as well as in the comparison framework.

What are the competitors

For creating the Magic Quadrant, Gartner has analysed the BPMS market using multiple criteria and different weightings. The full Quadrant can be seen in Appendix Figure 1. As described in the previous chapter, Pallas Athena is one of the niche players, with their tool BPMone. The following three tools be can be considered as the closest competitors (based on the Quadrant); IBN FileNet, HandySoft and PNMsoft. To compare these tools with BPMone, Appendix Table 1 shows the advantages and disadvantages for each suite.

Why BPMone

The reason that Gartner identified Pallas Athena as one of the top 25 vendors in the BPM community is a good reason to choose BPMone. Since the focus will be on flexible approaches, especially the addition that it fits the case management principle exceptionally well is worth mentioning. But the three competitors mentioned in Appendix Table 1 are also listed in the top 25. FileNet is part of the big IBM network, but as mentioned in the table, it is not as intuitive in its ease of development and execution. The disadvantage both Sequence and BizFlow is the low brand recognition, which makes these two tools not suitable for this thesis either. In comparison, Pallas Athena is known for its BPM product and especially for its case handling tool. Multiple articles are written about case handling and Pallas Athena by professors from the Eindhoven University of Technical together with Pallas Athena (Aalst & Berens, 2001; Aalst, Weske, & Grünbauer, 2005), and without collaboration from Pallas Athena (Aalst, Stoffele, & Wamelink, 2003; Reijers, Rigter, & van der Aalst, 2003; Vanderfeesten, Reijers, & van der Aalst, 2006; Vanderfeesten, Reijers, & van der Aalst, 2009). This makes BPMone a suitable tool for this thesis. Since there is a good collaboration between the Eindhoven University of Technical and Pallas Athena, the required unique development skills as mentioned in Appendix Table 1 can be dealt with. The remark that there are few installations beyond the Benelux region is also not relevant as a disadvantage in this study.

1.3.2 Software engineering

The second field of work in which process modeling is an important aspect, is the field of computer science, more specifically; software engineering. Software engineering is interested in the business process field because of the resemblance with software processes. A software process can be seen as a business process with the goal of developing a program according to a given set of requirements (Badica & Badica, 2003). The field of software engineering has a completely different background than the business process management field, but the fields are converging, as described in chapter 1.1.

1.3.2.1 Model Driven Engineering

Model driven engineering (MDE) is a software engineering approach that focuses on creating and utilizing domain models (Bézivin, 2006) instead of focusing on computing concepts. The goal is optimizing productivity by creating the opportunity to reuse standardized models, simplifying the design process by modeling with recurring design patters, and optimizing communication between the different users of the process by using standard languages (Bézivin, 2006). This standardization is also shown in one of the important aspects of MDE. MDE focuses on bridges between technological spaces, and on integrating bodies of knowledge from different research communities (Favre, 2004).

Model driven engineering (MDE) is a software development methodology based on domain models. A domain model can be seen as a conceptual model of a field of interest. MDE consists of three major concepts; model, metamodel and transformations (Favre & NGuyen, 2004). A model can be seen as a set of statements about a system (Seidewitz, 2003). A metamodel can be described as the relationship between concepts in a given domain. It specifies key semantics and constraints associated with these concepts (Schmidt, 2006). Transformations are ways to use one or more source models in order to create one or more target models which follow a set of transformation rules conforming the metamodel (Sendall & Kozaczynski, 2003).

There are different organizations who claim to support MDE. The main focus will be on Model Driven Architecture (MDA) since this was one of the first approaches to standardize MDE, and to make it known to the public. The Object Management Group (OMG) launched MDA in 2000 as one example of the broader MDE vision (Bézivin, 2006). MDA is not an entirely new specification, but it uses existing OMG specifications like the Unified Modeling Language (UML), the Meta Object Facility (MOF), and the Common Warehouse Metamodel (CWM; Truyen, 2006).

For this thesis, Mendix was chosen as the tool to represent MDE. It consists of a domain model, different forms, secondary documents, and microflows (Mendix, 2010). The microflows use a graphical notation based on business process modeling notation (BPMN). BPMN is a standard developed by the Business Process Management Initiative, which has merged with the OMG (Business Process Management Initiative (BPMI), 2004). This OMG standard defines the notation, metamodel and format of BPMN. Mendix's domain models are quite similar to UML class diagrams (Meertens, Iacob, & Nieuwehuis, 2010). The entity that can be created in the domain model resembles a class, with its name, attributes, and associations. The added significant value is shown when Mendix automatically creates a data view form out of these created entities. Another feature of Mendix is the abstraction opportunity. A group of elements can be abstracted, whereas Mendix can treat the group of elements as one element. After creating the different forms the user can fill out, custom made 'microflows' can be added to navigate through the different created forms. More operations that add value become clear in the explanation of the Mendix model in chapter 3. Mendix is not a typical MDE tool, since it only provides one out of three major concepts; the model. However, since it is a popular tool with well-known customers (e.g., ABN-Amro, Achmea, Mammoet, Sandd, Sanoma, TNT, WE) and well-known partners (e.g., Accenture, Atos Origin, and Centric) Mendix was chosen. More about Mendix, and why it is chosen, in section 1.3.2.3.

1.3.2.2 Model driven engineering RCM

To have a visual overview of these concepts, a Reference Concept Map (Rodríguez-Priego, García-Izquierdo, & Rubio, 2010) is used again. The explanation of MDE mentioned in the previous section is made visual in Figure 2. The full Reference Concept Map can be seen in Appendix Figure 2. The numbers in the figure represent the references, where 1 represents (Favre & NGuyen, 2004), 2 represents (Seidewitz, 2003), 3 represents (Schmidt, 2006), and 4 represents (Peltier, Bézivin, & Guillaume, 2001).

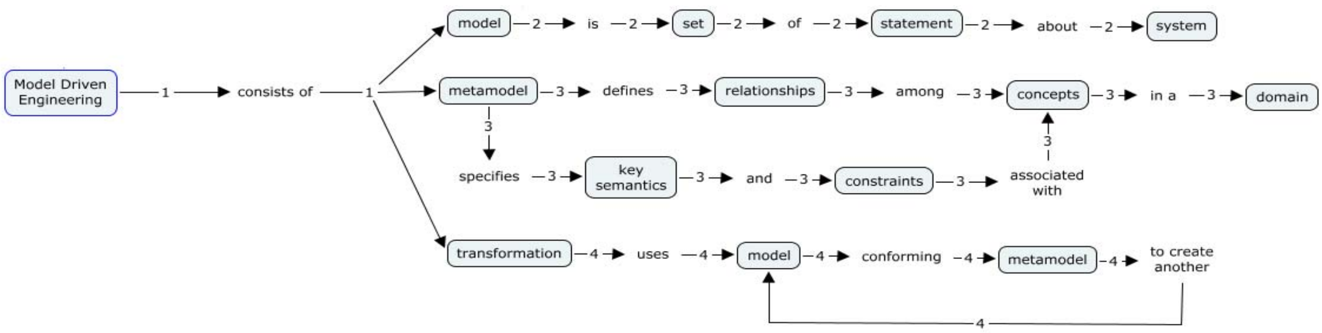


Figure 2: Reference Concept Map for Model Driven Engineering

1.3.2.3 Mendix for model driven engineering

The model driven engineering tool that was selected is Mendix. In the next three sections an explanation of Mendix will be given, some of its competitors will be described, and the choice for Mendix is explained.

What is Mendix

Gartner listed Mendix, the market leader in model-driven development for business solutions, as a 'Cool Vendor' in the category 'Cool vendors in application development' in 2009 (Norton, 2009). A 'Cool Vendor' is described as a company that offers technologies or solutions which comply with three characteristics. These characteristics are (Roos, 2009):

- **Innovative**: it enables users to do things which they could not do before;
- **Impactful**: it has, or will have, business impact (so it is not just technology for the sake of technology);
- **Intriguing**: it has caught Gartner's interest or curiosity within approximately the past six months.

Mendix is providing tools to quickly design, build, test, integrate, deploy, manage and optimize service-oriented business applications within any existing business and IT environment. Mendix claims that you can build applications 5 times faster, at about half of the cost of traditional development platforms, by using their tool.

What are the competitors

Obviously Mendix is not the only tool in its kind. Other companies have similar products available. To be able to compare a number of them, Appendix Table 2 shows the advantages and disadvantages of Mendix and three of its competitors; Skelta, Outsystems, and BizAgi. More features on Mendix can be seen in the Mendix model description in chapter 4, and in the comparison framework.

Why Mendix

As described in section 1.3.2.1, Mendix is not a typical MDE tool, since it misses two out of three MDE principles (transformations and meta-models), but there are a number of reason to still choose for Mendix. The first reason to choose for Mendix is because its popularity and it combines user friendliness – by means of intuitive forms, and the usage of your favorite internet browser for the web application – with industry standards like BPMN. Besides the intuitive way of working, the user experience is also rather high because of all the downloadable custom widgets that are available, and the big online community that can be accessed for help. Besides that, Mendix originated in the Netherlands and has connections with the Eindhoven University of Technology, which makes it approachable when we would get stuck modeling. This in contrast to the other three tools which are all foreign companies. Furthermore the recognition of Gartner, the SAP certification, the well-known customers, and the well-known partners give the confidence in Mendix as the correct choice.

1.4 Research environment

The environment in which this study takes place is a healthcare environment. Specifically, the problem owner of this research was the Eye Care Network. Usually a healthcare environment is a rapidly changing environment where flexibility is one of the requirements. Furthermore, organizations within this domain have not fully recognized the potential of using IT support for business processes, so there is a lot of potential for improvements. The Eye Care Network, which has its origin in Rotterdam, is a collaborating network of different players within the ophthalmology domain. It is an initiative of the Eye-Hospital in Rotterdam, as a response to the increasing demand for ophthalmologists, higher quality demands from the patients, and strong competition. To tackle the long waiting lists for patients and the high working pressure for professionals, the idea arose to collaborate more intensively with the different players within ophthalmology (Het Oogzorgnetwerk, 2010). These different players include the general practitioner, optician shops, optometrists and ophthalmologists, working in- and outside of the hospital. All these different players are seen as partners. To ensure that different players can trust each other's level of expertise, random checks are performed to verify if correct decisions are made. The original network is active in Rotterdam, but the Eye Care Network is growing throughout the Netherlands. By keeping the patient as a focus point, the network aims to improve the quality of the medical care within the network. All sorts of protocols are available for creating different treatment paths for a patient. These protocols, and other background information is stored digitally (a website called "Oogzorgnet") and can be accessed by the partners within the Eye Care Network. These protocols can be seen as a standardized process description. The idea is for all the players to use these standardized processes, but the network members who are working with these processes sometimes want to or need to deviate from this standardized path. That is why flexibility is a key requirement in this context.

When a new hospital is linked to the network, their processes are evaluated. Process models are made for the different treatment paths and possible improvements are discussed. Different hospitals are situated in different points of time based on the process of connecting to the network. To discover which hospital was most suiting for the case study, multiple interviews in different hospitals connected to the network were performed. After interviewing staff at the Westfriesgasthuis in Hoorn, it became clear that it concerns a hospital where improvements are being realized at the moment. An application, within the hospitals EPR, custom built for the ophthalmology department was used to coordinate all steps a patient goes through. At the other side of the digitalization spectrum lies the Reinier de Graaf hospital in Delft. Within this hospital most processes are not yet performed digitally. The processes are evaluated by the Eye Care Network, but no improvements have been realized yet. Since most possible improvements can be realized here, the hospital in Delft acts as input for the case study in this study.

The goal of the Eye Care Network of using the application is threefold. First the application should support automated execution, by using the application to support the hospital staff in its work. In this way not all tasks require to be performed manually which can save both time and errors. Since automated execution support is a typical enactment goal (see chapter 1.2), this case is well suited for this thesis. Besides this goal, the Eye Care Network has some other goals which include facilitating communication between different players of the network, and possibly supporting process improvement. Since the doctors can decide to deviate from the standardized treatment path, not each patient will follow this exact standard path. With a process monitoring facility, calculations can be done on how many patients follow what specific path. When the BPM applications show that the standardized treatment path is not the path which is followed most of the times, this standard path can be adjusted.

The goal of this thesis is to examine which BPM approach is best suiting the requirements given by the Eye Care Network, for their business process. Furthermore, the possibility to generalize these findings outside of this case study will be checked.

1.5 Different steps in the study

The structure of the comparison between the two BPM approaches can be seen in Figure 3. The first step was constructing a BPMone application, influenced by standards and strengths of the case handling community. Besides this BPMone application, the first step also includes the construction of a Mendix application, influenced by the standards and strengths of the MDE community. The second step describes the comparison between the two applications, to find the best approach for this specific case, in this domain, with the required level of flexibility. Since BPMone appeared to be the better approach, an improved Mendix application is constructed in step three, based on the apparently stronger BPMone aspects. This improved application is based on reusable patterns, constructed in the design level. This improved Mendix application is then compared to the first version of the BPMone model.

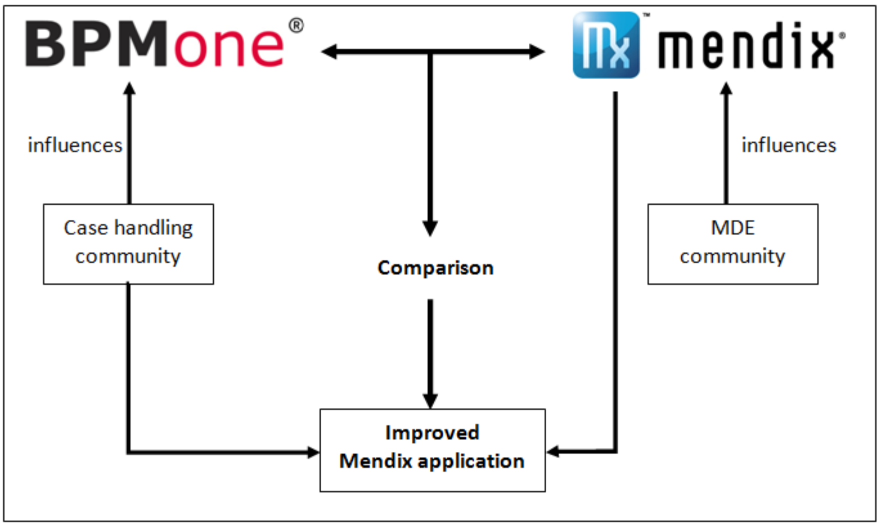


Figure 3: Structure of the comparison

Besides the specified goals mentioned in chapter 1.4, the Eye Care Network has provided some user requirements to which the model should comply. These requirements are discussed in chapter 2.1. For the case study, two possible processes were studied. These two processes are the treatment plans for the two most prevalent eye-diseases; cataract and glaucoma. After explaining these two processes in more detail in chapter 2.2, one process was chosen to implement an executable prototype in chapter 2.3. The comparison framework is formed in chapter 3. Chapter 4 describes the actual modeling steps taken in this research, combined with the model comparing, based on the comparison framework. The study concludes with a discussion in chapter 5.

2. Case study

This chapter describes the case study from the Eye Care Network. It starts with chapter 2.1 where the user requirements for the business process model, from the Network are summed up. Chapter 2.2 describes the two possible processes that can be modeled, and chapter 2.3 explains which of the two processes was chosen to model.

2.1 User requirements

The Eye Care Network is a company looking for improvements on their business processes. One of the uses of the process models will be to communicate with the different players within in the network. The usage of the models by the hospitals will be to support and/or improve their processes. The Network has some requirements for such a model, which are shown in Figure 4.

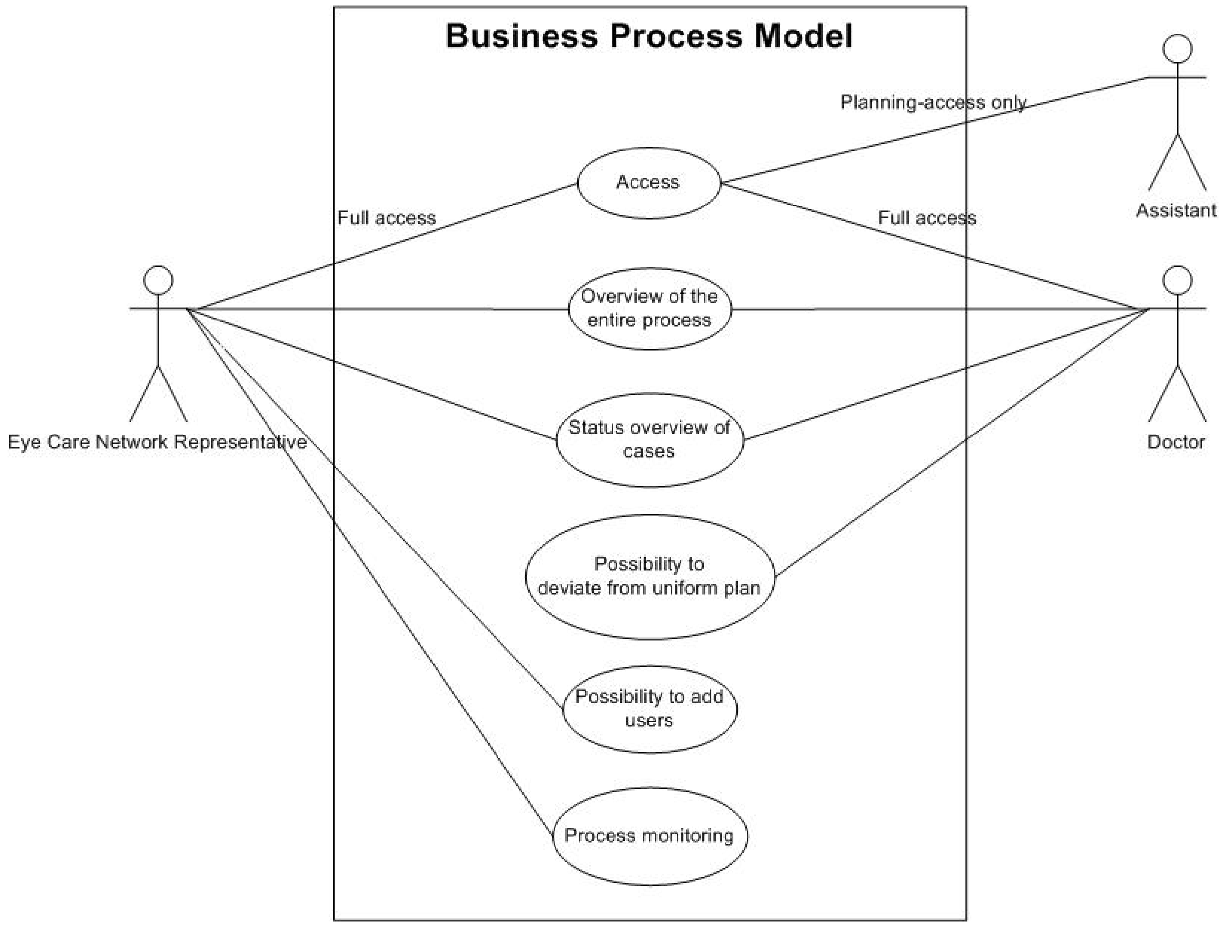


Figure 4: Use case diagram for requirements

In the diagram, a representative from the Eye Care Network is presented as one player. Within a partner hospital, two different users can be found; an assistant and a doctor. The assistant only has access to those parts of the model that include planning new appointments. The other actor – referred to as 'Doctor' - has access to the whole model. This role can be fulfilled by an ophthalmologist, but also a TOA or optometrist can have access to all the steps. This distinction in accessibility has to do with patient's privacy. The less people that have access to the patient's information, the better. This role-based access control is proposed by (Motta & Furuie, 2003). Since the assistant does not need detailed medical information to make an appointment, this role is restricted from those details. The previously mentioned medical staff (everybody except the assistant) does need access to the entire process to make decisions. To verify how the process is followed in real life, the Eye Care Network also needs full access to the system as well. The Eye Care Network serves as a kind of administrator to the model, so they should also have the opportunity to add users that can use the model. The model shows the uniform plan that the different players should follow. When something unscheduled happens, the medical specialist should be able to respond to this. In order to respond to these ad hoc decisions, the specialist should have the possibility to deviate from the uniform plan. The Eye Care Network is aware of the fact that these deviations take place, but they want to know how often this occurs and which different routes are taken. Therefore, process monitoring is an important requirement. With this aspect, the Network can check the different steps taken in a case. If there are major deviations from the uniform plan, it could possibly be improved. Another requirement is that the medical specialist should have a status overview of the different cases. This combined with an overview of the entire process gives them the opportunity to choose the correct case to handle. In this way, the user can see which cases are finished and how far the unfinished cases are along the process. Then, the user can choose which case should be performed next.

Besides these user requirements, there are also data requirements from the Eye Care Network. These requirements can be seen in a class diagram in Appendix Figure 3. Each class can be seen as a different step in the process. And the attributes of each class can be seen as the data objects that should be availabe to fill out in that particular step. The process that is described by these requirements is the treatment path of the disease glaucoma. The explanation of this process, and why it is chosen is explained in chapter 2.2.

2.2 Two possible processes

As suitable options for building a process model, two processes from the Eye Care Network are described in this chapter. The two processes under study are the treatment plans for cataract and glaucoma. Both processes will be described in the following sections (2.2.1 for cataract and 2.2.2 for glaucoma). Besides the explanation of the disease and a description of the treatment plan, a process model is included for both processes to make it visual. After the description of both processes, one of them is chosen in chapter 2.3 to implement an executable prototype. For the Eye Care Network it should be a process that is performed a lot in the joined hospitals. That is why these two processes are chosen, since cataract and glaucoma are the number on and number two leading causes for blindness worldwide (WHO, 2011). The treatment processes are being handled every day in each hospital. For the chosen BPM approaches it is important that the process requires a certain level of flexibility, since this is one of the main drivers of case handling.

2.2.1 Cataract

With cataract, a clouding of the lens in the eye makes you see blurry, due to an obstruction in the passage of light (WHO, 2011). Most of the cataract cases are age-related, but some people are born with it while others may develop cataract after other eye diseases. According to the World Health Organization, age related cataract is responsible for 48% of world blindness, which stands for roughly 18 million people. Cataract can be removed by surgery, but it remains the main cause of blindness worldwide, because of inadequate surgical services in many countries. With about 150.000 surgeries annually, cataract surgery is the most performed and most successful surgery carried out in the Netherlands (Oogartsen.nl, 2011). During the surgery, the clouded lens is removed and replaced by a clear artificial lens (Het Oogziekenhuis Rotterdam, 2009). During the procedure, a small incision is made in the cornea, through which the cloudy lens is removed and the new artificial lens is inserted. The entire cataract surgery takes about 25 minutes. Because the incision is only a few millimeters, there is usually no need for stitches which is beneficial for the healing time of the wound. When both eyes require surgery, these are performed separately with enough time in between to recover from the surgery.

The cataract treatment process a patient undergoes at the Reinier de Graaf Group in Delft was documented in 2010 (this is not the uniform situation, but the current situation). Based on this current situation modeling, some possible improvements are documented to have the process more in line with the uniform situation. This documentation is used to create the uniform process model shown in Appendix Figure 4.

The process starts with a patient who is referred to the hospital by a GP/optometrist/optician who suspects the patient of having cataract. The patient needs to call the hospital to make a first appointment. The receptionist enters the name and date of birth of the patient as well as the date of the appointment. The receptionist also checks whether the patient is referred to the correct place. If this is not the case, the patient is referred to the correct place and the case is closed. When the referral was correct, the patient is first examined by a TOA (Dutch abbreviation for Technical Ophthalmology Assistant). The TOA runs some tests and enters patient's eyesight, eye pressure and some possible notes. After that, the patient sees the ophthalmologist for a first consultation, where the doctor decides whether the patient has cataract or not and if the patient has other diseases besides cataract. If the patient does not have cataract or if the patient has other diseases, the process is ended. If the patient does have cataract, the ophthalmologist informs the patient about the options. Then, the patient needs to decide whether he/she wants the surgery to be performed. If not, a new consult is planned for over 6 to 12 months by the receptionist. When the patient agrees with the surgery, an appointment is planned to carry out an echo and a POS (Dutch abbreviation for Pre Surgery Screening). The echo is performed and the screening takes place at another department in the hospital; anesthetics. When these steps are completed, an appointment can be made for the actual surgery. The surgery takes place at the surgery department of the hospital. After surgery there is a check whether there were abnormalities during the surgery. If this is the case, the post-op takes place in the hospital and eyesight, eye pressure and some possible notes are entered. If there were no abnormalities during surgery, the post-op takes place over the phone. The patient is asked whether everything went well, and some notes are entered. After each post-op step, a check for abnormalities is performed again. If everything went according to plan, a final checkup appointment is made. If some deviations are noticed, a consultation with the ophthalmologist takes place first (before planning the final checkup). During the final checkup again the eyesight, eye pressure and some possible notes are recorded. After this final checkup the case is closed.

2.2.2 Glaucoma

Glaucoma can be seen as a group of eye diseases resulting in the slow dying of the eye nerve (Vision 2020, 2005). Approximately 1 to 2 percent of the western population has glaucoma (Korne, et al., 2009), around 2 to 3 percent of poor sight is caused by glaucoma and around 6.4 to 18 percent of blindness is caused by glaucoma (Vision 2020, 2005). Glaucoma cannot be fully cured. The treatment process is only aimed at slowing or stopping the worsening process (Korne et al., 2009; Vision 2020, 2005). As a result, people suffering from glaucoma are patients for life. The possible treatments include medicine to reduce the eye pressure, or laser or general surgery, again to reduce the eye pressure.

Glaucoma care requires many repeated examinations which can be performed by a variety of parties within ophthalmology (Korne, et al., 2009; Peeters, 2008). In 2005 there were an estimated 84.700 patients undergoing glaucoma treatment. It is estimated that there will be about 119.100 patients undergoing treatment in 2020 (Vision 2020, 2005).

The Eye Care Network tries to create a uniform protocol for glaucoma treatment as well. So their partner hospitals are able to optimize patient wellbeing and process quality. The glaucoma treatment process a patient undergoes at the Reinier de Graaf Group in Delft was also documented in 2010. This documentation is used to create the uniform process model shown in Appendix Figure 5.

The process starts again with a patient who is referred to the hospital by a GP/optometrist/optician who suspects the patient of having glaucoma. The patient needs to make a first appointment by calling the hospital. The name and date of birth of the patient as well as the date of the appointment are entered by the assistant. The assistant also checks whether the patient is referred to the correct place. If this is not the case the patient is referred to the correct place and the case is closed. On the first visit – when the referral was correct – the patient is examined by a TOA. The TOA runs some tests and records patient's eyesight, eye pressure and some possible notes. After that, the ophthalmologist decides whether the patient has glaucoma, ocular hypertension, or whether glaucoma runs in the patient's family. If this is not the case, the process is ended. The next step is for the ophthalmologist to decide on whether the glaucoma meets the referral criteria. If so, a regular consult with a TOA or optometrist is planned. If not, a consult with an ophthalmologist is planned. During this consult some tests are done where the eyesight, eye pressure and some possible notes are entered. Furthermore, the doctor decides whether the glaucoma is stable or not. If it is not considered to be stable, a new consult with an ophthalmologist is planned. If it is stable, a regular consult is planned. Before the regular consult, a GDX scan is performed and some notes are entered. After this step the consult takes place where the eyesight and eye pressure are measured and some possible notes are recorded/written down. The next step is to decide whether a HFA test (a vision test) is required. Once the test is performed – or not – it is checked whether there are some deviations. If there are no deviations, a next consult – for over 6 months – is planned with either an ophthalmologist or a TOA. If there are any deviations, a consult with an ophthalmologist is planned, where the usual tests are performed and entered. After this consult, an appointment is made for a next consult for over 6 months with either an ophthalmologist or a TOA. Since glaucoma cannot be cured, this process is repeated until the patient dies or chooses not to be treated anymore.

2.3 Which case is selected

Since both processes are processes that are performed a lot in all hospitals, this criteria is not sufficient to make a choice between the two cases. The flexibility criterion does make a difference. The cataract process is a rather straightforward process with not a lot of possible deviations from the uniform process plan. Not a lot of flexibility is required for this process, since most patients follow the exact same route. The glaucoma process however requires a lot more flexibility. First of all the process is repeated every six months, since the disease is not curable. This repetition itself does not necessarily require more flexibility, but after each run through the process, it can be decided to follow the next process under supervision of an ophthalmologist, or under supervision of a TOA.

Furthermore, there are several routes a patient can follow throughout the process which makes the process monitoring aspect more interesting. Because of the different routes that can be followed, more flexibility of the model is needed for the users to effectively use it. Since both the chosen modeling approaches are considered to be more flexible than 'regular' approaches, this is the most suitable option of the two. Therefore, the selected case for implementing an executable prototype is the glaucoma treatment process.

2.4 Case study characteristics

Flexibility plays an important role in the case study, as previously described. But to classify what kind of flexibility is required, four distinct types of process flexibility can be investigated. These four types of flexibility are (Schonenberg, Mans, Russell, Mulyar, & Aalst, 2008):

- *"Flexibility by design:* for handling anticipated changes in the operating environment, where supporting strategies can be defined at design-time.
- *Flexibility by deviation:* for handling occasional unforeseen behavior, where differences with the expected behavior are minimal.
- *Flexibility by underspecification:* for handling anticipated changes in the operating environment, where strategies cannot be defined at design-time, because the final strategy is not known in advance or is not generally applicable.
- *Flexibility by change:* either for handling occasional unforeseen behavior, where differences require process adaptations, or for handling permanent unforeseen behavior."

All the possible steps a patient can undergo during the glaucoma treatment are known, only the decisions the ophthalmologist makes, can vary per patient. So the sequence of steps, and the total number of performed steps per patient can vary. These deviations can be seen as anticipated changes, which can be dealt with supporting strategies during the design of the application. In other words, the required type of flexibility for this case is *flexibility by design*.

3. Comparison framework

To have a complete comparison between the two BPM approaches under investigation in this thesis, a new comparison framework is developed. First chapter 3.1 describes the four different levels that the comparison can include. Chapter 3.2 describes four existing frameworks, and how their principles fall in these four different levels. Finally chapter 3.3 explains the new framework based on these four existing ones.

3.1 Classifying the approaches on different levels

To provide a full comparison between the two approaches, the approaches are classified on various levels; the case study level, the design level, the language level, and the tool level (Van Gorp & Eshuis, 2010). The case study level describes the applications made for the specific case study of this thesis. The design level describes the decisions the developer of a certain application can make. The tool level describes the properties and limitations from the tool that the developer uses to construct the application. The language level can be seen as the underlying modeling language on which the underlying tool characteristics are based. This classification is required to compare the correct aspects in the correct context. For example, a limitation of BPMone, as a case handling based modeling program, can be the consequence of a design decision, instead of being the result of using a case handling based tool. This study will not present the complete taxonomy, but only those parts that are of interest to this case study.

3.2 Existing frameworks

Because of the widespread increase in modeling approaches, the number of users who are no modeling experts is growing as well (Becker, Rosemann, & von Uthmann, 2000). This increases the importance of the understandability of business process models. Based on this concept, multiple frameworks have been developed in literature, considering the four different levels. Based on four of these frameworks, one new framework was created to compare the two modeling approaches. These frameworks are from (Becker, Rosemann, & von Uthmann, 2000), (Green & Petre, 1996), (Mendling, Reijers, & Aalst, 2008), and (Bosilj-Vuksic, Ceric, & Hlupic, 2007). First these four frameworks are described, and a classification is made to decide what level is described by each relevant principle of the frameworks.

Becker et al. created the Guidelines of Modeling (GoM) framework which is built on six principles that are of importance for business process modeling. The six principles are: correctness, relevance, economic efficiency, clarity, comparability, and systematic design. The first three principles are a necessary precondition for the quality of a model. The last three are more optional (Becker, Rosemann, & von Uthmann, 2000) and are therefore not further elaborated upon. The *correctness* is based on using the correct syntactical rules and terminology. Since syntax rules are set in a modeling language, this principle belongs to the language level. The idea behind *relevance* is developing a relevant model. To measure this principle, the degree to which the application meets the user requirements is determined for each approach. When all requirements are met, the model can be seen as a relevant model for the selected case. Since it compares meeting the requirements from this specific case study, this principle belongs to the case study level. The *economic efficiency* is based on the effort it takes to create the models. This principle can be split in two; the number of elements needed to model the case, and the opportunity to re-use (parts of) the models. The reusability principle belongs to the design level. When a model is designed as generic as possible, the opportunity of reusability is greater than when the model is specifically designed for a certain case.

The number of elements needed to model the same case with the same characteristics depends on the language used. In this case study, the required skip/redo functionality is built-in BPMone because it is part of the case handling paradigm. In this way, this functionality requires fewer elements in comparison to Mendix, since the MDE paradigm does not support the skip/redo functionality, so it needs to be created manually.

The second framework is based on Bosilj-Vuksic, Ceric & Hlupic (2007), which uses Hlupic, Paul & Irani (1999) as input. It consists of five main groups of categories, and each group is further classified into subcategories. Only those categories are used that apply to this context, some subcategories have been moved to another category, and some additions are made in line with other subcategories. The framework with the categories and subcategories is shown in Table 2.

Coding aspect	**Programming flexibility**	**Built-in function**	**Support of programming concepts**	
	-provided -not provided	-provided -not provided	-provided -not provided	
User support	**Documentation and tutorial**	**Training course**	**Demo models, libraries**	**Online help**
	-provided -not provided	-provided -not provided	-provided -not provided	-provided -not provided
General features	**Experience and education required for software use**	**Ease of learning**	**User friendliness**	**Animation**
	-none -some -substantial	-easy -not easy	-high -medium -low	-possible -not possible
Verification support	**Logic checks**	**Error message**		
	-provided -not provided	-provided -not provided		
Explicitness of process model	**Display of the workflow path at design time**	**Display of the workflow path at run time**		
	-provided -not provided	-provided -not provided		

Table 2: Comparison framework

All three subcategories of the principle *coding aspect* clearly belong to the tool level. The same is true for the *user support* principle, as well as for the four subcategories from *general features* and the two subcategories from *verification support*. The two subcategories from *explicitness of process model* belong to the design level. The workflow path is, for example, automatically visible at run time in BPMone, but this can be recreated in Mendix, so it is not dependent on the tool used.

The third framework proposed a set of seven process modeling guidelines (7PMG)(Mendling, Reijers, & Aalst, 2008). These guidelines are built on strong research foundations and they are designed to be intuitively to practitioners. The seven guidelines are prioritized by industry experts, which leads to the following list.

1. Model as structured as possible
2. Decompose a model with more than 50 elements
3. Use as few elements in the model as possible
4. Use verb-object activity labels
5. Minimize the routing paths per element
6. Use one start and one end event
7. Avoid OR routing elements

How well you perform on the first guideline really depends on the tool used and the possible options. Therefore, this principle belongs to the tool level. The second principle clearly belongs to the design level, since it is just a decision the developer can make during the development phase. The third principle is in line with the economic efficiency principle from the GoM framework. The idea is similar, so this principle belongs to the language level. Principles four, five, six and seven are again decisions the developer can make during the development phase, which make them belong to the design level.

The final framework is the cognitive dimensions framework (Green & Petre, 1996). This framework is based on thirteen principles; abstraction gradient, closeness of mapping, consistency, diffuseness, error-proneness, hard mental operations, hidden dependencies, premature commitment, progressive evaluation, role-expressiveness, secondary notation, viscosity, and visibility.

Abstraction gradient describes the possibility of abstraction (a group of elements treated as one entity). As argued by Green & Petre (1996), languages can be described as abstraction-hating, abstraction-tolerant, or abstraction-hungry. Therefore, this principle obviously belongs to the language level. The *closeness of mapping* and *consistency* principle describe the ease of learning, which is comparable to the *general feature* principle from the second framework. The *diffuseness* principle describes how many symbols are needed to express a meaning. Just like the economic efficiency principle from the GoM framework, and the third principle from the 7PMG framework, this can be linked to the language level. The *error-proneness* principle explains how languages contain features that increases the chance of making a mistake, or how difficult it is to find a mistake once made. Consequently this principle belongs to the language level. *Hard mental operations, hidden dependencies, role-expressiveness, secondary notation, viscosity, and visibility* are all related to how clearly the user can read a model, how easy it is to understand it. According to Green & Petre (1996), some languages make this easier than other languages, which again places these principles in the language level. The *premature commitment* principle describes whether developers have to make decisions before they have all the required information. Problems can arise when the language notation contains many internal dependencies, so again this principle belongs to the language level. The final principle is *progressive evaluation*, which describes the opportunity of a tool to execute a partially-complete application to obtain feedback. Therefore this principle belongs to the tool level.

3.3 New combined framework

The new combined framework contains principles for each of the four different levels. Using this framework, a broad comparison (based on the highlights of the four other frameworks) can be made between the two BPM approaches from this study, as well as other approaches that can be compared in other studies. Especially the efficiency principle can be seen as a highlight of the four existing frameworks, since it is based on the economic efficiency principle from the GoM framework, the third principle from the 7PMG framework, and the diffuseness principle from cognitive dimensions framework. The new framework can be seen in Table 3. With this framework, specific aspects can be related to a certain level, to have a more structured comparison.

Case study level	**Does the model meet the case study requirements?**	
	-Requirements met -Requirements partly met -Requirements not met	
Design level	**Can (part of) the model be reused?**	**Is the workflow path displayed?**
	-Reusability possible -Reusability not possible	-Displayed at design time -Displayed at run time -Displayed at run time and design time
Language level	**How many elements are needed to create the model (efficiency)?**	**Can a group of elements be abstracted to one element?**
	A quantification of needed elements	-Abstraction possible -Abstraction not possible
Tool level	**Is documentation and (online) support available?**	**How user friendly is the tool?**
	-Available -Not available	-Low -Medium -High

Table 3: New comparison framework

4. Modeling

After selecting the process to model and the two tools to model it in, a first version can be modeled in both tools. This is explained in chapter 4.1. After the comparison of the two models, BPMone appears to be the better solution of the two. But there are some possible improvements for the Mendix model which are described in chapter 4.2.

4.1 First models

A first version of the process model is made in both tools. Section 4.1.1 describes the first Mendix model, section 3.1.2 shows the first BPMone model, and section 4.1.3 compares the two models. As described in chapter 1.6, each model is build using the strengths of its used tool, influenced by their own community.

4.1.1 Mendix model

For the glaucoma process a first model is made in Mendix. All the different steps in the process are modeled as different forms that can be filled out by the different players in the process. In line with the requirements, two kinds of users are implemented to meet the security standards. One user category has the name ‘Assistant’, whose access is limited to the forms where an appointment needs to be made. The other user category has the name ‘Doctor’. Users in this category have full access to the model. This can be arranged in the security settings from Mendix. For each created user, you can decide which forms and microflows this user has access to (see Appendix Figure 6).

To be able to create the different forms the users can fill out, a domain models needs to be created first. In this domain model all the forms with their attributes are defined, this can be seen in Figure 5. The names of the entities in the domain model are simply ‘Fxx’ to stress the default execution order, but the names of the forms that are visible in the application are more recognizable, as can be seen in Appendix Figure 7, where F02 is named ‘First examination’.

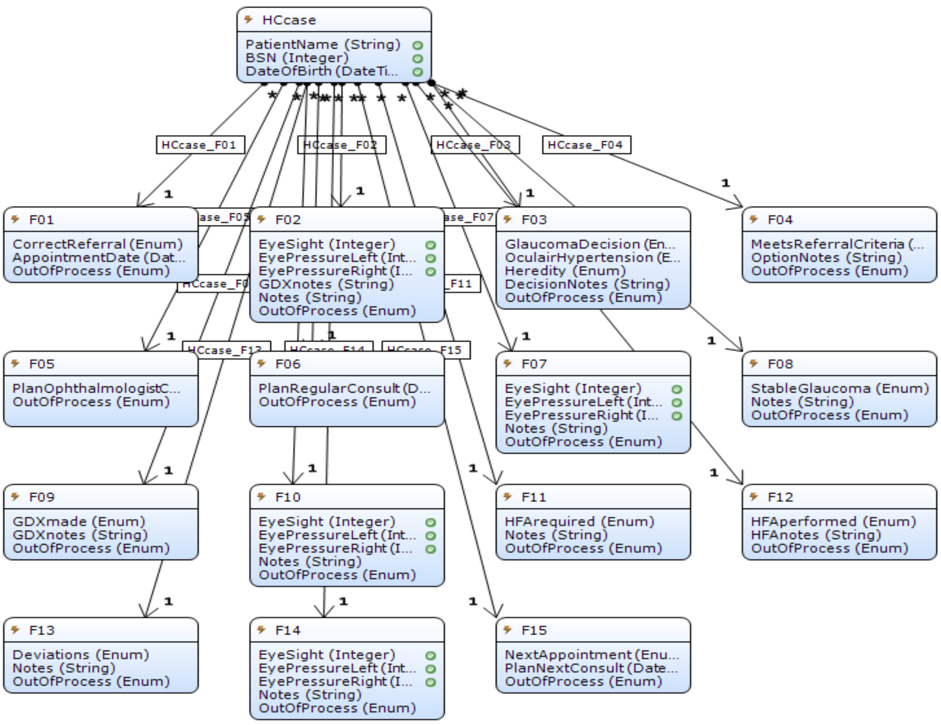


Figure 5: Mendix domain model for Glaucoma First Version

Attributes can be of different types, like an *integer* for the eye pressure, or a *string* for notes, and *Date and Time* for the date of birth from a patient. Choices can also be sort of type, which is referred to as *enumerations*. These enumerations can be manually defined, which makes it possible to give all sorts of names to the different options.

The different forms can be generated with the information found in the domain model. To be able to access these forms when the model is started, each form requires a unique microflow which creates a form when a new case – a new patient – is entered into the system. This is done by adding an 'event handler', a feature which executes all the different microflows that create all the forms, after creating a new case automatically. The different forms are similar as the example form found in Appendix Figure 7. All the forms can be accessed through the Home Form (see Appendix Figure 8), which is also the form that is shown when you start the model. To be able to go to all the different forms, a unique microflow is needed for each form. To be able to navigate from one form to the other with the 'next' button (see Appendix Figure 7) again a unique microflow is needed for each form. Finally to be able to see the status of the case, by checking which forms have been filled out already – the 'yes', 'no' or 'OoP' (short for Out of Process) which can be seen in Appendix Figure 8 – a unique microflow is needed for each form to check whether all the required information is recorded. All together a lot of forms, microflows, enumerations, entities, and attributes need to be created. To quantify this necessity of elements, Table 4 shows an overview of all required elements that are created manually.

Different elements needed	Amount	Total
Attributes	57	
Entities	16	
Enumerations	3	
Forms	17	
Microflows	60	
Elements in microflows	262	
Event handlers	30	**445**

Table 4: Needed elements in Mendix first model

Not all of the different elements will be discussed here, but a few of them will be explained, so the basics of the model can be understood. The complete Mendix model can be found in Share, which gives the opportunity to actually use the model. Share is an online sharing environment which can provide access to a tool which otherwise needs to be installed and configured (Van Gorp, 2011). The website contains multiple files of documentation on how to use the SHARE system. The virtual machine with the Mendix models can be accessed via the share website:
http://is.ieis.tue.nl/staff/pvgorp/share/?page=Home

By searching images on description, with the search keyword 'Muijres' the correct machine can be found.

Each form has a unique microflow to check whether the actions on the form has been completed – to account for the status of the case. This status is based on an enumeration (with the name OutOfProcess) with the possible values 'Yes' (for: this step is completed), 'No' (for: this step is not completed), and 'OoP' (for: the case is out of process). As an example of one of the completion microflows, Figure 6 shows the microflow to check the completeness of form 1 (or F01 as can be seen in the parameter symbol). The orange exclusive split is based on the attribute 'Correct referral' from form 1. If this is answered with a ‘yes’ in the form, the action ‘Change object’ (the blue rounded square on the top), changes the status attribute of form 1 into ‘Yes’. This can be seen at the Home form, in the completion table from all forms; the completion value for of form 1 is yes. If the correct referral attribute is answered with a ‘no’, the status attribute is changed into ‘OoP’ (the blue rounded square on the bottom performs this action), which stands for out of process. If nothing is answered to the referral question, no status change is performed (the ‘(empty)’ line which does not contain an action activity). The red diamond at the right of the figure is a merging symbol, to merge the lines that were split previously. Each form has such a microflow, but based on the number of attributes that should be filled out on the form, the number of elements on the microflow will increase as well. Since each attribute should be checked on completion.

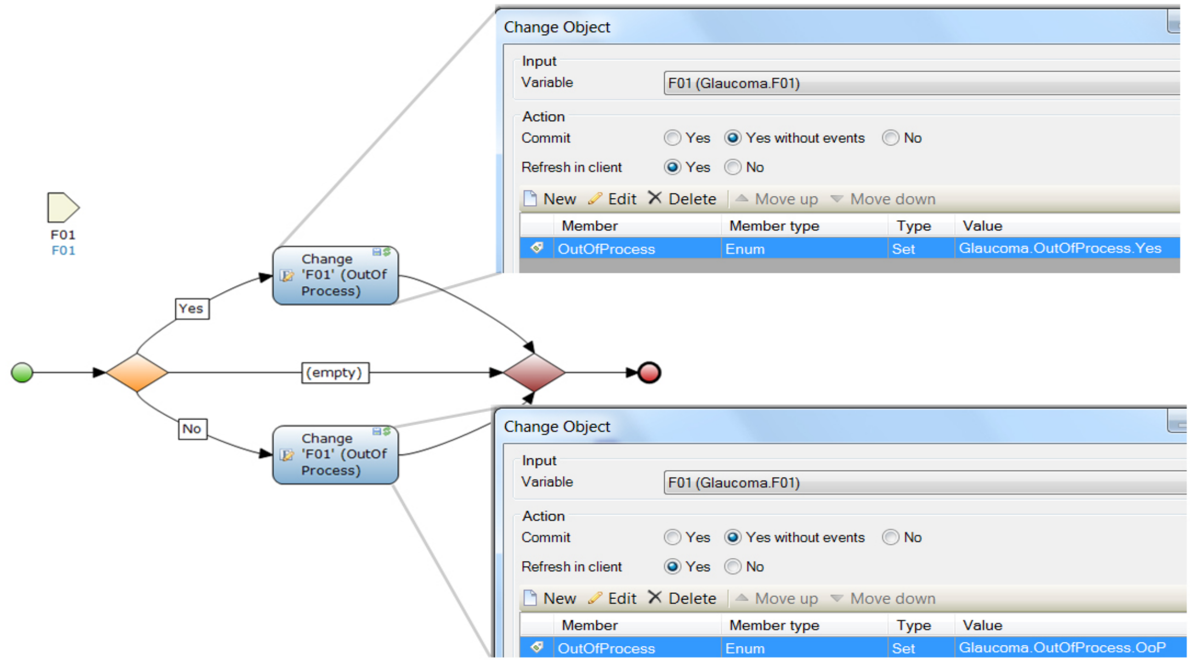


Figure 6: Microflow to check the completeness of form 1; from Mendix first version model

The microflows that make it possible to go from the home form to a selected form are fairly easy. Figure 7 shows the microflow to go to form 1. The first action activity finds the correct form 1 that belongs to this particular case (the parameter is now the HCcase). The second action activity shows form 1. If this form had already been opened before and some information was stored on it, this information is shown now. The microflows to show the other forms are all similar to this one.

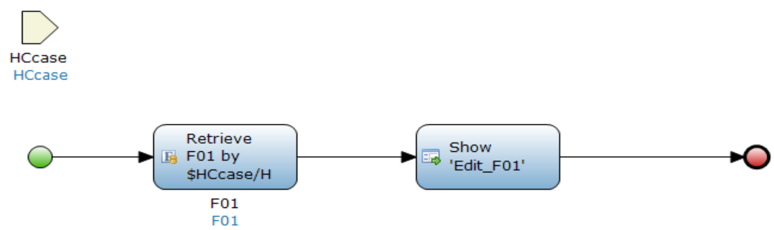


Figure 7: Microflow to directly go to form 1; from Mendix first version model

Finally there are microflows to activate the ‘next’ button on each form. Figure 8 shows the microflow to go from form 1 to form 2 (with the ‘next’ button on form 1). The first action activity (Change ‘F01’) makes sure all the information filled out on form 1 is saved. Next the exclusive split (orange diamond) splits the action based on what is answered to the question ‘is the patient referred correctly?’. If this is still empty, an information message is shown which reminds the user to fill out this attribute. If the answer is ‘no’, the home form is shown, since there is no further treatment possible, so the patient will be out of the system. When the answer is ‘yes’, the previously generated form 2 which belongs to this HCcase is retrieved, and shown.

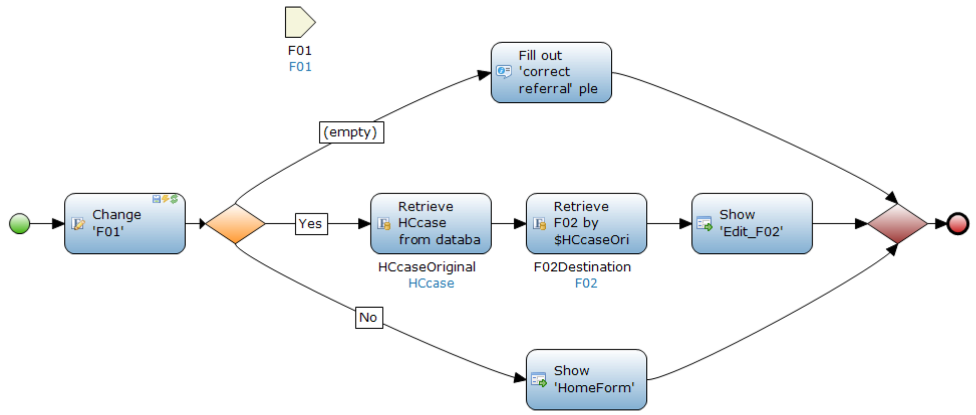


Figure 8: Microflow to go from form 1 to form 2; from Mendix first version model

Based on the different options on each form the other microflows are similar to this one, or a bit more extensive. By using the different navigation methods (starting at form 1 and using the ‘next’ button on each form, or using the ‘to Fi’ button to directly go to a form) the entire process can be followed. At the home form a list of cases can be found, sorted on the patient’s name. There is a search option to search for a patient based on its name, BSN number, or date of birth.

4.1.2 BPMone model

Just like with the Mendix model, in BPMone a model is created for the glaucoma process. Again all the different steps in the process are modeled as different forms that can be filled out by the different players in the process. Also in BPMone two kinds of users are implemented to meet the security requirements. One user category has the name ‘Assistant’, whose access is limited to the forms where an appointment needs to be made. The other user category has the name ‘Doctor’. Users in this category have full access to the model.

The first step in building a working BPMone model is creating the process model in the process designer. This can be seen in Figure 9, where the screen is divided a ‘process’ part (where the actually process overview is seen), a ‘data’ part (where all the different data elements and applications are shown), and a ‘role’ part (where the different users can be seen). After creating the different roles, the first activity can be created. Each activity can be seen as a form to be filled in by the user later on. The different parts of the form that should be filled in can be seen as ‘data elements’, so for each item that should be visible on the screen, a data element is added (for example ‘PatientName’ in Figure 9). To be able to see these elements on the form, an application should be added to each activity. After creating the entire process with these activities, data elements, and applications the steps in the process designer are done.

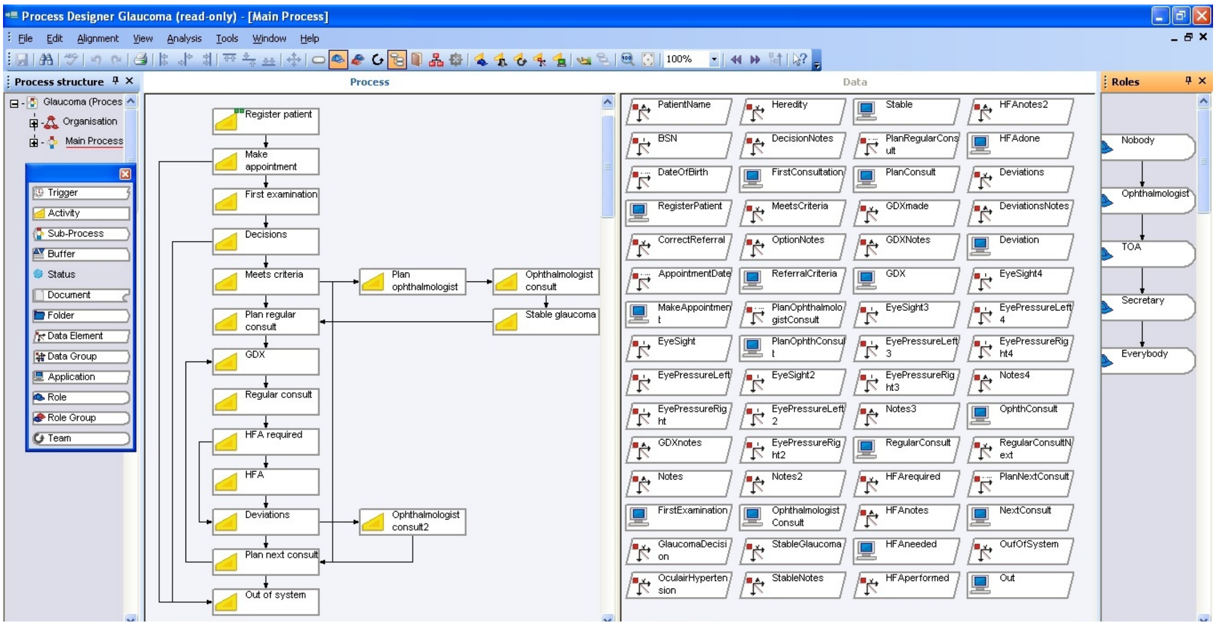


Figure 9: BPMone process designer; Glaucoma model

Just like with the Mendix models, a lot of different elements are needed to build this model. The quantification of these elements is shown in Table 5.

Different elements needed	Amount	Total
• Activities	17	
• Applications	17	
• Data elements	43	
• Forms	17	94

Table 5: Needed elements in BPMone model

The next step is to convert the model into a 'case type model', see Figure 10. In here the forms are created automatically, but they can be (and often have to be) adjusted manually. Figure 10 shows one of the forms, and the entire overview of the process. Each of the red lines indicates a guard on the line, to go to the correct next step, based on the choices made in the previous step.

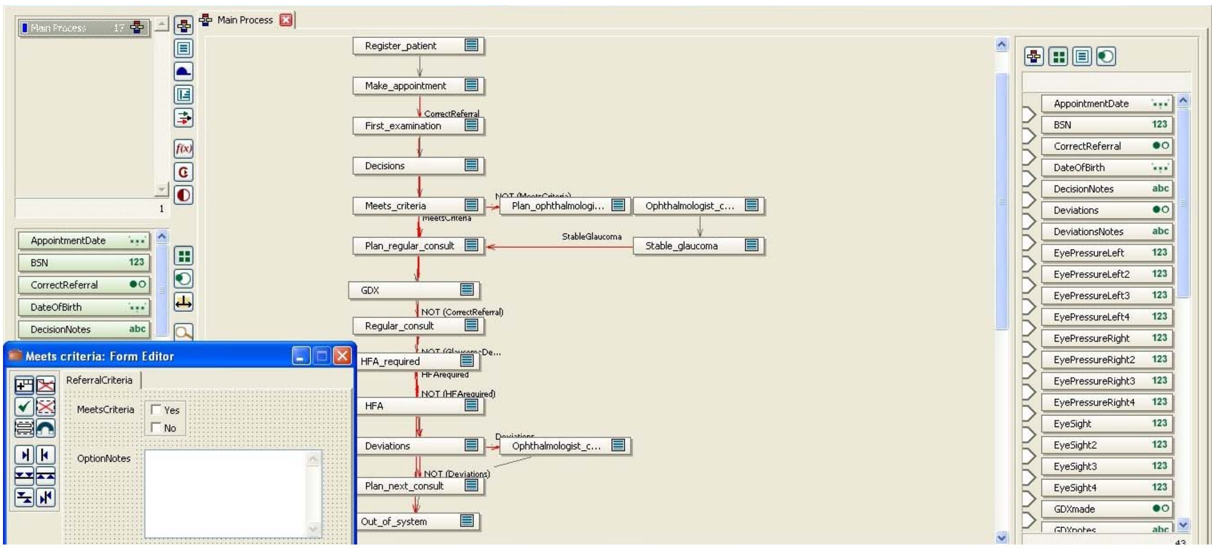


Figure 10: BPMone case type model; Glaucoma model

When the user logs in to the tool, an overview is seen like in Appendix Figure 9. The cases that are still in progress (Patient 1 – Patient 4) are seen in the worklist 'Glaucoma' as long as the user still has to perform tasks on that case. When the case is done, it can be found in the worklist 'Case Instances'. The flexibility a user has in this model is shown in Appendix Figure 10. For each task a 'skip' and a 'redo' function can be added. In this way the user is free to choose a new step, or redo a previously executed step. In this way the user can easily deviate from the uniform plan. An extra function that is built in with BPMone is the log function, you can access for each case you're working in. It records every value entered in the forms (also when the same step is redone), the time of entering the value, and the user that entered it. This can all be seen in Appendix Figure 11.

The BPMone model can also be found in Share, which gives the opportunity to actually use the model. The virtual machine with the BPMone model can be accessed via the share website: http://is.ieis.tue.nl/staff/pvgorp/share/?page=Home

By searching images on description, with the search keyword 'Muijres' the correct machine can be found.

Besides the modeling part of BPMone, there is also a monitoring section where detailed information can be found about the processes. In this way it can be discovered whether it is possible to adjust the process, by finding some bottlenecks for example. The monitoring aspect is not tested in this thesis, since it only gives useful outcome with real input. So after using the process model application for a while, and having generated a substantial number of cases, the outcomes of the monitoring section will be useful. Besides all sorts of different graphs and tables with the monitoring information, BPMone also presents some animation outputs, where the flow of the different cases throughout the process, or throughout the entire organization can be seen. Some examples of these animations can be seen in Figure 11.

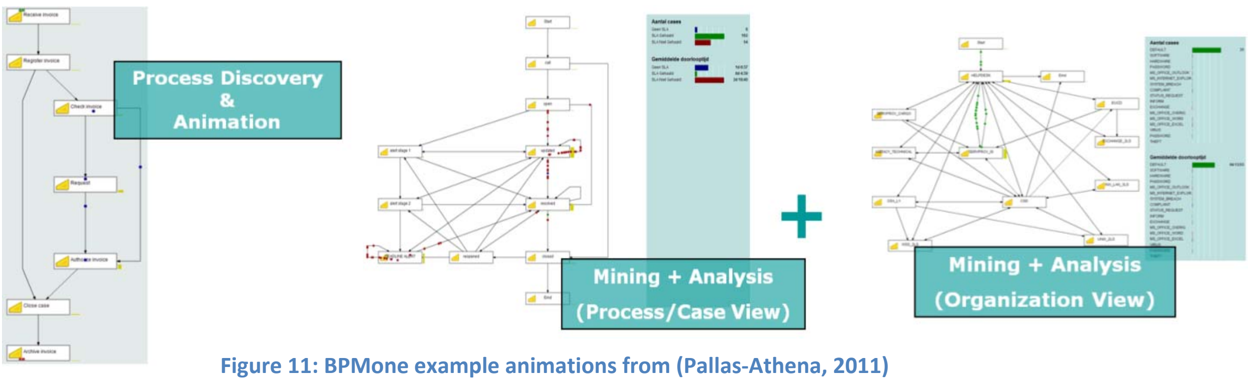


Figure 11: BPMone example animations from (Pallas-Athena, 2011)

4.1.3 Verifying the two applications using a S.W.O.T. analysis

This section shows the results of a S.W.O.T. analysis, to verify that the two applications are representing the same process model, perform the same tasks, and therefore can be compared. Both models are made to represent as many of the user requirements as possible. The main differences are on the built-in features, and the programming flexibility. The Mendix user has the possibility to add XML code, so the programming flexibility of Mendix is provided. Mendix provides the opportunity to the user to add programming concepts. Both options are not found in BPMone. BPMone has built-in 'skip/redo' functionality supporting the case handling paradigm. Mendix does not have this build-in functionality. The complete analysis is shown in Table 6.

Strengths	**Weaknesses**
BPMone: • Aimed at flexibility • Built-in 'skip/redo' option • Built-in 'form completion' check • Built-in feature to show workflow path at design time and run time • All data requirements processed Mendix: • All data requirements processed • Date reminding pop-up for appointments	BPMone: • No possibility to add date reminding pop-up for appointments • Requires unique development skills Mendix: • No built-in 'skip/redo' option • No built-in 'form completion' check
Opportunities	**Threats**
BPMone: • Monitoring opportunity after using the application for a while Mendix: • Opportunity to recreate the 'skip/redo' option • Opportunity to recreate the 'form completion' check • Opportunity to recreate the feature to show workflow path at design time and run time	BPMone: • Not a lot of programming freedom Mendix: • Possibility that there is too much programming freedom which can lead to higher chance of errors

Table 6: SWOT analysis

4.1.4 Comparing the two models with the comparison framework

The different principles from the comparison framework described in chapter 3 are discussed in this section. Table 7 gives the filled out framework for the models made in both tools. The (B) represents the BPMone model and the (M) represents the Mendix model.

Case study level	**Does the model meet the case study requirements?**	
	(B) Requirements met (M) Requirements partly met	
Design level	**Can (part of) the model be reused?**	**Is the workflow path displayed?**
	(B) Reusability not possible (M) Reusability not possible	(B) Displayed at run time and design time (M)Displayed at run time and design time
Language level	**How many elements are needed to create the model?**	**Can a group of elements be abstracted to one element?**
	(B) 94 (M) 445	(B) Abstraction not possible (M) Abstraction possible
Tool level	**Is documentation and (online) support available?**	**How user friendly is the tool?**
	(B) Available (M) Available	(B) Medium (M) High

Table 7: Comparing the two first version models

The workflow path is displayed at run time and design time in BPMone by a built-in feature. This feature is created in Mendix. Both tools have an extensive documentation and online support availability. The possibility to abstract a group of elements is available in Mendix, but not in BPMone. This feature benefits the user friendliness of the tool, since for example large microflows can be abstracted into smaller versions with better overview. Also the creation of forms is highly intuitive in Mendix, whereas for example the size of the different element on a form in BPMone cannot be quickly changed. The user has to manually insert the height and width in numbers, instead of the expected drag and drop option. Certain principles require some extra explanation, which is covered in the following corresponding sections.

Requirements

A quantification of how many of the requirements are met for each tool is shown in Table 8. Some of the theoretical requirements from chapter 1.2 are cooperative work support, communicability, and reusability. Reusability is not really provided in either of the tools. Concerning the communicability aspect, both applications have a clear overview of the process, so both applications can be used to communicate between the different users. The cooperative work support is reviewed in the S.W.O.T. analysis. Both applications are representing the correct model and whether the required options indicated by the Network can be used by the user of the application is shown in Table 8. This table shows the requirements set by the Eye Care Network. For each requirement it is shown whether Mendix and/or BPMone meets it. The possible answers are + for 'requirement met', - for 'requirement not met', and ++ for 'requirement met very good'.

Requirement	Mendix	BPMone
Accessibility options for different users	+	+
Overview of the entire system	+	+
Status overview	+	+
Possibility to deviate from the uniform plan	+	++
Possibility to add users	+	+
Process monitoring	-	+

Table 8: Relevance comparison based on requirements

Both tools have created different levels of accessibility for the different users. Both tools have created an overview of the entire system and an overview of the status of each case. The skip/redo functions give BPMone one more point compared to Mendix's possibility to deviate from the uniform plan. In Mendix it is possible to deviate from the standard sequence of steps, but when a step is redone, the information filled in on a form from the first time, will be lost. In both tools a user can easily be added. The process monitoring part is only available in BPMone. BPMone also has the build-in functionality to simulate the use of the process. Based on the input information an animation can be created to quickly see the cases run through the process. Mendix does not have this option. Based on Table 8, the requirements principle of the comparison framework is won by BPMone.

Required elements

To actually quantify the differences in modeling effort (efficiency), all the different elements that were needed to create two working models are summed up in Table 4 and Table 5. Mendix needs more than four times the number of elements that are needed to create the same model in BPMone, as can be seen in Table 7. Based on the opportunity to re-use (parts of) the models, both tools fall short. When a new model is created, the modeler basically has to start from scratch. So the required elements principle from the comparison framework is clearly won by BPMone.

Conclusion

In the context of this case study in the healthcare domain, with the required level of flexibility, the comparison framework in Table 7 shows the application build in BPMone is the better one of the two applications. More user requirements are met, and this is achieved with less elements. This is with the assumption made that no strange design choices are made which could have influenced the needed amount of elements in both applications.

4.2 Improved models

So in the context of this healthcare case study, BPMone proved to be the better of the two tools. a certain number of aspects of the Mendix application can be improved to see if this improved version scores better compared with the BPMone application. These improvements are valid for the case studied in this thesis, in this healthcare domain, with the flexibility by design aspect. Since the only adjustments that can be made to the Mendix application are in the design level, that level was adjusted. The adjustments made to the Mendix application were influenced by the stronger features of the BPMone application. For example 'skip/redo' feature is recreated in Mendix. Another improvement is aimed at improving the reusability aspect of the application, to compensate for the larger number of required elements. BPMone was already the winner, so no improvements are made for that model. First the improved Mendix model is explained and then the comparison framework is used again to make a new comparison.

4.2.1 Mendix model

Mendix's primary weakness is the lower level of abstraction, compared to BPMone. This results in the need of a higher specification effort, which leads to the risk of making bad design decisions. Decisions that are already taken implicitly in BPMone. One example of the need of a higher specification effort is the lacking process monitoring part. This part is standard in BPMone, but has to be created in Mendix. The further improvements made to the Mendix model are mainly aimed at improving the efficiency, and then specifically the re-using possibility. The created patterns are reusable and even generalizable outside of the healthcare domain, this is shown in section 4.2.1.1.

To recreate the skip and redo feature from BPMone, some additions were made to the Mendix model. The steps that can be performed more than once have received a logfile activity, so the information filled out during the first time is not lost, when the step is redone. On the home form there is the opportunity to directly go to each form the user wants, so when certain steps can be skipped, this is possible. In this way, the user can choose the skip and redo options at runtime.

The first Mendix model had one microflow for creating each form, so 15 microflows were needed to create all the used forms. With these 15 microflows, also 15 event handlers were needed to automatically create these forms after creating a new case. In the improved model, one initial microflow ('initHCcase') is modeled to create all the forms needed in a case. So this is 14 microflows and 14 event handlers less than in the first version. The microflows to check the form completeness and the microflows to go from the Home form to a selected form are the same as in the first version. The other major change is the underlying technique for the 'to next' button. Instead of one form for each button, like in the first version, there is now one microflow 'proceed', which either calls a next form from microflow 'toNextFormTrue' or from microflow 'to NextFormFalse'. These microflows are almost the same, so only of them is manually created and the other one is duplicated. In this way the improved model has more elements in its microflows, but since they are not manually made, this is not a problem. The only difference between the two microflows is the association that is retrieved in the beginning (see Figure 12). Either the NextFormTrue or the NextFormFalse association is retrieved. This is also based on an improvement in this model. All form entities are generalizations of the abstraction form 'AbstrForm'. The initial microflow decides the sequence of the flow, which varies for the NextFormTrue and the NextFormFalse association. The 'proceed' microflow checks all the values entered in the current form, and depending on these values, either the microflow 'toNextFormTrue' or the microflow 'to NextFormFalse' is called for. In this way for example the 'to next' button on form 8 opens either form 5 or form 6, depending on the information filled out on form 8. Figure 12 shows part of the 'proceed' microflow which makes this happen. When the outcome of the exclusive split is 'no', the microflow 'toNextFormTrue' is called, in which form 5 is given as output, when form 8 is the input (see bottom part of Figure 13).

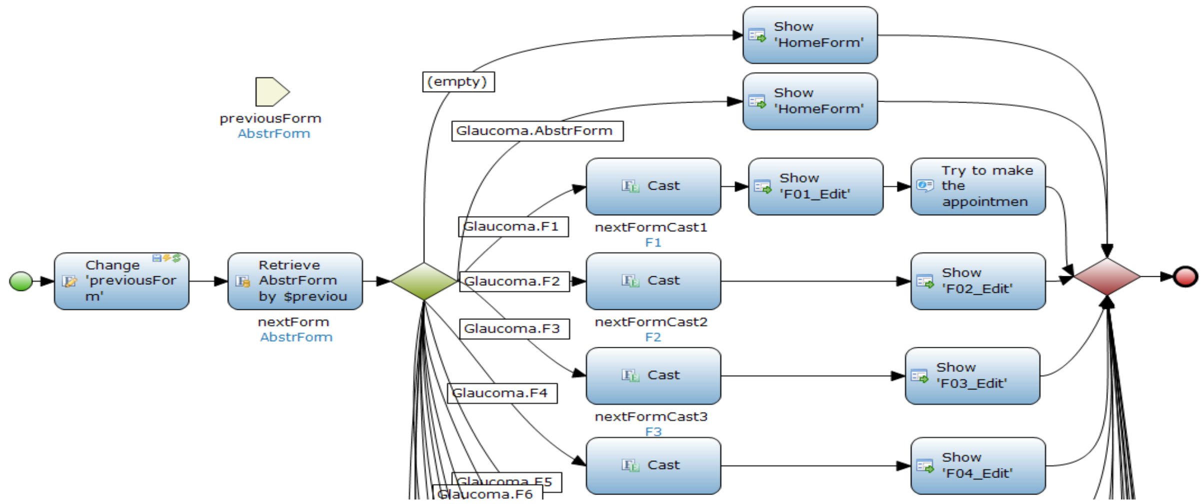


Figure 12: Part of microflow 'toNextFormTrue'; Mendix improved version model

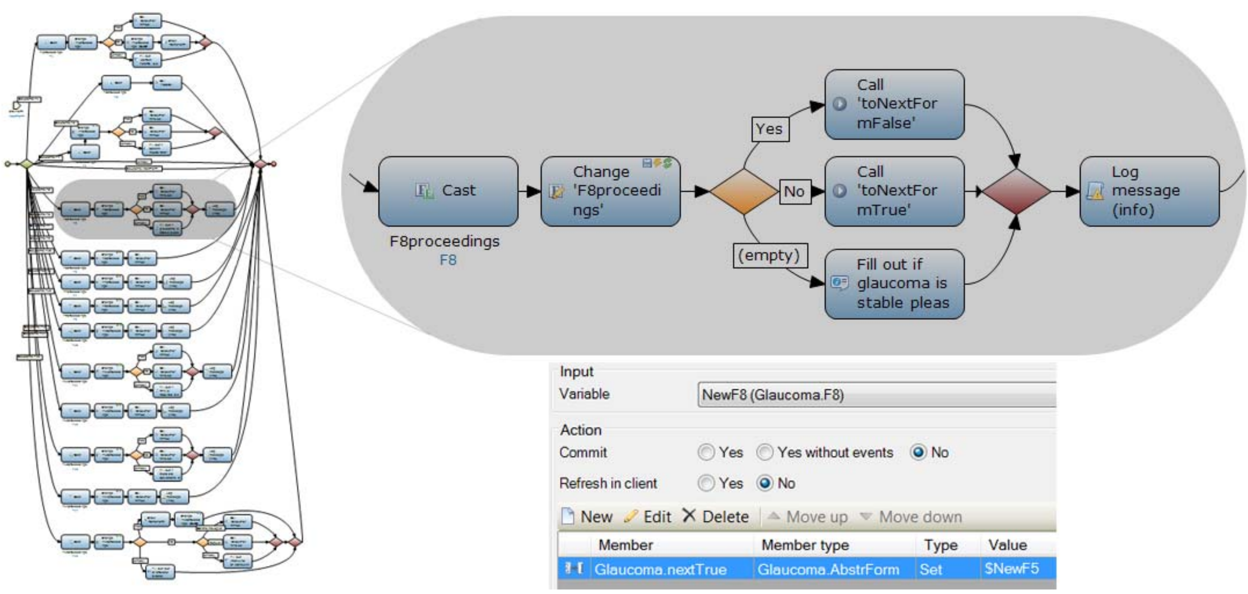


Figure 13: Part of microflow 'proceed'; Mendix improved version model

An extra sub-microflow is created for the path of choice (like the example for form 8 in Figure 13) for form 3. This is done since this form is rather big because it includes three options in one form. So in the 'proceed' microflow only a link to the sub-microflow is shown.

Besides these changes, the working of the model is the same as the first version model. The model starts again with the Home form. A new case can be created there, and the process can be started by clicking on button 'to F1'. Or another form can be chosen as a starting form, since all forms can be directly accessed from the Home form.

4.2.1.1 Generalizable improvements

As mentioned in section 4.2.1 are the patterns created in Mendix reusable in other models, and even generalizable to domains outside of the healthcare. To prove this claim, the patterns are reused in another model. The model is based on a process in a bicycle factory. This process is described first, before the process model is discussed. This process is used exclusively for research and educational purposes. Adapted from "Towards Designing Modular and Evolvable Business Processes", Van Nuffel, Universiteit Antwerpen 2011 (Nuffel, 2011).

Process description

A small company is manufacturing custom build bicycles. As soon as the sales department receives an order, the business process is started. Each manufacture request is denoted as a separate order. In the first step, the order is registered and the customer details are noted as well. The next step is to evaluate the order. At first there is an engineering check and a financial check. First, based on the number of parts needed and the way to procure them, the order type is determined:

- If the bicycle needs less than 100 parts, either bought or custom made, the order is categorized as "easy";
- If the bicycle needs between 100 and 500 parts, the order is categorized as:
    - "easy": if all parts can be bought;
    - "difficult": if some parts need to be custom built.
- If the bicycle needs more than 500 parts, it is categorized as "complex".

Next, the financial check depends on the outcome of the engineering check: if the order is easy and the expected revenue of the order surpasses $5,000, the order receives a financial status "ok". A difficult order is only accepted if the expected revenue is at least $7,500; if the revenue ranges between $5,000 and $7,500, the order is attributed the status "medium". A complex order will only be accepted if the estimated revenue exceeds $10,000, and receives the status "medium" when the revenue is between $7,500 and $10,000.

After the engineering and financial checks, a senior manager verifies the order to check whether it fits the style of the company. Orders with the financial status "ok" will be accepted, regardless of the manager's decision. Orders with a Medium financial evaluation will only be accepted if the evaluation of the manager is "high". In case the order is rejected, the customer receives a notification. When the order is accepted, the engineering department will start including the order in the production schedule and performs a number of assembly preparation tasks; checking if the required quantity of each part is available. When all the parts are available, the bicycle can be manufactured. If not all of the parts are available, the missing parts need to be ordered. At that moment is check is performed whether it is possible to order the needed parts. If this is not the case, the customer is informed that the order cannot be completed at this point in time. When the parts can be ordered, this will happen, and the manufacturing of the bicycle will start when the parts arrive. After assembly, the bicycle is shipped and invoiced. When the payment from the customer is received, the case is closed. Appendix Figure 12 shows the process model of this case, which is used as an input for the Mendix model.

So the domains of the cases differ, but the same flexibility type (*flexibility by design*) is required. The main process flow is known (and shown in the process model in Appendix Figure 12), but some deviations are possible again. The managerial check is not always necessary, as described, so it should be possible to skip this step. Also the opportunity to redo certain steps is required in this case. For example, when something goes wrong during the manufacturing phase, this phase needs to be executed again. The check whether all parts are available could also be performed more than once. Especially with the complex bicycles (more than 500 parts), it can happen that the manufacturer discovers that more parts are needed, during the manufacturing phase. At that moment it should be checked whether those parts are also available. When these parts are not available, they need to be ordered, so this process can also be performed more than once. So the same skip and redo functionality from BPMone, that was recreated in the Mendix patterns can be used in this case. The steps that can be performed more than once (form 4 and form 6) have received a logfile activity, so the information filled out during the first time is not lost, when the step is redone. The home form has the opportunity to directly go to each form the user wants, so when certain steps can be skipped, this is possible. In this way, the user can choose the skip and redo options at runtime.

Bicycle model in Mendix

The basic idea of this model is the same as for the glaucoma model. There is a Home form (see Appendix Figure 13) with which the application starts. On it, all the different forms can be accessed via the 'To Fi' buttons. The microflows connected to these buttons are re-used from the improved glaucoma model. Besides that, the status of the case can be seen on the Home form, by checking whether each form is completed or not. The microflows doing this are also re-used from the improved glaucoma model. Next there is also the microflow 'initCase' that is performed automatically when a new case is started. This microflow shows the sequence throughout the process. This microflow is re-used from the 'initHCcase' from the improved glaucoma model. Then there is the 'proceed' microflow which is also adapted from the improved glaucoma model. As can be seen in Figure 14, the idea is the same. The information filled out in form 4 is checked, and the exclusive split decides whether 'toNextFormTrue' or 'to NextFormFalse' is called for. Form 4 checks whether all the parts are available. If not, 'toNextFormFalse' is opened, and as can be seen in the bottom part of Figure 13, form 6 will be opened next. Form 6 is the check whether it is possible to order parts. This is all in line with the process model in Appendix Figure 13.

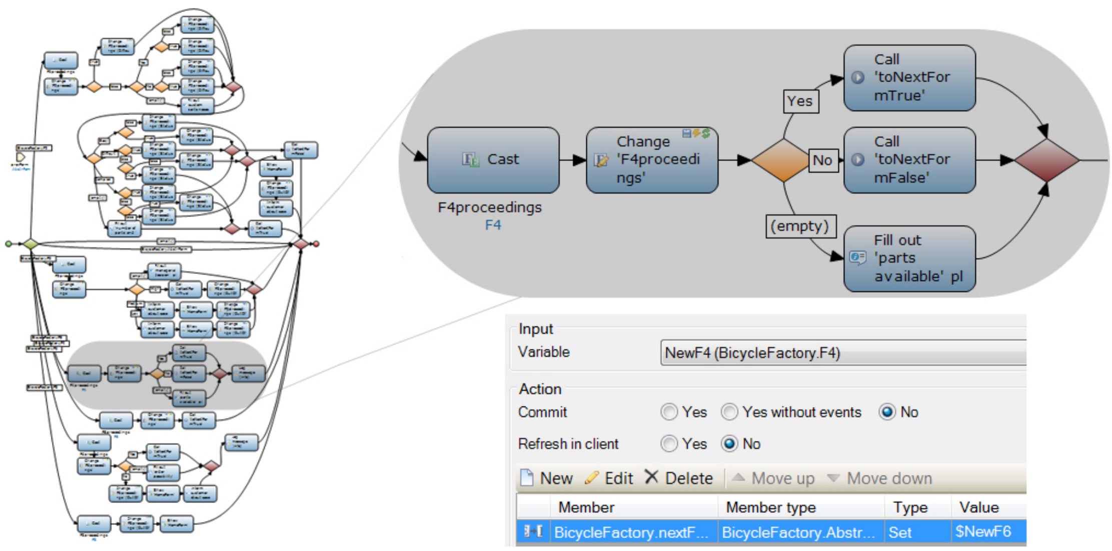


Figure 14: Part of microflow 'proceed'; Mendix bicycle factory model

So actually all of the microflows and the forms used in this bicycle factory model are adjusted version, reused from the improved glaucoma model, which is a plus for the economic efficiency. This is further described in section 4.2.2.

4.2.2 Comparing the two models with comparison framework

The Mendix model is changed a lot, so all these values are reevaluated. The BPMone model has not changed, so these values are the same as in chapter 4.1. The comparison framework is used again to compare the improved Mendix model, with the first version BPMone model. Table 9 gives the filled out framework for the models made in both tools. The (B) represents the BPMone model and the (M) represents the Mendix model.

Case study level	**Does the model meet the case study requirements?**	
	(B) Requirements met (M) Requirements partly met	
Design level	**Can (part of) the model be reused?**	**Is the workflow path displayed?**
	(B) Reusability not possible (M) Reusability possible	(B) Displayed at run time and design time (M)Displayed at run time and design time
Language level	**How many elements are needed to create the model?**	**Can a group of elements be abstracted to one element?**
	(B) 94 (M) 474	(B) Abstraction not possible (M) Abstraction possible
Tool level	**Is documentation and (online) support available?**	**How user friendly is the tool?**
	(B) Available (M) Available	(B) Medium (M) High

Table 9: Comparing the improved Mendix model with the first BPMone model

The requirements principle has not changed in Table 9, but it has changed a bit. This can be quantified by comparing how many of the previously noted requirements are met for each tool. Table 10 shows the requirements set by the Eye Care Network. For each requirement it is shown whether Mendix and/or BPMone meets it. The possible answers are + for 'requirement met', - for 'requirement not met', +/= for 'requirement partly met', and ++ for 'requirement met very good'.

Requirement	Mendix	BPMone
Accessibility options for different users	+	+
Overview of the entire system	+	+
Status overview	+	+
Possibility to deviate from the uniform plan	+	++
Possibility to add users	+	+
Process monitoring	+/-	+

Table 10: Relevance comparison based on requirements

The only part missing for the Mendix model in this comparison is the process monitoring part. As improvement to the improved glaucoma model, some logging activities are added. The forms that can be accessed more than once per case, have received a log-activity in this version of the model. Since no built-in process monitoring activities are available, the BPMone model still wins on this requirement, but the Mendix model has already improved. Based on Table 10, the requirements principle of the comparison framework is still won by BPMone.

Besides this requirements principle, the only other two principles that have changed are the number of used elements, and the reusability of the model. To quantify these differences both principles are discussed in the following section.

Number of elements needed & reusability

To actually quantify the differences in modeling effort, all the different elements that were needed to create two working models are summed up in Table 11. The numbers between brackets are the values for the first version Mendix model.

Tool	Different elements needed	Amount	Total
Mendix	• Attributes	45 (57)	
	• Entities	17 (16)	
	• Enumerations	3 (3)	
	• Forms	17 (17)	
	• Microflows	35 (60)	
	• Elements in microflows	341 (262)	
	• Event handlers	16 (30)	**474 (445)**
BPMone	• Activities	17	
	• Applications	17	
	• Data elements	43	
	• Forms	17	**94**

Table 11: Comparing elements needed in both tools for the improved version

Mendix still needs more than four times the number of elements that are needed to create the same model in BPMone, as can be seen in Table 8. But what is improved in this version is the opportunity to re-use (parts of) the model. The reusability of the improved model has been proven in section 4.2.1.1. The model is not only reusable for most parts in a similar case, but it is even reusable for a case from an entirely different domain (from healthcare to manufacturing) but with the same flexibility by design requirement.

So maybe more elements are needed in this improved model, which lowers the efficiency, but the level of reusability is very high, which improves the efficiency. This higher number of elements only need to be created once, after which they can be used for other models. So added the number of elements is a one time job, which can save a lot of work in the futures. As described in section 4.2.1.1, almost all elements were reused in the bicycle factory application. Since all attributes, entities, enumerations, forms, and microflows need to be adjusted, it still requires some work, so these elements cannot be neglected. The event handlers that were reusable did not require any adjustments, so these elements can be neglected. But the biggest difference is in the amount of elements needed in the microflows. Still a lot of those elements needed to be adjusted or even some elements needed to be added, but a good amount of elements were reusable without adjustment. Since the amount of elements needed in the microflows depends on the data requirements (which decides the amount of information on each form), it is difficult to quantify an exact number of reusable elements in the microflows that will not require any alteration, if the microflows were reused in another case. But when this application would be reused for a similar sized case, and the assumption is made that half of the elements in the microflows can be reused, without alteration, that is already 170 elements. The difference between the required elements in Mendix and BPMone was 380. When the improved Mendix model is reused for three similar sized cases (with the same flexibility requirements), the Mendix model is more efficient than the BPMone model, since the BPMone model falls short on the reusability aspect. Because every activity or data element needs to be altered for a new model, it is probably faster to start from scratch when making a new model. The Eye Care Network has different hospitals connected, who treat all kinds of eye diseases according to standardized treatment paths. From this aspect it is not that difficult to find three similar cases that can be model reusing the improved Mendix model. Taking this into account, the economic efficiency principle is won by the improved Mendix model.

Conclusion

In the context of this case study in the healthcare domain, with the required level of flexibility, the comparison framework in Table 9 shows the application build in BPMone still seems the better one of the two applications. But when the reusability aspect as described in the previous section is taken into account, and the Eye Care Network (or any other company that requires process models) requires more than one process model application, Mendix is also a good alternative, since they also win on user friendliness. But looking at the user requirements, the process monitoring aspect is still won by BPMone. So actually no clear winner can be seen with these improvements made to the Mendix model.

5. Conclusion

The goal of the present study was to examine which BPM approach is the best in meeting the requirements for the business process of the Eye Care Network. Furthermore, the study generalized the solutions outside of this specific healthcare domain. The business process chosen to model was the treatment path a glaucoma patient would undergo in a hospital connected to the Eye Care Network. The Eye Care Network had multiple goals for the business process models. First the application should support automated execution, by using the application to support the hospital staff in its work. In this way not all tasks are required to be performed manually which can save both time and errors. Besides this enactment goal, the Eye Care Network has some other goals which include facilitating communication between the different players in the network, and possibly supporting process improvement. By using process monitoring, and the option to do some calculations on these outcomes, the actual standard patient flow can be found out. This can be used to possibly adjust the standard treatment path when it shows that this is not the most used path. Some of the requirements linked to these goals are cooperative work support, communicability, and reusability. These requirements were also considered at the comparing of the two approaches. Besides these theoretical requirements for the process models, the Eye Care Network has provided some user requirements for the models to comply to. The first requirement is process monitoring support. Furthermore, there should be the option to add new users. It should be able to divide these users into different categories, who have different levels of access to the models. It should be possible to have an overview of the entire process, as well as a status overview of the different cases in the process. Finally, there was a flexibility requirement, which demanded the possibility to deviate from the uniform plan. These deviations can be seen as anticipated changes, that can be dealt with supporting strategies during the design of the application. In other words, the required type of flexibility for this particular case was *flexibility by design*.

To meet (as many of) these requirements, two different BPM approaches were studied and used. The first approach was the case handling paradigm, which has the goal to overcome the limitations of former existing workflow management systems. The second one is model driven engineering (MDE), as a software engineering approach. For each approach, a matching tool was chosen to produce the model. BPMone from Pallas Athena was selected as case handling tool, while Mendix was the selected tool for MDE. Although Mendix is not a typical MDE tool, it falls short on two of the three MDE concepts, it is a popular tool, with Dutch roots, well-known customers and well-known partners. Therefore, Mendix was chosen.

Based on already existing frameworks, a comparison framework was formed to objectively compare the models made by each approach. To have a complete comparison the framework described the following four levels; case study level, design level, language level, and tool level. The seven principles divided over these four levels were; meeting the requirements, model reusability, workflow path display, efficiency (amount of elements needed), element abstraction possibility, user support, and user friendliness.

With both tools, a first version of the glaucoma process model was prepared. All the different aspects of the models were considered and compared using the comparison framework. BPMone was the better one meeting the requirements and on the number of elements needed. The principles reusability, workflow path display, and documentation showed no clear difference between the two tools. Mendix was the better one based on the abstraction and the user friendliness principle. The overall score is in favor of the glaucoma model, made with BPMone.

Although BPMone seems to be the best, the Mendix model could still be improved. The only adjustments that can be made to the Mendix application are in the design level, so that level was adjusted. The adjustments made to the Mendix application were influenced by the apparently stronger features of the BPMone application. For example, the 'skip/redo' feature is recreated in Mendix. Also the reusability aspect of the application was improved, to compensate for the larger number of required elements. BPMone was already the winner, so no improvements are made for that model. Both models lacked the reusability aspect, although this was one of the theoretical requirements. Therefore, this was the main focus for improving the Mendix model. A number of patterns were added to the Mendix model to improve the reusability of the model. To prove these patterns were reusable, a totally different case was described and the improved glaucoma model was reused to model this new case. It was not only a different case, but it was also a case from a different domain, but with the same flexibility by design. This case was no healthcare case, but it represented the manufacturing process of a small bicycle factory. All the microflows and all the forms used in the Mendix model were reused from the improved glaucoma model.

Based on this new Mendix model, and the unchanged BPMone model, a new comparison was performed. The workflow display, abstraction option, documentation and user friendliness principles have not been changed, so these values were the same as with the first version models. The changes made to the Mendix model were visible in the reusability and efficiency principle. Although the number of elements needed in the improved Mendix model is still a lot higher than the BPMone model, and even higher than the first Mendix model, the reusability can make up for this issue. All the comparison principles taken together resulted in the finding that there was no longer a clear winner. BPMone was only the better one on the requirements principle. The principles workflow display and documentation showed no clear winner, and Mendix was the better tool based on the reusability, abstraction, and user friendliness principle. The number of elements required principle depends on the situation.

The goal of the present study was to check which of the two approaches – case handling or MDE – meets the requirements from the Eye Care Network the best. Two comparisons were made. Case handling (BPMone) scored higher than MDE (Mendix) on one of them, while the other comparison showed no clear winner. Since both approaches were acceptable, it really depends on the situation which one of them is most suitable. When only one process model is needed (with the same user requirements and flexibility requirements as stated in this study), modeling with BPMone is highly preferable based on the required modeling effort. When multiple processes need to be modeled, Mendix is recommendable based on the reusability aspect. Developing the first model would require more time compared to the BPMone approach, but all other models can be created by reusing parts of this first model. With the assumption that half of the number of elements needed in the Mendix microflows can be reused without alteration, the extra effort required for the first Mendix model is accounted for after making three similar models. In this case, with the Eye Care Network, who treat all kinds of eye diseases according to standardized treatment paths, it is not that difficult to find three similar cases that can be modeled reusing the improved Mendix model.

Future research

As previously mentioned, the improved Mendix model still falls short on the process monitoring aspect, which is with a standard option in BPMone. Therefore, some new patterns could be created in Mendix, to recreate this feature. With this extra feature in place, a new comparison could be performed to check whether Mendix would then perform better than BPMone.

Another interesting subject for future research is the aspect of Mendix not being a typical MDE tool. As described in the introduction, Mendix falls short on two out of three main MDE concepts (metamodel and transformations). Therefore, it could be examined if Mendix would score even better – compared to BPMone – when it would become a more typical MDE concept.
Moving towards the MDE concept means complying to more than one of the three main MDE concepts. After creating the different patterns in Mendix, a subject for future research could be adding transformations to the Mendix solution. For example, when the initially created microflow 'initHCcase' would be automatically generated by means of model transformations. In line with the previous work of Van Gorp and others (Van Gorp & Eshuis, 2010), some modeling efforts can be avoided. This could possibly lead to even better results for the Mendix model, compared to the BPMone model.

References

Aalst, W. v., & Berens, P. (2001). Beyond Workflow Management: Product-Driven Case Handling. *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001)* (pp. 42-51). New York: ACM Press.

Aalst, W. v., & Hee, K. v. (2002). *Workflow Management: Models, Method, and Systems.* Cambridge: MIT press.

Aalst, W. v., Hofstede, A. t., & Weske, M. (2003). Business Process Management: A Survey. *Proceedings of the 1st International Conference on Business Process Management* (pp. 1-12). Berlin: Springer-Verlag.

Aalst, W. v., Stoffele, M., & Wamelink, J. (2003). Case handling in construction. *Automation in Construction* , 303-320.

Aalst, W. v., Weske, M., & Grünbauer, D. (2005). Case Handling: a new paradigm for business process support. *Data and Knowlege Engineering* , 129-162.

Badica, C., & Badica, A. (2003). Business Process Modelling using Role Activity Diagrams. *Sintes 11.* Craiova.

Badica, C., Badica, A., & Litoiu, V. (2003). Role activity diagrams as finite state processes. *Second International Symposium on Parallel and Distributed Computing* (pp. 15-22). Ljubljana: Wiley, John & Sons, Incorporated.

Becker, J., Rosemann, M., & von Uthmann, C. (2000). Guidelines of business process modeling. *Business Process Management Models, Techniques, and Empirical Studies* , 30-49.

Bézivin, J. (2006). Model driven engineering: an emerging technical space. In R. Lämmel, J. Saraiva, & J. Visser, *Generative and Transformational Techniques in Software Engineering* (pp. 36-64). Springer.

Bizagi. (2011). *Business process management, BPM and Workflow Automation Solutions | Bizagi BPMS*. Retrieved May, 2011, from http://www.bizagi.com/

Bosilj-Vuksic, V., Ceric, V., & Hlupic, V. (2007). Criteria for the Evaluation of Business Process Simulation Tools. *Interdisciplinary Journal of Information, Knowledge, and Management* , 73-88.

Business Process Management Initiative (BPMI). (2004). *Business Process Modeling Notation.* BPMI.org.

Curtis, B., Kellner, M., & Over, J. (1992). Process modeling. *Communications of the ACM* , 75-90.

Davenport, T., & Short, J. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review* , 11-27.

Favre, J. (2004). Towards a basic theory to model model driven engineering. *Proceedings of the workshop on software model engineering (Satellite workshop at the 7th International Conference on the UML).* Lisboa.

Favre, J., & NGuyen, T. (2004). Towards a megamodel to model software evolution through software transformation. *SETRA Workshop.* Elsevier ENCTS.

Fowler, M. (2004). *UML Distilled.* Boston: Perason Education, Inc.

France, R., & Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. *Future of Software Engineering* .

Gorp, P. V. (2011). *Share*. Retrieved on July 2011, from Sharing Hosted Autonomous Research Environments: http://is.ieis.tue.nl/staff/pvgorp/share/?page=Home

Gorp, P. V., & Eshuis, R. (2010). Transforming Process Models: executable rewrite rules versus a formalized Java program. *Applications track of the ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems* (pp. 258-272). Oslo: Lecture Notes in Computer Science.

Green, T., & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages and Computing* , 131-174.

Guenther, C., & Aalst, W. v. (2005). Modeling the case handling principles with colored Petri nets. *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools* (pp. 211-230). Aarhus, Denmark: University of Aarhus.

Hailpern, B., & Tarr, P. (2006). Model-driven development: The good, the bad, and the ugly. *IBM Systems Journal* , 451-461.

HandySoft. (2011). *BizFlow Plus Handysoft*. Retrieved June, 2011, from HandySoft: http://www.handysoft.com/content/bizflow-plus

Het Oogziekenhuis Rotterdam. (2009). *Staar (Cataract)*. Retrieved June 2011, from Het Oogziekenhuis Rotterdam: http://www.oogziekenhuis.nl/oogaandoeningen-en-oogafwijkingen/staar-cataract-behandelingen.html

Het Oogzorgnetwerk. (2010). *Achtergrond en initiatief*. Retrieved June 2011, from Het Oogzorgnetwerk: http://www.oogzorgnetwerk.nl/

Hlupic, V., Paul, R., & Irani, Z. (1999). Evaluation framework for simulation software. *International Journal of Advanced Manufacturing Technology* , 366-382.

IBM. (2011). *IBM - FileNet Business Process Manager - Software*. Retrieved June 2011, from IBM: http://www-01.ibm.com/software/data/content-management/filenet-business-process-manager/

Korne, D. F., Sol, K. J., Custers, T., van Sprundel, E., Ineveld, M. v., Lemij, H. G., et al. (2009). Creating patient value in glaucoma care: applying quality costing and care delivery value chain approaches. *International Journal of Health Care Quality Assurance , 22* (3), 232-251.

Meertens, L., Iacob, M., & Nieuwehuis, L. (2010). Goal and model driven design of an architecture for a care service platform. *SAC'10* (pp. 1-7). Sierre, Switzerland: ACM.

Mendix. (2010). *Modules - Reference Guide 2.5*. Retrieved June 2011, from Mendix Community Platform: https://world.mendix.com/pages/releaseview.action?pageId=9208487

Mendling, J., Reijers, H., & Aalst, W. v. (2008). Seven process modeling guidelings (7PMG). *QUT ePrints Report, Queensland University of Technology* .

Motta, G., & Furuie, S. (2003). A contextual role-based access control authorization model for electronic patient record. *Information Technology in Biomedicine* , 202-207.

Mutschler, B., Weber, M., & Reichert, M. (2008). Workflow management versus case handling: results from a controlled software experiment. *Proceedings of the ACM Symposium on Applied Computing*, (pp. 82-89).

Norton, D. (2009). Cool Vendors in Application Development, New Tools. *Gartner* .

Novotný, T. (2009). *Workflow - BPM Systémy.*

Nuffel, D. V. (2011). *Towards Designing Modular and Evolvable Business Processes.* Antwerpen: Universiteit Antwerpen.

OMG. (2008). *Business Process Management Initiative*. Retrieved June 2011, from http://www.bpmi.org/

Oogartsen.nl. (2011, April 14). *Staar (Cataract)*. Retrieved June 2011, from Ogen: http://www.oogartsen.nl/oogartsen/ooglens_staar/staaroperatie_cataract/

Outsystems. (2011). *Agile Development for Enterprise Web Applications*. Retrieved May 2011, from http://www.outsystems.com/

Pallas-Athena. (2011, 3). *Business Process Management Software - Pallas Athena*. Retrieved June 2011, from Pallas Athena: http://www.pallas-athena.com/

Paul, R., Giaglis, G., & Hlupic, V. (1999). Simulation of Business Processes. *The American Behavioral Scientist* , 1551-1576.

Peeters, A. (2008). *Primary open-angle glaucoma and ocular hypertension: cost-effectiveness of early detection and treatment.* Maastricht University Hospital, Department of Ophthalmology. Maastricht, The Netherlands: Datawyse | Universitaire Pers Maastricht.

Peltier, M., Bézivin, J., & Guillaume, G. (2001). MTRANS: A general framework based on XSLT for model transformations. *WTUML'01, Proceedings of the Workshop on Transformations in UML.* Genova.

PNMsoft. (2011). *Sequence BPM software*. Retrieved June 2011, from PNMsoft Business Process Management software: http://www.pnmsoft.com/sequence_bpm_workflow.aspx

Reijers, H., Rigter, J., & van der Aalst, W. (2003). The case handling case. *International Journal of Cooperative Information Systems* , 365-391.

Rodríguez-Priego, E., García-Izquierdo, F., & Rubio, A. (2010). Modeling Issues: A Survival Guide for a Non-expert Modeler. *ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems.* Oslo: IEEE.

Roos, D. (2009, april 8). *Mendix listed as a cool vendor in Gartner report.* Retrieved Februari 2011, from Mendix: http://www.mendix.com/blog/mendix-gartner-cool-vendor/

Salimifard, K., & Wright, M. (2001). Petri net-based modelling of workflow systems: An overview. *European Journal of Operational Research* , 664-676.

Schmidt, D. (2006). Model-Driven Engineering. *IEEE Computer* , 25-31.

Schonenberg, H., Mans, R., Russell, N., Mulyar, N., & Aalst, W. v. (2008). Towards a taxonomy of process flexibility. *CAiSE* , 81-84.

Seidewitz, E. (2003). What models mean. *IEEE Software* , 26-32.

Sendall, S., & Kozaczynski, W. (2003). Model transformation: the heart and soul of model-driven software development. *IEEE Software* , 42-45.

Sinur, J., & Hill, J. (2010). *Magic Quadrant for Business Process Management Suites.* Gartner.

Skelta. (2011). *BPM Workflow Software*. Retrieved May 2011, from Invensys Skelta: http://www.skelta.com

Truyen, F. (2006). *The fast guide to model driven architecture.* Cephas Consulting Corp.

Vanderfeesten, I., Reijers, H., & van der Aalst, W. (2009). Case Handling Systems as Product Based Workflow Design Support. *Enterprise Information Systems* , 187-198.

Vanderfeesten, I., Reijers, H., & van der Aalst, W. (2006). Product Based Workflow Design with Case Handling Systems. In *BETA Working Paper Series.* Eindhoven: Eindhoven University of Technology.

Vasko, M., & Dustdar, S. (2006). A view based analysis of workflow modeling languages. *Proceedings of the 14th Euromicro international conference on parallel, distributed and network-based processing*

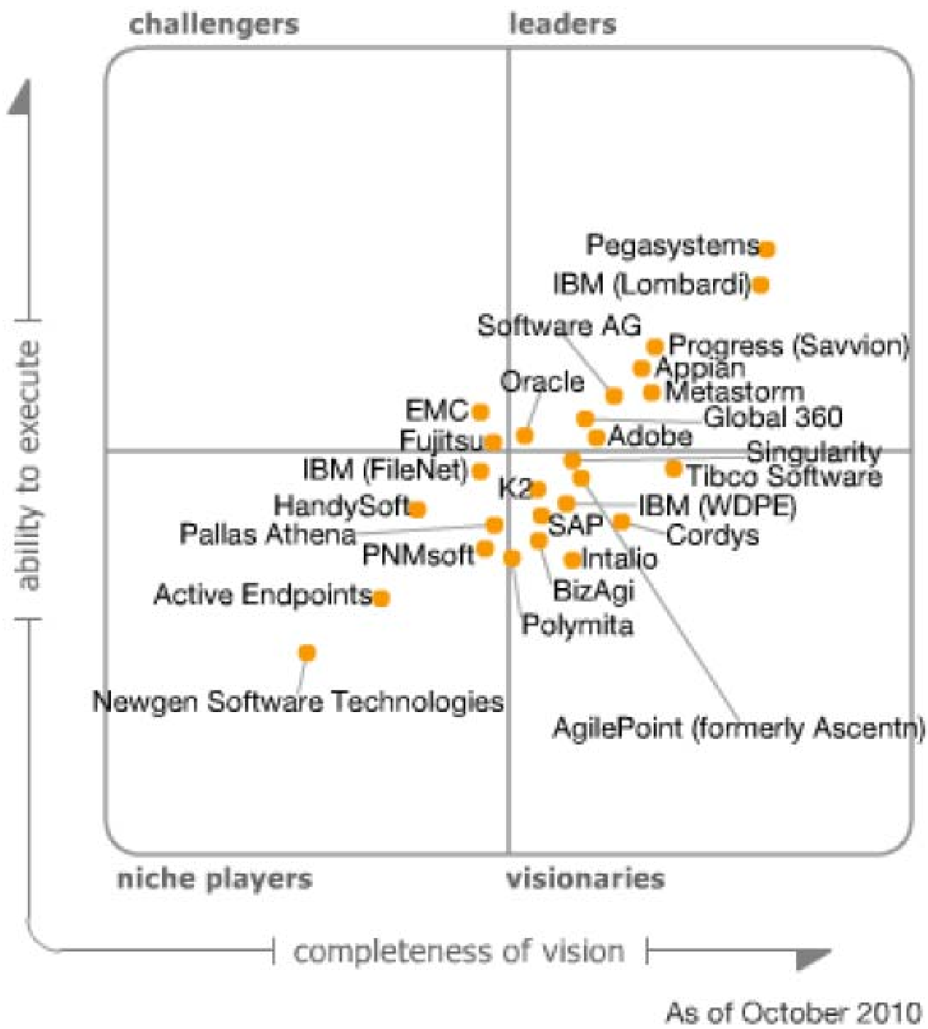
Vision 2020. (2005, March). *Vermijdbare blindheid en slechtziendheid in Nederland - samen werken aan het voorkomen van vermijdbare blindheid en slechtziendheid.* (H. Limburg, Ed.) Retrieved October, 2010, from Vision 2020: http://www.vision2020.nl/contents/V2020NLrapport.pdf

Weber, B., & Reichert, M. (2010). Investigating the effort of using business process management technology: Results from a controlled experiment. *Science of Computer Programming* , 292-310.

White, S. (2004). *Introduction to BPMN.* Retrieved June 2011, from http://www.bpmi.org/downloads/Introduction_to_BPMN89.pdf

WHO. (2011). *Priority Eye Diseases*. Retrieved June 2011, from World Health Organization: http://www.who.int/blindness/causes/priority/en/index1.html

Appendix 1

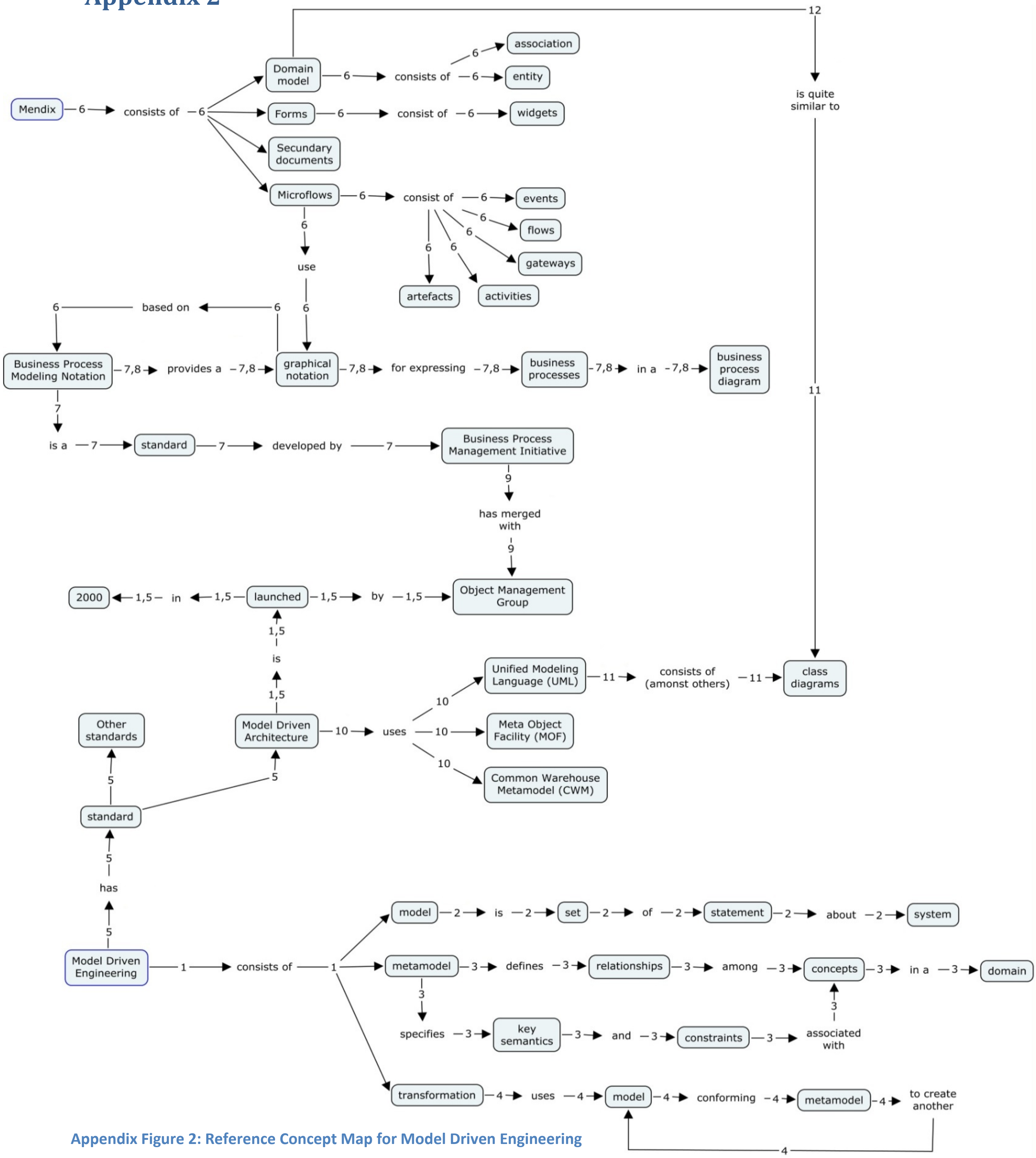


Appendix Figure 1: Magic Quadrant for Business Process Management Suites from (Sinur & Hill, 2010)

Tool	+	-
BPMone Tool from Pallas Athena	• Highly intuitive for business roles due to high leverage of visualization and animation technologies. • Intuitive and unique automated business process discovery capability that contributes significantly to process optimization. • Strong support for process dynamism in production. • Value-based pricing model that appeals to clients.	• BPMone requires unique development and system administration skills. • The modeler is not based on open standards, like BPMN. • The runtime environment is not based on open Web service protocols. • There are few installations beyond the Benelux region.
FileNet Business Process Manager Tool from IBM "FileNet manages workflow among people and systems for content and case-based processes." (IBM, 2011)	• Supports greater change control over process aspects by business roles. • FileNet Business Process Manager emits events. • FileNet offers a very broad and deep set of industry solutions through its partner channel. • Well-proven for content-intensive process interaction patterns, in which changes to one element trigger changes to other elements.	• FileNet is not as intuitive in its ease of development and execution. • FileNet has not maintained a high level of innovation in its BPM features. • FileNet is mostly aimed at content-heavy usage scenarios.
BizFlow BPM Suite Tool from HandySoft "BizFlow is a full featured, dynamic BPM solution platform. Everything needed to collaborative, process-driven application is included." (HandySoft, 2011)	• BizFlow handles unstructured and collaborative processes well. • BizFlow is proved with collaborative human workflows in large-scale, complex installations. • HandySoft has a strong partner channel of value-adding resellers. • HandySoft has attractive prices.	• Most installations are only in government sector. • Low brand recognition as a BPMS provider. • HandySoft's focus on the BPMS horizontal market has wavered in the past years.
Sequence Tool from PNMsoft "Sequence BPM software enables your organization to establish, maintain and continually improve efficient and effective processes for better business performance." (PNMsoft, 2011)	• Provides good support for unstructured and highly collaborative processes. • Highly integrated with Microsoft Office (because of its Business Process Alliance membership with Microsoft). • One of the few to target professional service providers.	• Geographic presence is primarily in Europe. • Low brand recognition and buyer awareness. • Few partners or solution accelerators available.

Appendix Table 1: BPMone competitors comparison (based on Gartner's Magic Quadrant (Sinur & Hill, 2010))

Appendix 2



Appendix Figure 2: Reference Concept Map for Model Driven Engineering

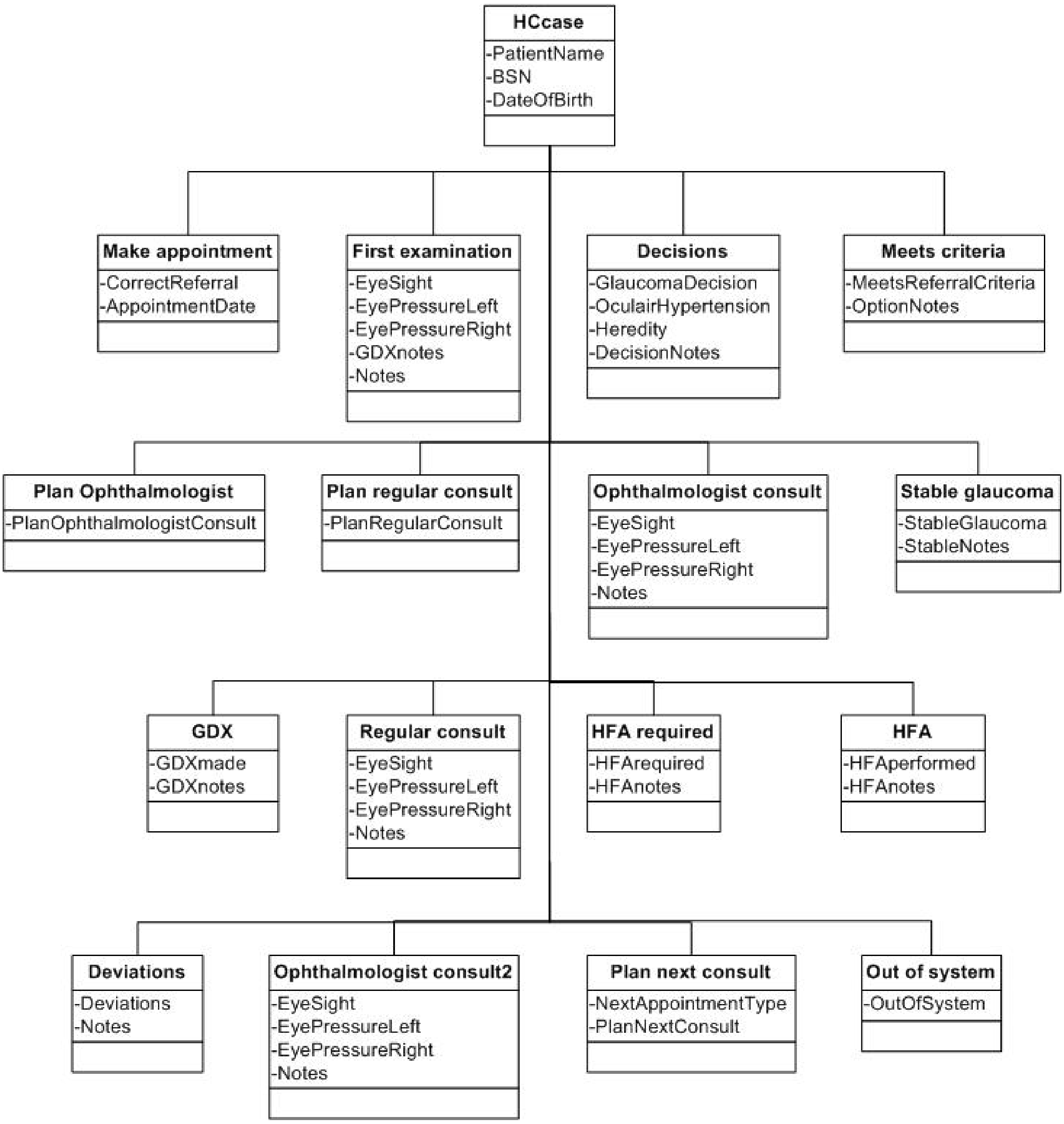
The numbers in the figure represent the references, where 6 represents (Bézivin, 2006), 7 represents (Business Process Management Initiative (BPMI), 2004), 8 represents (White, 2004), 9 represents (OMG, 2008), 10 represents (Truyen, 2006), 11 represents (Fowler, 2004), and 12 represents (Meertens, Iacob, & Nieuwehuis, 2010).

Tool	+	-
Mendix	• Runs on each operating system and works with every internet browser • Mendix App store has lots of custom widgets to enrich the user interface • Makes use of open standards as much as possible • Cloud services • Every change made to the model is immediately reflected in the web application • SAP Certified	• Since Mendix immediately generates a web application, instead of actual code, there is a risk that the modeler pays little attention to version control and release management. • Since there is no code, it is not possible to reuse the models in another tool.
Skelta "Skelta BPM provides a solution for BPM requirements of every class. Skelta BPM is an enterprise wide Business Process Management and Advanced workflow solutions product. It provides enterprises with a strong and collaborative platform to develop a business application, using a seamless and efficient process model." (Skelta, 2011)	• Cloud services • Advanced BPM workflow software functionality • Highly integrated into Microsoft, which gives to opportunity to feed tasks to the user's Outlook, so user friendly • Advanced reporting	• Only aimed at Microsoft (.NET) technology • No use of open standards • Low user experience, no custom apps/widgets to add
Outsystems "Outsystems the Agile Platform is a complete solution to deliver your custom enterprise web apps with high productivity." (Outsystems, 2011) It enables companies to create, modify and maintain entrepreneurial applications that can be changed during each stage of its life cycle.	• Cloud services and mobile apps • Based on .NET and Java • Continuous integration, so after implementation, the processes can immediately be tested by supplier and customer • Sap Certified • Easy access audit trail	• Since Mendix immediately generates a web application, instead of actual code, there is a risk that the modeler pays little attention to version control and release management. • Since there is no code, it is not possible to reuse the models in another tool. • Company originates in Portugal, first Dutch consulting partner in 2010
BizAgi Bizagi Business Process Management solutions makes modeling, executing and improving business processes	• Graphical tool that allows you to easily model business processes with the BPMN notation (Novotný, 2009) • Every change made to the	• BizAgi is functionally less complete compared with other leading pure-play BPMS tools (e.g., rules and simulation/optimization)

easy for everyone, no matter how small or big the company is. To manage the complete process life cycle, the platform consists of three tools; Bizagi Process Modeler (diagram and documentation module), Bizagi Studio (construction module) and Bizagi BPM Server (execution and control module). (Bizagi, 2011)	model is immediately reflected in the web application • Java-based and Windows-based versions • More suitable for beginners and presentations	(Sinur & Hill, 2010) • As is to be expected of a newer product, users report various weak areas, including the forms designer, the integration layer, documentation, and few skilled consultants (Sinur & Hill, 2010) • Does not support simulation, animation or analytics • Less suitable for analytics or larger projects (slower with large number of models

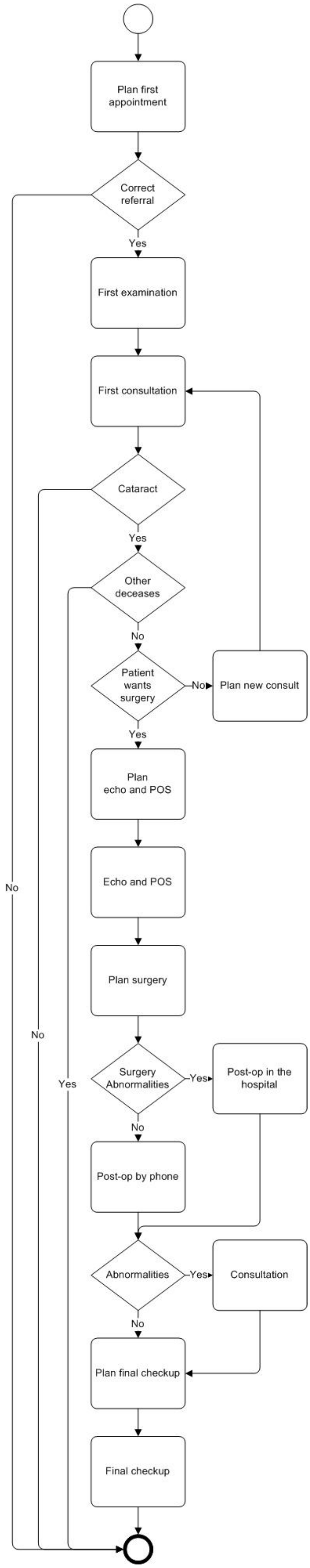
Appendix Table 2: Mendix competitors comparison

Appendix 3

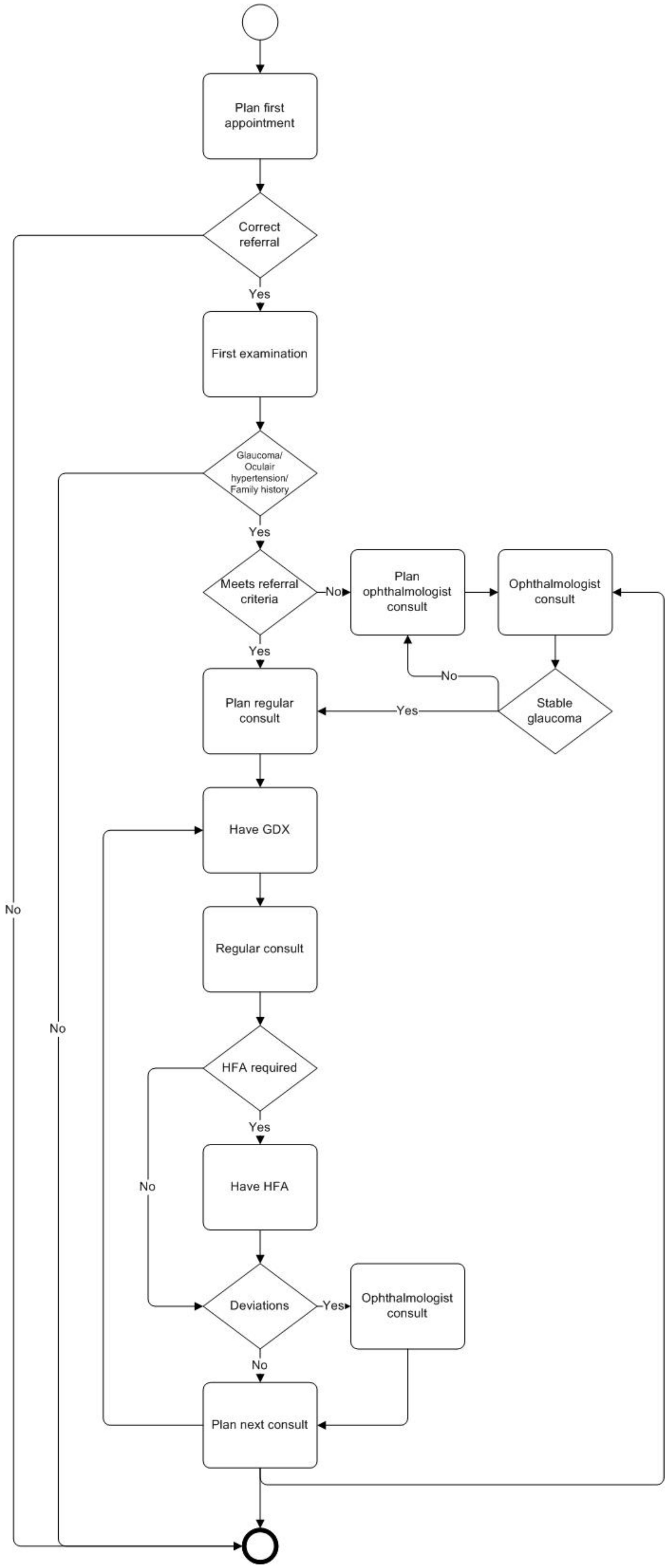


Appendix Figure 3: Class diagram of data requirements

Appendix 4

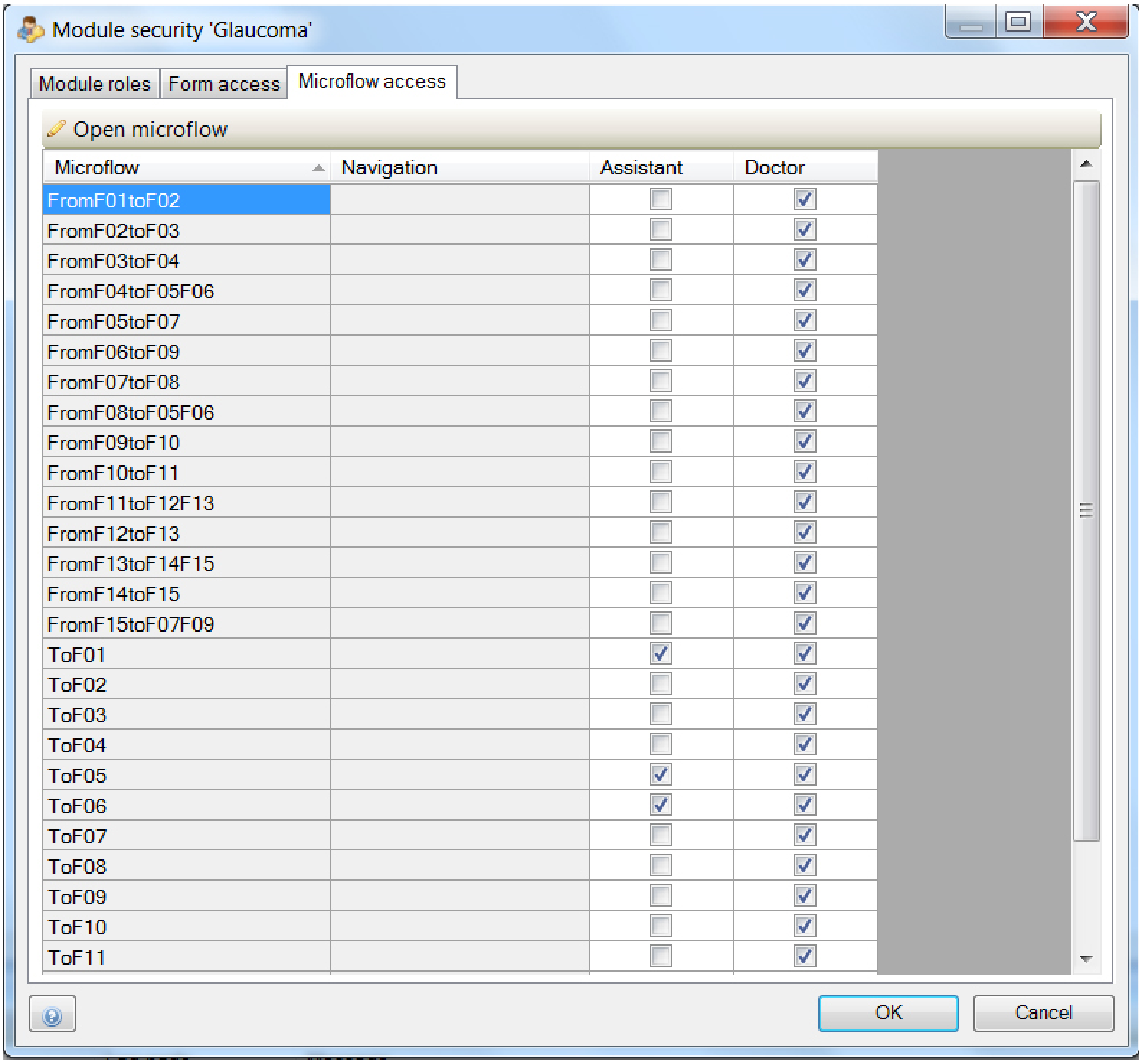


Appendix Figure 4: Cataract input process model



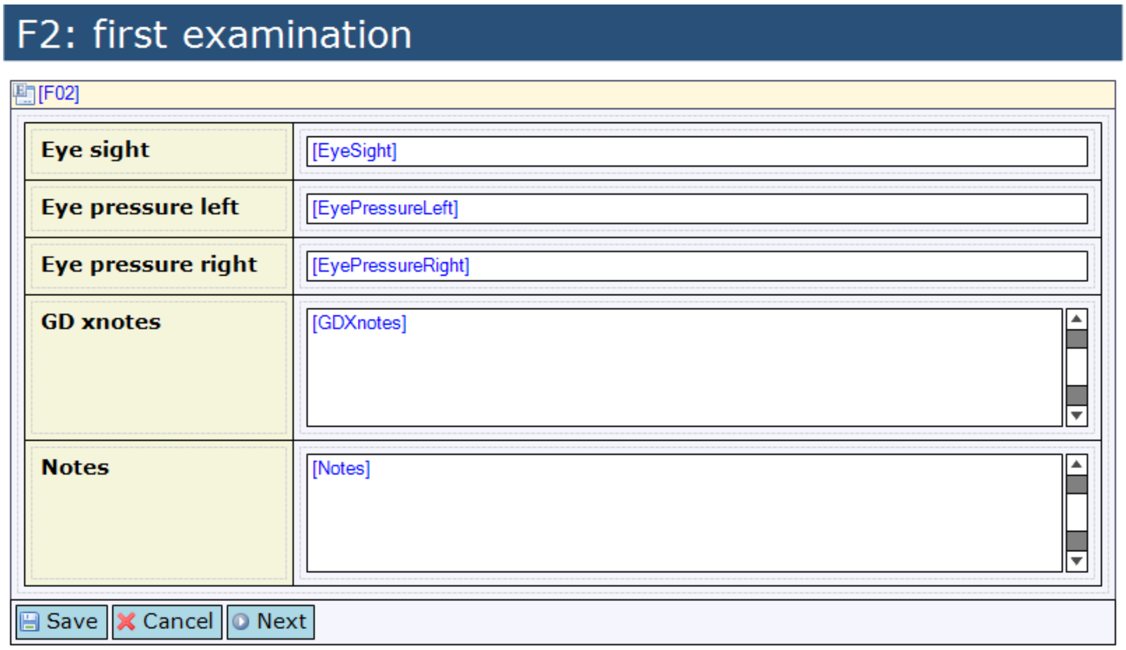
Appendix Figure 5: Glaucoma input

Appendix 5

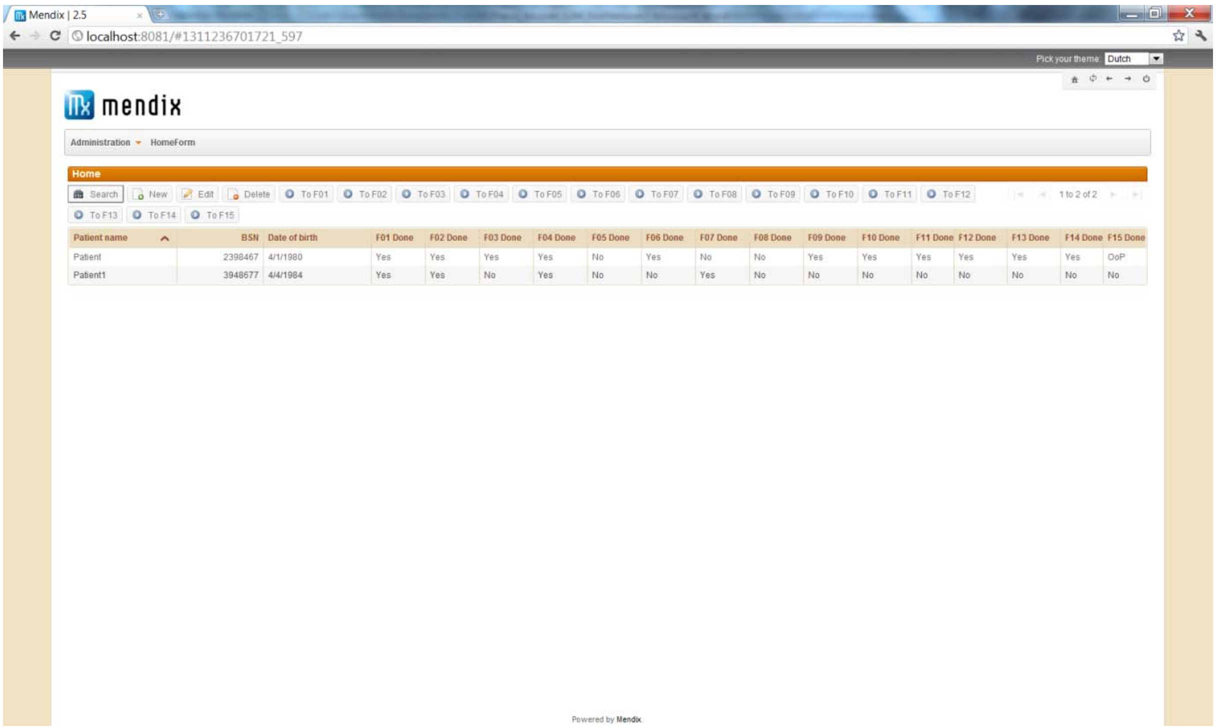


Microflow	Navigation	Assistant	Doctor
FromF01toF02			✓
FromF02toF03			✓
FromF03toF04			✓
FromF04toF05F06			✓
FromF05toF07			✓
FromF06toF09			✓
FromF07toF08			✓
FromF08toF05F06			✓
FromF09toF10			✓
FromF10toF11			✓
FromF11toF12F13			✓
FromF12toF13			✓
FromF13toF14F15			✓
FromF14toF15			✓
FromF15toF07F09			✓
ToF01		✓	✓
ToF02			✓
ToF03			✓
ToF04			✓
ToF05		✓	✓
ToF06		✓	✓
ToF07			✓
ToF08			✓
ToF09			✓
ToF10			✓
ToF11			✓

Appendix Figure 6: Security settings

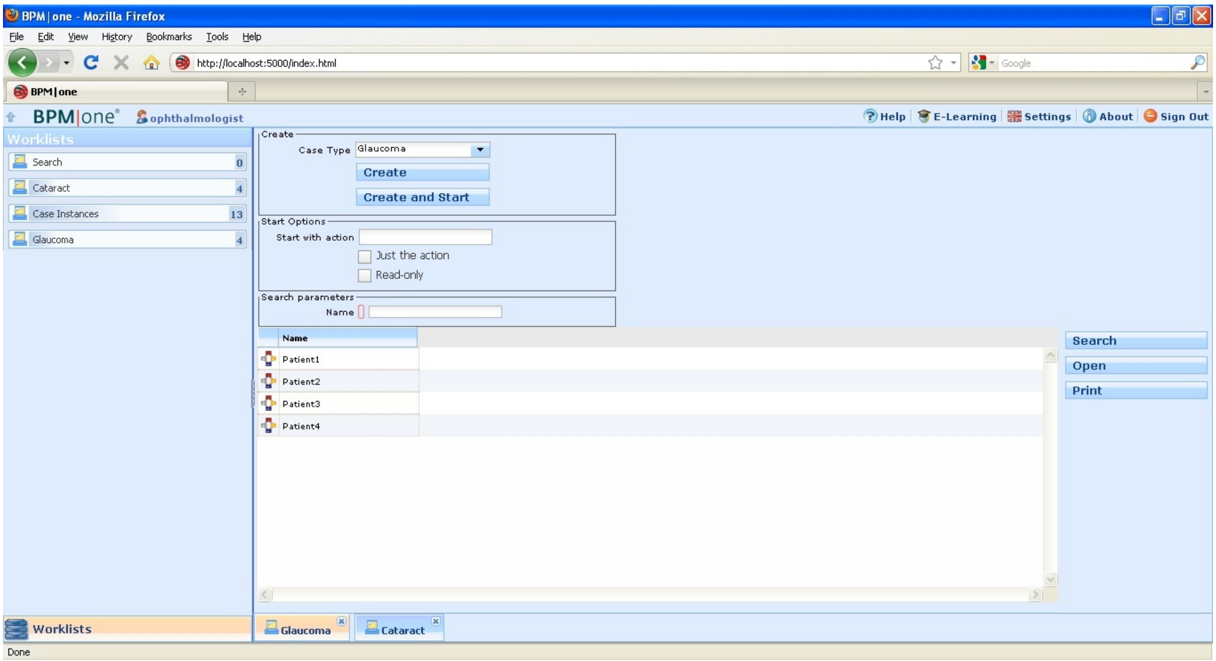


Appendix Figure 7: Example Mendix form from Glaucoma First Version

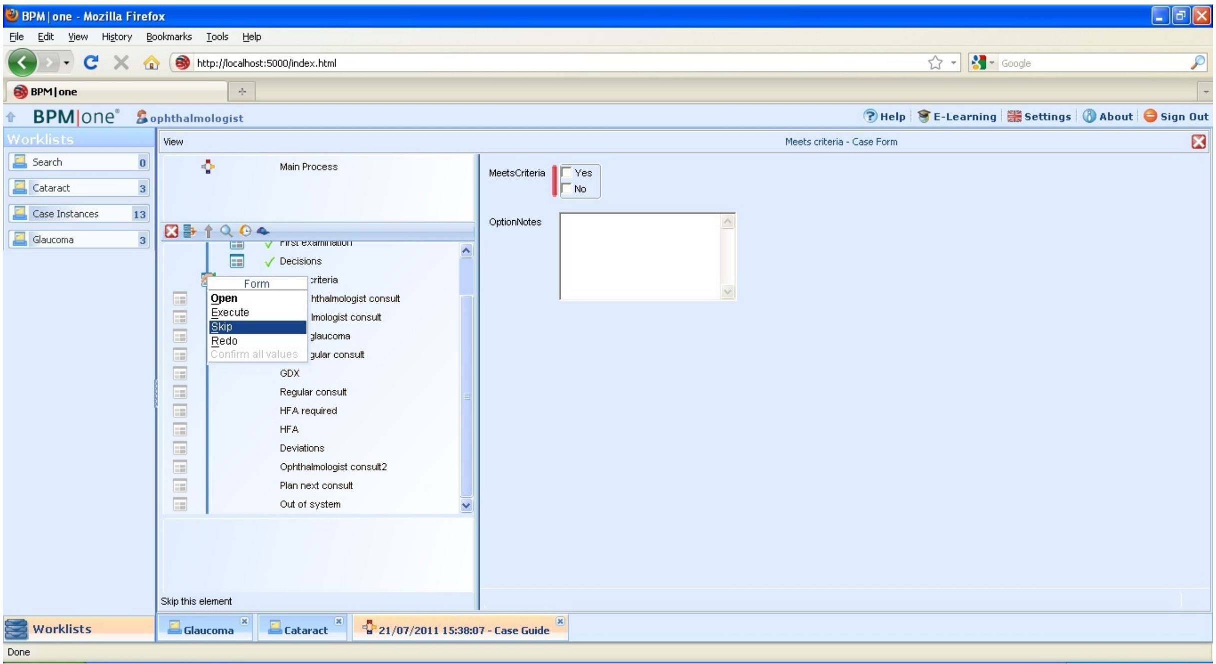


Appendix Figure 8: Mendix home form from Glaucoma First Version

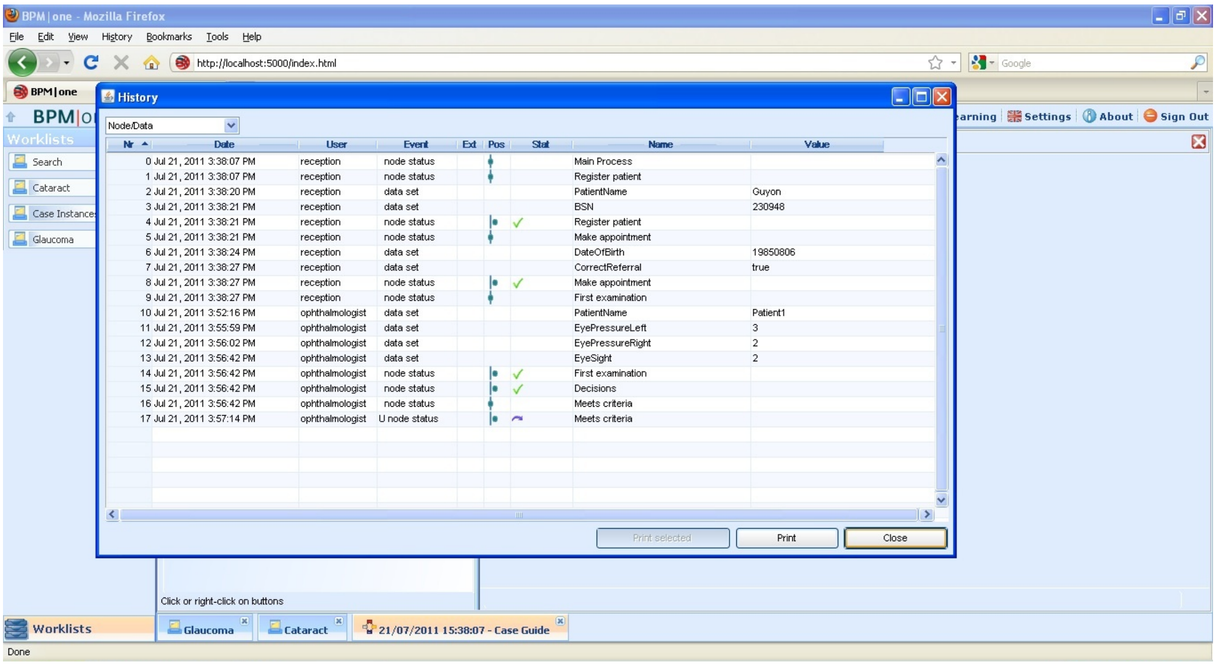
Appendix 6



Appendix Figure 9: BPMone worklists

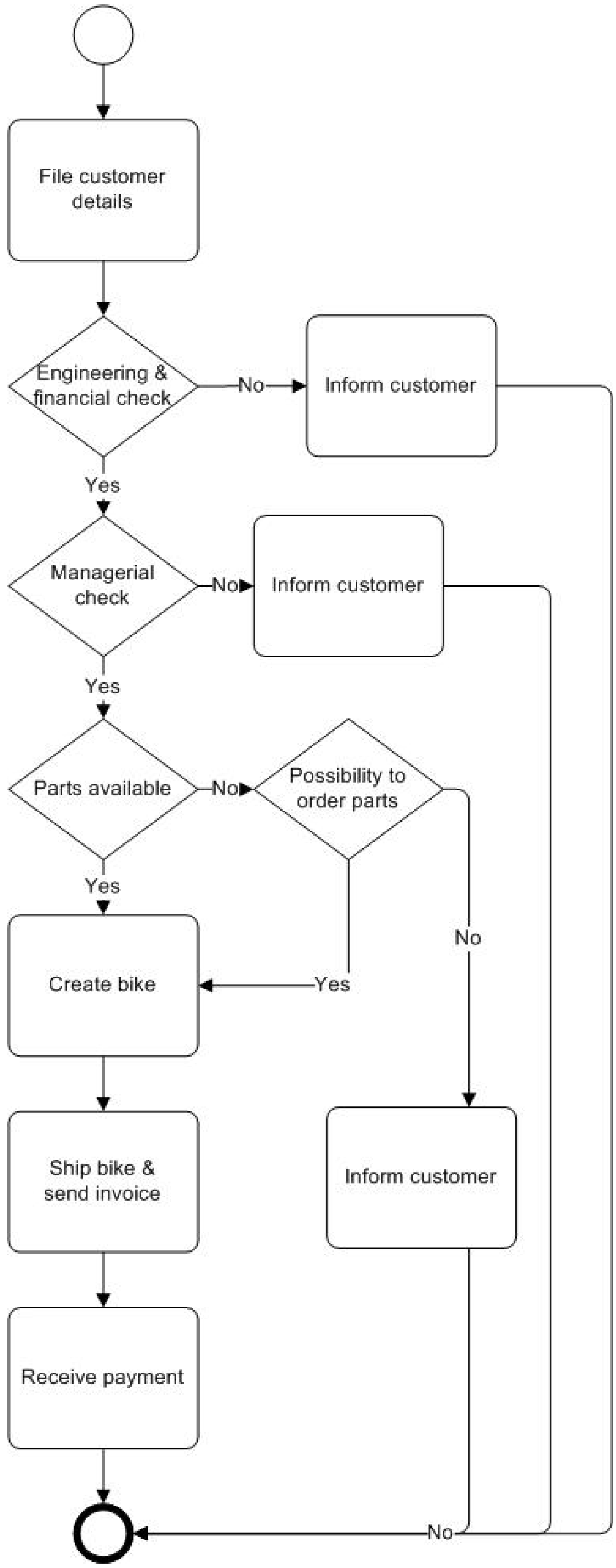


Appendix Figure 10: BPMone skip/redo function; Glaucoma model



Appendix Figure 11: BPMone History; Glaucoma model

Appendix 7



Appendix Figure 12: Bicycle factory

Appendix 8

Appendix Figure 13: Home form for Mendix bicycle factory model