Facilitating model-driven infomation intensive business process applications a framework

van Lee, A.M.

Award date:
2012

Eindhoven, August 2012

Facilitating model-driven information intensive business process applications: A framework

by

A.M. van Lee

BSc Industrial Engineering and Management Sciences — TU/e 2011
Student identity number 0614705

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
Dr. M. Comuzzi, TU/e, IS
Dr.ir. H.A. Reijers, TU/e, IS
Ir. E. Sabbah, O&i

TUE. School of Industrial Engineering.
Series Master Theses Operations Management and Logistics

Subject headings: information intensive processes, application creation, EAD method

I. Preface

This document contains the master thesis report. The master thesis is the final project for the study Operations Management and Logistics at Eindhoven University of Technology.

Dr. Marco Comuzzi was my first supervisor from the university, dr. ir. Hajo Reijers was the second supervisor. This thesis is written in cooperation with O&i Partners in Business Agility, particularly ir. Eyad Sabbah, senior managing consultant at O&i, who acted as my mentor.

The objective of this project is twofold, first to make a contribution to scientific literature, and second to gain new insights that can be applied in practice right away. Both goals will be satisfied since the created EAD method will be new in literature and be interesting to today's practice. The cooperation with the Dutch consultancy firm O&i ensured practical relevance, the Eindhoven University of Technology guarded the scientific soundness.

The environment and colleagues at O&i were very pleasant. I have enjoyed working there on my thesis. During the process of creating a method I have learned a lot, which will be helpful to me in my professional career. I hope someone will continue the work I have started with ir. Sabbah and dr. Comuzzi

I would like to thank Marco and Eyad for their effort and guidance during my research. I have learnt a lot in the past six months, just as during the past six years I spent studying. I would also like to thank my mother for her insights, just as the rest of my family and friends for their support and help. It was really appreciated.

Arthur van Lee

Augustus 2012

II. Management summary

How to increase productivity in a production environment has been studied for a long time. It all started with the industrial revolution of the 18th century. By 1914 Henry Ford started to build cars on a moving assembly line. During the last century a lot of effort has gone into further optimizing production processes. In contrast, less research is done on optimizing information intensive processes. Information intensive processes are prominent in service oriented business, which started to grow after World War Two. It can be stated that the research field of information intensive processes is still in its infancy, what leads to the fact that improvements of these processes can still enhance major benefits.

One of the possibilities a company has to improve its information intensive processes is to make an application that supports the process. With a proper application the execution of the process can go faster and with fewer errors, which will increase the profit margin. Often a business consultant is asked to create this application since not every business has the knowledge to create such an application themselves. But still the actual creation of an application is so difficult that a business consultant either needs special training or asks a programmer to create what the business consultant envisions. We want to create a method that allows business consultants to create applications without receiving training or the help of a programmer. The Easy Application Development (EAD) method is presented to do just this.

The creation of a method requires several phases: specification, development, validation, and evolution. This research is focused on the specification phase, indicating what the EAD method is, to what purpose it can be used, what the different elements and their relations are, and how they should work. A vision captures the long term objective of the EAD method:

The EAD method allows consultants to model an information intensive business process and use the created model to automatically generate process supporting applications.

This EAD method consists of the following elements:

- **EAD application**: an application that can be used to create a model of an information intensive process by making use of standard elements.
- **Meta-model**: a set of standard elements that can be used to create an instance of an information intensive process.
- **Well-formedness rules (WFR)**: a set of well-formedness rules helps the user to make a valid platform independent model (PIM).
- **Transformation rules**: the application of the EAD method is able to transform the created model into a platform specific model (PSM). This transformation will be done on the basis of a set of rules. Specifically, we will focus on a PSM for two tools that can be used to generate applications that support information intensive processes, namely Be Informed and Mendix. This in turn will result in a process supporting application for the information intensive process.

Since this thesis has a focus upon the specification phase the EAD application as such will not be created. The application is a piece of software based upon the other three elements. Therefore these other elements are specified in this thesis. The EAD method is based upon the Model Driven Architecture (MDA) which is developed by the Object Management Group (OMG) (Object Management Group, 2003). OMG aims to create standards in the information systems field; by making use of these standards the EAD method will be compliant with other software. Figure A captures the different elements of the EAD method; one could say that the meta-model is a set of words and the WFR is the grammar. Together they can create a meaningful instance (PIM). Since eventually the EAD application will automatically (the business consultant will not know what exactly happens) transform the PIM into a PSM, the term black box is used to indicate the transformation. For every software tool specific transformation rules have to be created. The rules for Mendix and Be Informed are described. Next the elements meta-model, well-formedness rules, and transformation rules of the EAD method are discussed.

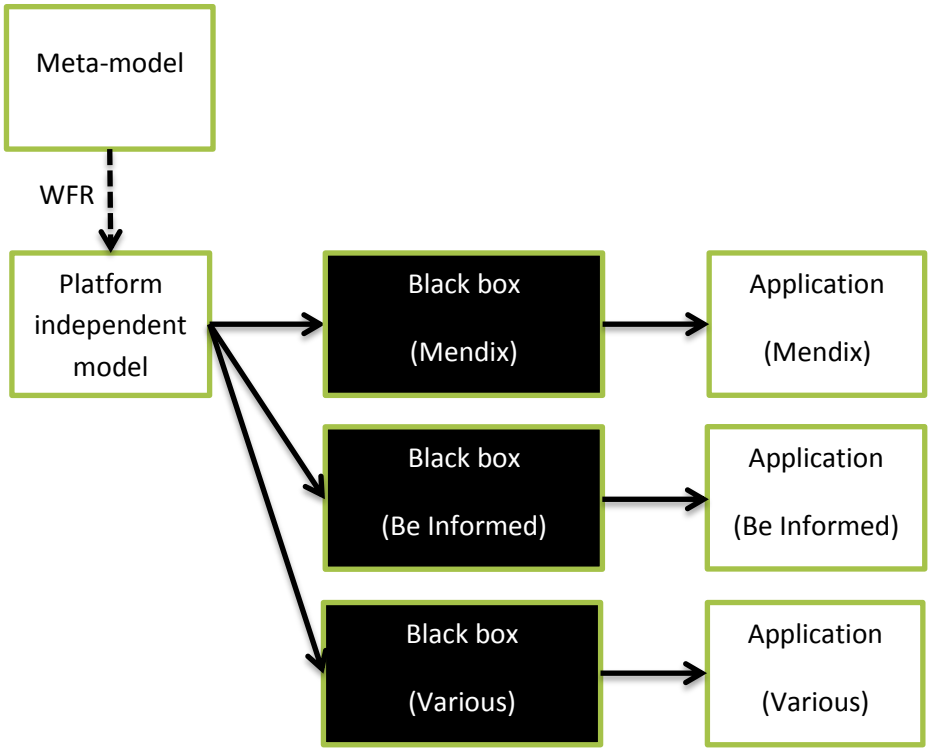


Figure A elements of the EAD method

Meta-model

The meta-model is created after reviewing scientific artifact centric workflow models, taking interviews with experts, and analyzing the structure of Mendix and Be Informed. The result after several iterations is a meta-model where elements are related to each other in UML style, as is compliant with the MDA.

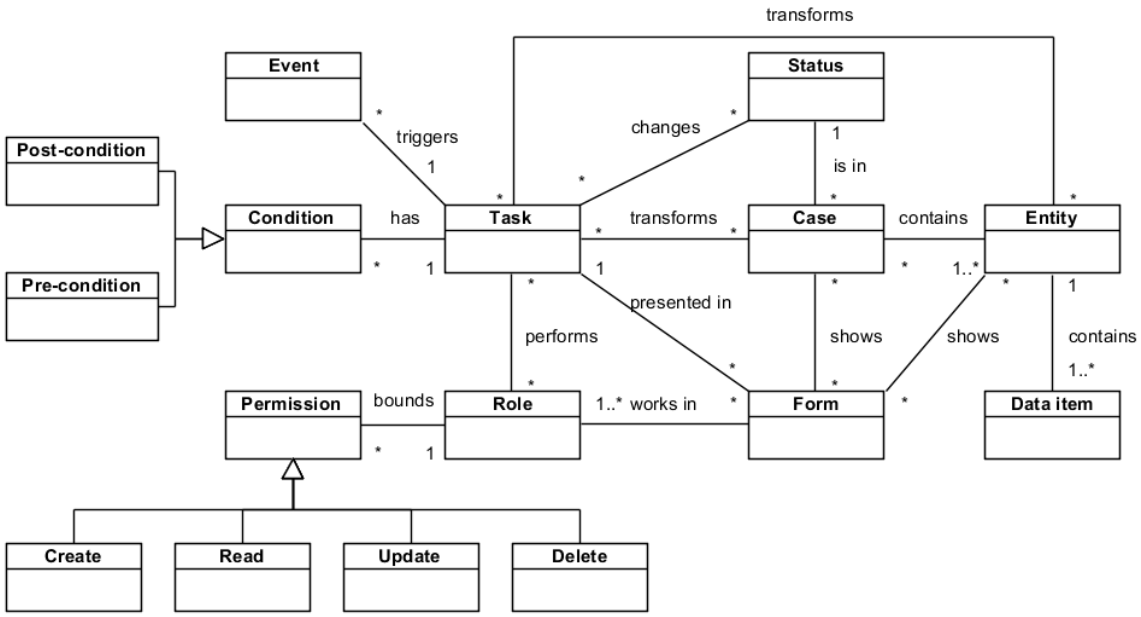


Figure B meta-model of EAD method

Well-formedness rules

After a basic set-up the WFR were improved by discussing them with experts.

1. The instance must be created in the UML format as prescribed by the OMG.
2. The associations that are set in the meta-model must be respected. For instance an entity can contain one to many data items and a data item can be part of only one entity.
3. For each element a name is created stating what it represents. For example status: 'processed', and status: 'handled'.
4. If a task is performed, the information about the execution is stored automatically. One can think of information about the employee who executed the task at what time.
5. Conditions can only refer to elements that are already described in another section of the model.
6. Conditions can refer to elements they are not directly linked to.
7. A data item can be of any type like 'string' or 'integer'.
8. Within each element it is possible to write further instructions in text.
9. It is not obligatory to make use of all elements when modeling a process.

To create an instance based on the meta-model you can start with any element. But obviously if you want to refer to another element, this element needs to be created first. You will see, for example, that in practice permissions and conditions can only be created when all other elements have been created since they highly depend on other elements.

Transformation rules

A set of transformation rules is presented for Mendix and Be Informed. These will be used to transform the PIM into one of the two software tools. The rules can be found in chapter 7. After writing the rules, they were verified by experts. Within the transformation rules two types are differentiated; basic rules and customization rules. This is done because in the two software tools customization of the applications is complex. It would be impossible to generate transformation rules that can transform any PIM to a functioning PSM at this stage of the development of the EAD method. The basic rules allow for transformation of a basic application that can be customized by hand. Customization rules are not describes since this list of rules would grow very long. When the EAD method is in the development stage these rules can be further developed. Possibly transformation rules are first created for standard problems, and later for more complex problems. Think of digital photo editing, a few years ago only red-eye reduction was simple, nowadays a lot of other transformations are possible to execute by non-photo editing experts.

The presented elements form a solid specification for further development and validation of the EAD method. However verification of the results could get more attention. The meta-model is applied to a couple of simple process descriptions. More trials in using the meta-model for making an instance would lead to new insights. The same accounts for the well-formedness rules and the transformation rules. Further research is needed to develop, validate, and market the EAD method.

List of tables

List of figures

Index

Part 1: Research definition

1 Introduction

How to increase productivity in a production environment has been studied for a long time. It all started with the industrial revolution of the 18th century. By 1914 Henry Ford started to build cars on a moving assembly line. During the last century a lot of effort has gone into further optimizing productions. From the seventies on, with the introduction of computers, businesses have been investing more and more in the management of workflows by means of software tools. First there was MRP (Material Requirements Planning), MRP II (Manufacturing Recourse Planning), and in the nineties ERP (Enterprise Resource Planning) arose. Successful WfMS deployment results in significant process cycle time reductions, cost reductions, improved accuracy, greater control, and greater worker satisfaction (Stohr & Zhao, 2001). The introduction of workflow management systems can result in productivity gains from process automation of 5% to 30% and cycle time reductions of 30% to 80% (Ader, 2000). Successful workflow projects met or exceeded ROI expectations approximately 89% of the time (Stohr & Zhao, 2001). Today the WfMS is finding its way to the internet, for instance with SaaS (Software as a Service), developments where the main focus is to deliver workflow management software in the cloud. The conclusion is that businesses obviously can gain from WfMS. At the same time it has become clear that successful implementation is complex and there is an urgent need for further research on ways to increase implementation success. One can think of investigating the impacts of workflow automation tools on humans, on the nature of work, on appropriate organizational structures, and on support for non-routine work (Stohr & Zhao, 2001).

One particular field of interest concerns businesses with information intensive processes, like insurance companies, banks, or internet businesses. These processes are highly represented in the service industry, or tertiary sector of the economy. The tertiary sector is still a young but fast growing sector; it only started to grow after the Second World War. For the last 30 years in the western world more than half of the GDP has come from the service sector, indicating that it is actually the major sector of the western world. It is a sector where improvements can have major benefits, however research in this young field is still in its infancy.

Nowadays business consultants are active to help improve productivity by creating software applications for executing information intensive processes. Some software toolss are available, but on the whole their work is complicated and for each assignment they have to start from scratch. A method is needed to facilitate the creation of these software applications, which will improve quality and creation speed of these applications for consultants, without receiving training for a specific software tools.

This research is aimed to set the path and take the first steps towards the development of such a method. Thereby this research may contribute to making improvements in productivity in service oriented businesses or any other businesses with information intensive processes.

1.1 Structure

In the first part of this thesis a research definition is presented, consisting of the objectives and the placement in the scientific literature. The objective in chapter 2 is to present a vision where this research attempts to lead to, also indicating to what extent this research contributes in meeting the stated vision. Chapter 3 ends the research definition part by defining some important terms that are at the basis of this thesis.

In part 2 the research design is presented, starting with the research questions in chapter 4, followed by the methodology in chapter 5. The research questions are answered in part 3 where the first steps are taken on the path towards the stated vision. In part 4, by the means of an example, the potential of the results are validated and proven. The thesis ends with a discussion and conclusion in part 5.

2 Objective

This chapter outlines the objectives of this thesis in the long term (vision) and the short term (goals).

2.1 Vision

What we want to accomplish in the long term is that a business consultant can develop applications that will allow more efficient process execution without following any training for a specific software tools. The model depicted in figure 1 shows three levels of abstraction. The customer in the first level has a business problem, either a business process needs to be improved, or a new information system to support the process needs to be created. The consultant in the second level has the ability to capture and describe the process in a process model. But most consultants are unable to create a process supporting application on the basis of the process model they have created. Therefore in level 3 of the model on the left we see a programmer that programs the application. By creating a meta-model - a model that defines the language for expressing a model (Object Management Group, 2003) - of the artifacts required by the workflow support application (figure 2), the programmer can be left out of scope. Now the consultant will be able to first model the process and then generate the supporting application automatically. The major advantage of this would be that one translation between the two levels is automated (level 2 to level 3), increasing the success rate of the project. First of all a translation layer is deleted between the consultant and a programmer, reducing translation errors. Secondly the development time for applications will be shortened, allowing faster returns on the investment of having the application built.

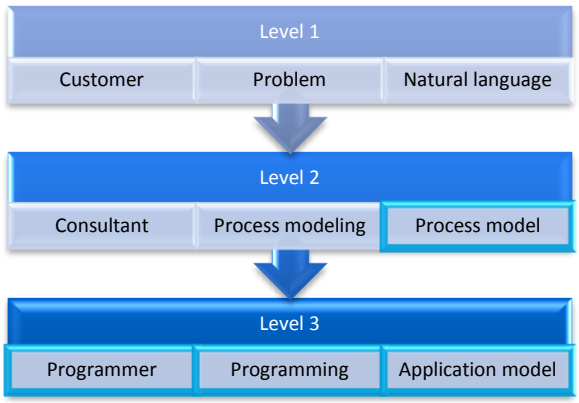


Figure 2 research objective current situation

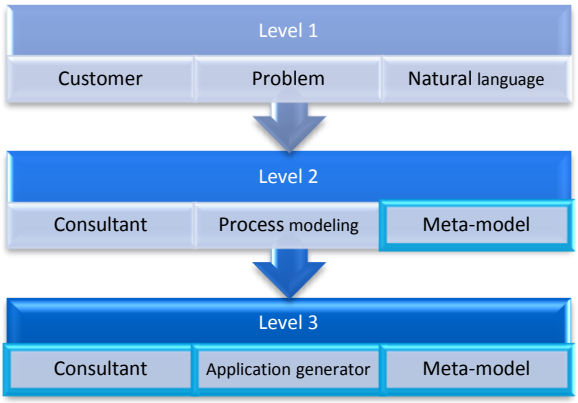


Figure 1 research objective to-be situation

The way to achieve this vision is to create a method which can be used by any consultant. This method must thus be simple to use therefore the method is called Easy Application Development (EAD method).

This EAD method consists of the following elements:

- **EAD application**: an application that can be used to create a model of an information intensive process by making use of standard elements.
- **Meta-model**: a set of standard elements that can be used to create an instance of an information intensive process.
- **Well-formedness rules (WFR)**: a set of well-formedness rules helps the user to make a valid platform independent model (PIM).
- **Transformation rules**: the application of the EAD method is able to transform the created model into a platform specific model (PSM). This transformation will be done on the basis of a set of rules. Specifically, we will focus on a PSM for two tools that can be used to generate applications that support information intensive processes, namely Be Informed and Mendix. This in turn will result in a process supporting application for the information intensive process.

The software that supports the EAD method will allow the consultant to create a process model that is independent of the preferred software tools. The transformation towards the preferred software tools is done automatically. The major advantage is that a consultant does not have to learn how to build an application in each different software tool or needs transfer the model to a programmer; instead he/she learns once how to create models by means of the EAD method.

In short the vision can be stated as follows:

The EAD method allows consultants to model an information intensive business process and use the created model to automatically generate process supporting applications.

2.2 Goals

The previous chapter made clear what the long term vision is. This chapter will clarify what steps are to be taken towards this vision and what short term goals for this thesis will be defined.
If you compare the development of a method with the development of a car, several steps can be identified for both developments.
Generic activities in all software developments are (Sommerville, 2004):

1) **Specification** - describing what the system should do and its development constraints
2) **Development** - production of the software tool
3) **Validation** - checking that the software is what the customer wants
4) **Evolution** - changing the software in response to changing demands

It all starts with the idea that something new will have a market potential. For example an electric sports car that can carry four people. For the EAD method a method that can be used by consultants to directly create an application as mentioned in the introduction there is market potential for this idea. If one decides to make a development out of this idea, the specification phase starts. For the car a description is made how all parts should work together, possibly a computer model is created. For instance, operation of the car must be possible for anyone with

a driving license. How the electronics make that happen is not interesting for the user. For the EAD method the specification phase consists out of describing what elements are required and how they interact with each other.
Next the development phase can start, in this phase all details are worked out, the development must be in such a state that it can be used. In the case of the electronic sports car, it must be prepared for going to production. However before the EAD method is used, it is first validated in a validation phase. After executing the validation the EAD method can be launched and the evolution phase starts. During this phase changing demands can cause a need to change the method.

This thesis will focus upon the first step, the specification. The focus will be on how the mentioned application of the EAD method should operate and which elements are needed, not creating this application. In some cases initial steps will be taken to show that what is proposed is possible. Development and later on the validation are beyond the scope of this thesis. These steps have to be taken by researchers in the future. Possibly each stage is a full master thesis project. The goals of the specification phase, and thus of this thesis are:

- Defining the meta-model for the EAD method
- Specifying the well-formedness rules
- Initiating transformation rules for a PIM to a PSM

The main part of this thesis is part 3 where the first steps for the development of the EAD method are taken. Figure 3 is introduced to clarify how each chapter contributes to this thesis. This figure will be used throughout to pinpoint where the reader is. In chapter 4, the research questions, the figure is clarified in more detail.

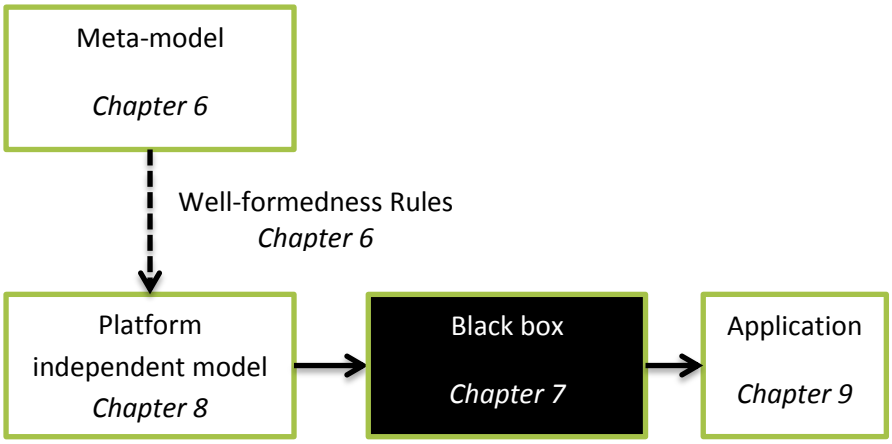


Figure 3 overview of thesis research

The third part starts with chapter 6, stating what elements are used to formulate the meta-model. The well-formedness rules of chapter 6.4 will tell you how to use the elements of the meta-model for the modeling of an information intensive business process. The next chapter, chapter 7 describes how an information intensive process created with the meta-model can be mapped in either Mendix or Be Informed. Since the end user will be unaware how this transformation is made, this step is referred to as the black box. In part 4 an example is used to clarify how all the parts described in this research translate to practice. Chapter 8 is an example of a platform independent model and chapter 9 will indicate how this will work out in Mendix and Be Informed. So first the deliverables of the EAD method are presented, followed by the validation. The PIM is a result of the meta-model and transformation rules, but is discussed after the third deliverable of the EAD method in the validation part.

2.3 Software tools

This thesis will show how the EAD method can be used to create applications. For the creation of these applications software tools are used that are already on the market today. Hence, the vision is that consultants do not have to learn how to create an application with each individual software tool. In time all different software tools can be used in this EAD method, but during this research the focus is upon two different existing tools: Mendix and Be Informed. Two tools are selected to show that the proposed EAD method will work with more than one application. These two specific tools are selected for reasons of practical accessibility. Amongst comparable software tools, each having its own strengths and weaknesses, that are excluded from this research are BizAgi, Oracle BPM suite, BizFlow Plus, and Cordys. Some background information on the selected systems can be found in the next chapter.

3 Terms and Definitions

This chapter gives an overview of the most important terms for this research. Definitions are stated which relate the term to information intensive processes in order to make clear what is exactly meant in perspective of this research. The following terms are discussed: agility, and artifact centric workflow. In general the terms will be introduced by presenting a number of accepted definitions and an example.
Furthermore some information is provided on Mendix and Be Informed, when those business where founded and what their main focus is. This chapter is ended with an initial comparison between these two systems.

3.1 Agility

The term agility is similar to the term flexibility. In natural language it means the ability to move quickly and easily or to think quickly and clearly. In the Information System (IS) domain Sarkis (2001) defined agility in the following way: "*Agility is the ability to thrive in an environment of continuous and often unanticipated change."* Burmeister, Arnold, Copaciu, and Rimassa (2008) give a definition for an agile process: "*An agile process should be able to pro-actively adapt itself quickly to a changing environment."* Reasons for change can be shifts in regulations, customer demands, or entrance of a competitor. A third definition comes from Tsourveloudis and Valavanis (2002): *"the ability of an enterprise to operate profitably in a rapidly changing and continuously fragmenting global market environment by producing high-quality, high-performance, customer-configured goods and services."* It is the result of a product of human abilities, skills, and motivations, and also advanced organizational and managerial structure and practice. Motivation is one of the main differences between agility and flexibility in the business context (Tsourveloudis & Valavanis, 2002). A more recent definition comes from Ross (2012): *"Being able to deploy change in business policies and business rules into day-to-day business activity as fast as business people and Business Analysts can determine the full business impact of the change and assess whether the change makes good business sense"* (Ross, 2012).

Sense and respond is a short way of saying that the company needs to be able to see change coming and to respond accordingly. If both happen in an effective way for the organization as a whole, one could say that the business thrives on change and demonstrates to be agile. Changes in the environment, e.g. a new law, can have a huge impact on how the final result can be obtained. For example for a courier, when a current law prescribes that no tax has to be paid on delivery cost, and this law changes so that one has to, then the processes need to be adapted. The agile response of a courier would not only make them quickly adapt to the new situation, but it would cause the courier to experience a growth in market share since they adapted faster than competitors.

With the aforementioned information we come to the following definition: *process agility is the ability of a process to thrive in a changing environment.* The main difference with flexibility is thus not only having the ability to *react aptly* to the changing environment, but to *thrive* in it.

3.2 Artifact centric workflow

In order to come to a definition for artifact centric workflow, we will first introduce workflow as such. "*A workflow is the automation of a business process in whole or in part, during which documents, information or tasks are passed from one participant to another according to a set of procedural rules. In other words, a workflow is a specific kind of process, whose transitions between activities are controlled by an information system, the WfMS"* (Stohr & Zhao, 2001). The workflow is typically based upon the process, but as we will see in this chapter it can also be based upon an artifact.

Now we can take a closer look to what artifact centric workflow is. There are four key elements in an artifact-centric workflow model: business artifact information model, business artifact macro-level lifecycle, services (tasks), and the association of services to business artifacts (Bhattacharya, Hull, & Su, 2009). There are a couple of important benefits of using artifact centric workflow design (Garcia, 2011).

Understanding and communication

Artifacts are basic blocks that enable a natural communication and intuitive way of thinking of business people about their operations (Nigam & Caswell, 2003; Cohn & Hull, 2009). Artifact centric models use a minimum of business entities, reducing their size and complexity (Garcia, 2011). The time and staff needed to do business transformations reduced measurably.

Reusability/maintainability

The design of artifact centric workflows allows change at conceptual and workflow levels while preserving the same operations model (Bhattacharya et al., 2009). The establishment of a library of artifact services enables their reuse and workflow composition (Hull, 2008). A generic workflow schema is defined and it can have multiple variations (e.g. in sequencing of services) to adapt to special circumstances.

Execution

Despite the fact that artifact-centric models are expressed in a way that is intuitive for business people, they include the formality needed for their automated implementation in a variety of tools (Cohn & Hull, 2009).

Different specific techniques are possible when conducting artifact centric workflow modeling, for instance the BALSA, FlexConnect, Product based workflow design, XFlow, and $\text{TriGS}_{\text{flow}}$.

3.3 Mendix

"The Mendix App Platform enables companies to quickly build web and mobile apps, integrate them seamlessly, deploy them instantly, manage them centrally, and maintain them effortlessly on an ongoing basis."[1]

When in 2005 Mendix was founded, they introduced their Mendix Agile Business Platform. This platform allows users to create applications that are integrated with existing systems. They

claim that the creation of such applications takes only a fraction of the usual time. *"This means less risk, more productivity, and a cutting edge way to respond to new business needs in days – not months."*[1] By combining the best of the existing systems and easily adding custom web applications the agility of the business is increased. Systems Mendix can be integrated with are for instance SAP, Microsoft Dynamics, IBM, Oracle, and Siebel.

Mendix has five general phases in order to develop an application: collaborate, design & build, deploy, operate, and iterate. So beside the standard design & build, deploy and operate steps Mendix also mentions collaborate and iterate. With collaboration they mean that project stakeholders have social productivity features like activity streams, agile project management and chat functionality. Mendix has a strong focus on feedback, in the iteration phase feedback is gathered form the users and managed within the system so that problems can be classified and addressed at once.

The vision of Mendix is: "*Mendix is the app platform company for the enterprise. We enable companies to build, integrate and deploy web and mobile applications faster and with better results, effectively driving ROI in days not months".*

For this research Mendix version 3.2.1 is used.

3.4 Be Informed

Be Informed is an internationally operating software supplier, founded in 2003. The platform they have created is aimed at administrative processes that become more and more knowledge-intensive. The vision of Be Informed includes a platform where not only the data, but also the logic can be managed separately. In most systems today the business logic is mixed with the application code, which makes it very difficult to change the business logic. By separating the logic from the code it becomes very easy to respond to changes in the environment of the business and thus business agility is increased. Since Be Informed is based on open standards, connection with other business applications or legacy environments is easy. "*In other words, business is in charge of agility, IT is in charge of stability."*[2]

Be Informed believes that the solution is not to reduce complexity, but to embrace complexity. When an organization can easily adapt to change in the sense that processes can be altered and modified, it will improve the power of the business. Be Informed believes that there is a paradigm shift from process centric to case centric. Elements of case centric are: every case is unique, mass customization, goal-driven, outside-in, and information-driven.

The vision of Be Informed is: *"With Be Informed, organizations can embrace complexity thanks to the clear separation of business logic and application code. With its model-driven approach, Be Informed is above-the-line technology, but made enterprise class."*

[1] www.mendix.com
[2] www.beinformed.com

During this research version 3.10.5 of Be Informed is used.

3.5 Mendix versus Be Informed

In its presentation Be Informed has a focus upon case management, instead of the traditional process focus. Mendix is more focused on the presentation and the fast development of the application. However both systems can create in a relative short time powerful applications. The synchronization with legacy systems is something that both software tools claim to support. The focus of Mendix on presentation and ease of use limits the complexity that can be handled with the system. Complexity is better dealt with by Be Informed. Since Mendix operates in the cloud, the system has less impact on existing infrastructures. This contributes to the fact that Mendix is much cheaper for a business to use, it is estimated that the Be Informed is a factor 20 to 40 more expensive as Mendix. As a preliminary conclusion one could say that for a fast and user friendly solution of a relative simple business problem Mendix is the best solution, and for a more complex process Be Informed is better suited for the job.

Part 2: Research design

This chapter will give a description what road is chosen to get to the vision: *The EAD method allows consultants to model an information intensive business process and use the created model to automatically generate process supporting applications.* Only the first steps on this road are taken during this thesis research, e.g. the steps of the specification phase.

4 Research questions

With the vision and goals in mind, a number of questions arise concerning how to reach these goals. The general method that is applied in setting the route towards the vision comes from the Object Management Group (OMG). This group aims to support development in the information systems field by posing standards. They capture their goals in a mission statement:

"OMG's mission is to develop, with our worldwide membership, enterprise integration standards that provide real-world value. OMG is also dedicated to bringing together end-users, government agencies, universities and research institutions in our communities of practice to share experiences in transitioning to new management and technology approaches like Cloud Computing."[3]

One of the areas OMG has created a standard for is IS modeling, the Unified Modeling Language (UML) (Object Management Group, 2011). UML is used by many consultants to create models that can be interpreted by others (humans and computers). In this area Model Driven Architecture (MDA) is developed, where MDA facilitates a standard for interaction of models between software tools. Figure 4 captures the core of the MDA framework.

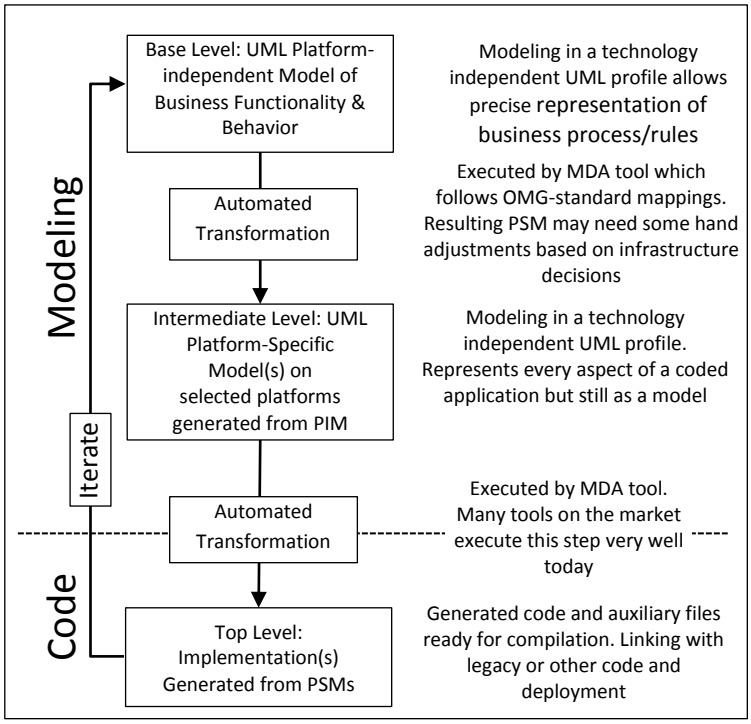


Figure 4 MDA framework[3]

[3] www.omg.org

In the base level a UML model is used to create a platform independent model (PIM), through automatic transformation an intermediate platform specific model (PSM) is created, by another automatic transformation a code is generated which represents the final product, for instance an application. A PIM is a model of something that is valid independent of any software tool; it is like a music score, where a tune can be played by more than one instrument. A PSM is a similar model, but now it is created for one specific software tool. It is like making sheet music for the piano and another set for the guitar, but for the same song. By making use of this MDA standard, it is possible to use existing tools for automatic transformations (Kleppe, Warmer, & Bast, 2003). This is convenient because there are already tools on the market that comply with this standard. If we translate the MDA standard to the specific goal of this research two problems arise. Firstly it must be possible to create a standard model of an information intensive process in UML. Secondly the resulting UML model must be transformed into a PSM, in our case a model in either Mendix or Be Informed. The third step, from the PSM to the implementation is already done by Mendix and Be Informed, so that part of the puzzle is already solved. The two problems are translated into three research questions, which are stated below. The first problem is tackled in two separate research questions.

R1. What elements are needed to describe an information intensive process model?
    R1.1 How do the identified elements relate to each other?
R2. How can the selected elements be used to create an instance of an information intensive process?
R3. How can the elements identified above be mapped into software tools, like Mendix and Be Informed, for the execution of information intensive processes?

Figure 5 places the research questions (R) in respect to the structure of this report. It indicates also in which steps an example (E) is used. In the E1 phase the example is used to show how the meta-model can be used to create a platform independent model. In E2 the created platform independent model is used to create applications with Mendix and Be Informed. Figure 5 indicates that there is a black box between the PIM and the application (R3). This is called a black box because users of the EAD method do not have to, and will not know what happens in that stage. In the long term vision users will just push a button which will turn their application into a software tool. This step will be crucial to the success of the EAD method.

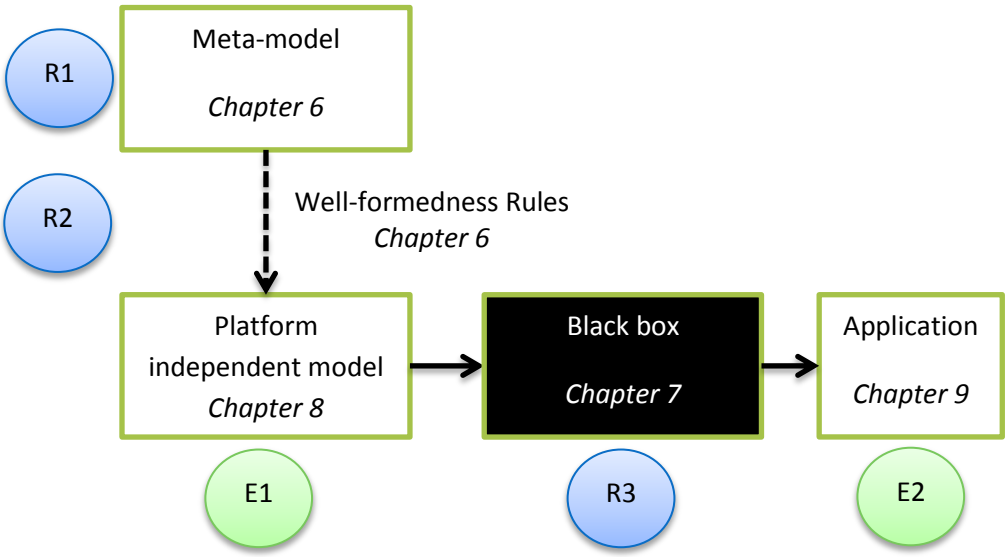


Figure 5 overview of research questions in relation to the EAD method

5 Methodology

Research in the information systems field is often balancing between the environment and knowledge base (Hevner, March, Park, & Ram, 2004). Two streams are identified, develop/build and justify/evaluate. This research is clearly in the develop/build stream, because a new method is developed. Therefore a methodology for design science is used. Hevner et al. (2004) suggest 7 guidelines for design science in information systems research:

1. Design as an artifact
2. Problem relevance
3. Design evaluation
4. Research contributions
5. Research rigor
6. Design as a search process
7. Communication of research

These seven guidelines are used as guidelines in this research, meaning that an attempt is made to follow the guidelines but practicalities might cause other decision making. In the discussion of chapter 10 a reflection is presented on how each guideline is respected. Now for each research question objective, methodology, and deliverable are discussed.

5.1 Question R1: Elements

R1: What elements are needed to describe an information intensive process model?

Objective

The goal is here to identify several elements that are needed to describe an information intensive process in such a way that it can be modeled into an actual application. One can think of roles, tasks, artifacts, business rules, and many other elements. The answer to this question will be the input for the second and third research question. Which elements are essential, will be researched. Additional to selecting the elements, the relation between the elements is also investigated (R1.1). The final objective of this first research question will thus be a group of selected elements and their relations, in other words: a meta-model.

Methodology

In order to obtain an answer for research question R1 several iterations will be conducted. First a starting point is created based upon literature and interviews with consultants from O&i. Also elements that are used by Mendix and Be Informed will be considered. At least five different scientific models for the modeling of information intensive processes that are described in literature will be evaluated; this will allow the evaluation of different approaches on modeling an information intensive process. An initial meta-model will be composed, showing the elements and their relationships. With this basis, new interviews will be held with the consultants, now also discussing the relations between the elements. After each interview, if necessary, the elements and or relations will be modified. Also the elements used by Mendix and Be Informed will remain input for the final meta-model.

The first round of interviews will be semi structured with a funnel approach (Voss, Tsikriktsis, & Frohlich, 2002). At least three consultants and one scientist will be interviewed in order to gain understanding of the possible elements.

During the second round of interviews the consultants will be asked to give their motivated preferences on the model of the latest iteration, again the format will be a semi structured interview.

In order to have insights from the academic point of view at least one staff member of the Eindhoven University of Technology will be interviewed in both rounds. Every step has the goal of gradually improving the meta-model; iterations will continue until no tangible improvements are made in a single step.

Deliverable

The deliverable for R1 is a meta-model which can be used to model information intensive processes, consisting of elements and their relations. A precise definition for each element and a motivated justification of the final selection of elements is an accompanying deliverable.

5.2 Question R2: Instance

R2: How can the selected elements be used to create an instance of an information intensive process?

Objective

The result of question R1 will be a meta-model that can describe an information intensive process. This second research question R2 will serve the purpose of identifying how this meta-model can be used to create an instance, i.e. a model of a specific information intensive process. This instance must be written in a well-defined language, a language with a well-defined form (syntax), and meaning (semantics), which is suitable for automated interpretation by a computer (Kleppe, Warmer, & Bast, 2003). One could say that the elements of the meta-model are words, which are meaningless without rules on how to make sentences, i.e. grammar. Creating these rules is the objective of this research question.

Methodology

The transformation of a meta-model towards a specific instance is described in the MDA methodology as 'well-formedness rules'. The rules will be stated and through iterations improved. When no significant improvements are achieved, the well-formedness rules are done. The quality will be evaluated on a number of measures. In general one can think of measures like coupling, cohesion, complexity, modularity, and size (Vanderfeesten et al., 2007).

Deliverable

The deliverable after answering this second research question R2 is a set of rules which will make the creation of an instance of the meta-model possible.

5.3 Question R3: Mapping

R3: How can an instance of the meta-model be mapped into software tools, like Mendix and Be Informed?

Objective

With answers to research question R1 and R2 only one step remains to abridge the gap between workflow modeling and application creation. In order to answer this research question we will look to two specific application creation tools: Mendix and Be Informed. They create applications that are Java based, the underlying information is described in XML. Of course each software package uses its own programming style, therefore for both packages the created models must be mapped into XML documents that will instantly create an application of the desired workflow. The research question R3 is thus how to do this.

It can take a very long time to achieve a full description where all possible scenarios from real life can be transferred into the two selected systems. As they say, the devil is in the details. Therefore the objective is to describe how the selected elements and their relations can be transferred into the two systems, exceptions are thus left out of scope. The quality of the description must be such that the next step can be the development of a tool that can make the transition of the instance to the software tools automatically. Eventually, when the method is in a later development stage, the transition can be automated fully.

Methodology

As mentioned before Mendix and Be Informed were selected because we have access to both. Dr. Marco Comuzzi from the Eindhoven University of Technology has knowledge on Mendix, Ir. Eyad Sabbah from O&i has much knowledge on Be Informed.

We will first identify for each of the two systems how the information must be transmitted to the systems so that applications can be created. Initially this will be done on the basis of how both systems work, in order to immediately check the results and clarify the transformation rules. A simple example is introduced. Finally the transformation rules will be verified by discussing them with the two aforementioned experts. Also, the transformation rules will be the basis upon which a more complex example will be built (part 4).

With this information we can make a description of how the elements described in the created meta-model must be transferred into selected systems.

Deliverable

The deliverable of research question R3 is a description in the form of a set of transformation rules, on how the meta-model can be transferred into Mendix and Be Informed.

Part 3: EAD method

So far the research has been introduced by giving the objective, and terms and definitions, followed by a discussion of the research questions and methodology. This part of the report gives an answer to these research questions and specifies the first elements of the EAD method. Validation will be presented in Part 4, in the form of an example.

6 Elements

This chapter answers the first research question R1, identifying which elements are needed to describe an information intensive process. First it is explained what steps are taken, and what the results are. Secondly the selected elements are presented in the form of a model that will visualize the associations between the elements. The model will be accompanied by a list where each element is defined and a list where all the associations are explained. Finally a discussion is held why elements were selected in the model and why others were not. After reading this chapter you know what elements and associations are that are used to describe an information intensive process, which is the input of the second and third research question.

6.1 Process of answering the first research question

In order to gain an answer to the first research question *what elements are needed to describe an information intensive process model,* three general steps were taken. This is done to maintain a balance between rigor and relevance, as well as practicality. The start was to create an overview of models that are described in scientific literature which allow modeling of information intensive processes. This overview can be found in table 1. The table indicates what elements are described by the different scientific models. The different models can be found in the horizontal rows, while in the columns a grouping of elements is made. The resulting matrix shows what element comes from what model and to which category the element belongs. These categories where formed after researching the different scientific models. They form the starting point of elements for the model of the EAD method. A more elaborate description of each model can be found in Appendix A.

The second step was to review Mendix and Be Informed, as displayed in table 2. The basic building blocks of both software tools were placed in the column of the elements specified in the first step. It is important to keep Mendix and Be Informed a part of the iterations conducted in the third step, because finally the model must be transformed to these two software tools. On the one hand, if they would be left out of scope until the moment of making this transformation the risk is that the model is almost impossible to transfer into Mendix or Be Informed, because the approaches are so different. On the other hand, the reason to select two software tools was to prevent that the created model only works with one software tool. The long term vision is that the EAD method can comply with other software tools, with different specialties.

The third step consisted of iterations, by interviewing experts and adapting the model. Two rounds of interviews can be distinguished. The goal in the first round of semi- structured interviews was to gain insight and interviewees were asked open-ended questions on what elements they see as relevant for the modeling of information intensive processes, and what elements they use in their daily practice. The second round of again semi-structured interviews was to validate the created model. Interviewees were shown the latest version of the model and a discussion was held on what they thought of it. After each interview the model was adapted according to the comments of the interviewee. The model was considered ready when suggestion by interviewees did not lead to significant improvements, this happened after nine iterations. Some key supporters of this research were interviewed multiple times per round. An overview of which individuals were interviewed in each round can be found in Appendix F.

Table 1 overview of artifact centric models

Scientific Model	Reference	Roles	Tasks	Data	Presentation	Status	Constraints
BALSA	(Bhattacharya, Hull, & Su, 2009)		Services	Artifact information model		business artifact lifecycle	Associations
PBWD	(Reijers, Limam, & van der Aalst, 2003)			data elements		Production rules + meta data	Constraints
FlexConnect	(Redding, Dumas, ter Hofstede, & Iordachescu, 2010)		JOB			COROB & ROB	Pre/post gateways & creation region
Xflow	(Marchetti, Tesconi, & Minutoli, 2005)	Role chart		Schema	Document interface description	Document workflow	
TriGS$_{flow}$	(Kappel, Rausch-Schott, & Retschitzegger, 2000)	Role model				Object-oriented database system	ECA rules
Declare	(Pesic, Schonenberg, & van der Aalst, 2007)		Work list		Designer	Framework	

Table 2 overview of elements in software tools

Type	Roles	Tasks	Data	Presentation	Status	Constraints
Mendix	Security	Micro flow	Domain model	Form builder		Permission
Be Informed	User groups	Eventtype	Dossier	Form	Status	Permissions

6.2 The selected elements and their associations

Let us take a closer look at table 1. What elements have the same function in the different models? Elements with equal/similar functions were grouped together, the group was titled and that lead to the first six element groups for the model of the EAD method. The comparison of scientific models in table 1 thus led to a meta-level; one abstraction level above the scientific models and their elements that were considered in table 1.
The six meta-elements as displayed in table 1 are:

- Roles
- Tasks
- Data
- Presentation
- Status
- Constraints

Two models, Xflow and $TriGS_{flow}$, incorporate for instance functionality to model roles. All the considered models incorporate functionality to model status. A more detailed analysis of each scientific model can be found in Appendix A. Not all models incorporate a presentation function, because these models are not directly used to create applications. Mendix and Be Informed are software tools to achieve just this; therefore they both have built in functionality to present the information to the user.

Based on these six meta-elements a first meta-model is now presented in figure 6. The majority of the initial elements are included. A number of elements are added, and some are changed depending on what was mentioned during the interviews. For instance the element lifecycle was transformed to status, and data is split into three steps (case, entity, data item). This is done to be able to distinguish between the three levels of data. Another reason was that Be Informed, and Mendix to some extent, incorporate this style of managing data. We will now discuss the meta-model as depicted in figure 6, definitions for each element are stated in chapter 6.2.1. and the associations are discussed in chapter 6.2.2..

In the center of the meta-model the task element is presented. Any information intensive process consists out of tasks; almost all scientific models have some element representing a task, just as Mendix and Be Informed. The task element is linked to almost all other elements, because a task brings elements together. Pre-post-conditions are added so it becomes possible to set conditions to the execution of tasks for a specific case (pre-conditions). For instance some tasks can only be performed when a case is in status X. Also conditions can be set on the ending of a task (post-conditions). A possible condition prescribes that a task can only start when some outside event has occurred. These events must also be given a place in the system, therefore the event element is added. Events can be online applications by customers, or other organizations delivering data (e.g. test results) or physical products. The event element allows

for pre-modeling standard events in order to have an associated task for acting on occurrence of the event.

The case element and the associated status element are also at the core of information intensive process. We have chosen to use the word case rather than artifact, and status rather than lifecycle. This was mentioned during one of the interviews, and confirmed in other interviews. In both cases the alternative is primarily used in literature and the chosen terms are used by the execution engines. Since the goal is to create a meta-model that can be used in practice, the terms are selected for the closer relation to practice.

Continuing clockwise, the element role and the associated element permission are presented. The role is included since in many processes many different kinds of expertise are required to complete the entire process. Expertise is something that can be classified as role and in many companies functional roles are listed. Giving permissions to roles rather than to individuals is done so that the model is independent of transformation of the workforce. In the permission element positive or negative permissions can be set for each role. One can think of role X is not allowed to perform task Y, or role Z can read but not write in case W.

In order to make it possible for the employees to perform the work in a task, forms need to be created that allow the reading and/or writing of tasks and their associated cases. The association between task and form is not a one-to-one association since some tasks are automatic tasks and do not need a form. In fact all tasks that are not associated to a form must be interpreted as automatic tasks. The MDA framework prescribes that the PIM must be in UML format, therefore the meta-model is in the same format.

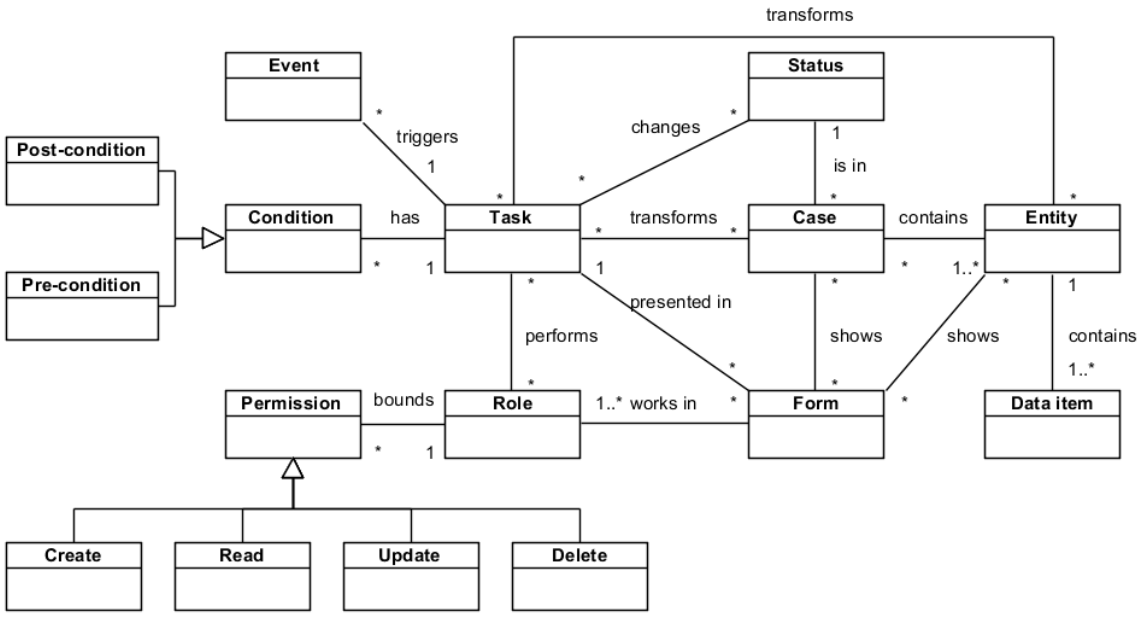


Figure 6 EAD meta-model

6.2.1 Elements

This section contains a definition of each element out of the meta-model, making clear what precise is meant. The elements are in alphabetic order.

Case - The case contains all information of the artifact, and also the information about the artifact that are needed for the handling of the case. It is a combination of entities.

Data item – Characteristic that describes and/or identifies the entity. A data item can be of several types; integer (e.g. age), string (e.g. name), float (e.g. price), date (e.g. birthdate), or enumeration (e.g. religion).

Entity - An entity contains information about one specific item, e.g. personal data.

Event - Something that happens outside of the organizational context that can trigger the start of a task.

Form - A visual presentation of a task. The form is used by the appropriate role to perform a task. Since the task can transform a case, a form is used to do the actual transformation. A form can also be used to just view an entity.

Role - A role describes responsibilities that are assigned to one or more employees with the same organizational role. A role performs tasks and is bounded by permissions.

Permission - Permissions are statements that delineate, limit or bound what a role can do. Permissions are presented as a Boolean function; it can either be true or false. For instance, role X is allowed to create new cases of type Y. And, role X is not allowed to delete cases of type Z.

Create, Read, Update and Delete are added as specializations sins as well as in Mendix as in Be Informed they are standard integrated permissions which are straightforward in their application. Other permissions are possible, but need special attention.

Pre-post-Condition - Statement that must be respected by the associated task. The pre-condition may prescribe under what circumstances a task may be performed. Post-conditions may prescribe what must be done before the task can be completed, and what the influence of completing the task is on other elements.

Status - The status shows the position the case is in i.e. a predefined set of possible statuses is given for each type of case. A task can change the status of the case.

Task - A task is an activity that impacts a case. Several cases can be used in one task, none, several, or all statuses of the cases may be changed by the task. A task is performed by a role or by a computer (automatic task). The task is either started by the reception of a case or through

explicit triggering mechanisms. When the task is finished no residual information is held in the task; all cases are either sent out or are discarded (Nigam and Caswell, 2003).

A task is something that must be done, either by a human or automatically by a computer. The question is how one can specify what exactly has to be done in a task, especially if it is an automatic task how the computer should do it. There are two ways to deal with this problem. The first is to describe the task in such a way that a computer can fully understand what has to be done. For example the automatic retrieving of information from an external source, let's say the full address when only the postal code is given. Describing this task on a meta-level must then be already put in a language computers can understand, like XMI. The alternative way to deal with this problem is to just give a textual description of the task on the meta-level. When this model is then transferred to a platform specific model only a shell for the task is created, the actual task needs to be developed within the specific platform. The major downside of this approach is that it is not possible to make changes in the meta-level model. For example you want to design a process by using the meta-model. After the modeling phase you transform the PIM to a PSM, which is not functioning yet. Therefore you fill in all the tasks in such a way that a running application is created. After a validation phase of running the PSM you decide that some elements need changing. Now you can do two things, the first is to make the changes directly in the PSM, with the downside that you do not have the ease of the PIM. Or you decide to make the changes in the convenient PIM, with the downside that you need to transfer the PIM to a new PSM and fill in all the tasks again. This immediately makes clear that the ideal situation would be to be able to go back and forth between PIM to PSM easily. If this is possible you can generate a fully working PSM from a PIM, and generate a new PIM from a stable working PSM.

Where does this discussion lead to one might wonder? If we zoom out and look at the goal of this research project we can see that the preferred situation where it is possible to go back and forth between a flexible PIM and a stable PSM belongs to the long term vision. In this first phase the second alternative, where the PIM creates a basis PSM that needs to be completed, is used. Hence, a full transformation between a PIM and PSM is envisioned for the future, and not a realistic first step.

6.2.2 Associations

Since the UML notation is used for the meta-model, the nature of the associations is visible in the model of figure 6. Each association will be further discussed in this chapter, sorted alphabetically.

Case – contains – **Entity**
Every case is built out of entities. A case contains at least one entity, but can contain more. Each entity can be linked to zero or more cases.

Case – is in – **Status**
Every case is in a status, which can be changed by a task. Each case is at least linked to one status, but more statuses are possible.

Entity – contains – **Data item**
An entity is built out of data items.

Event – triggers – **Task**
An event can trigger the execution of a task.

Form– shows –**Case**
Forms are used to represent the information to users. A form thus shows the content of a case. The status and entities that are associated to the case are shown in the form.

Form – shows – **Entity**
A form is used to show the information that is contained in an entity.

Permission – bounds – **Role**
Permissions are bounding what a role is and is not allowed to do. This can be the viewing of data, or the performing of a task.

Role – works in – **Form**
A user who has a specific role uses a form to either perform a task or to view data. These data can be a case or an entity.

Role – performs – **Task**
A role performs a task it is connected to.

Task– transforms – **Case**
A task transforms information in a case. The information that is changed can be stored in an entity that is linked to the case.

Task – has – **Condition**
The task can be bound by conditions. These can be pre-conditions and/or post-conditions.

Task – transforms – **Entity**
A task may transform an entity. For instance, changing the address in the entity person.

Task – presented in – **Form**
If a role performs a task, a form is used to represent the task. For instance, the task 'change the address' is done in a change form.

6.3 Discussion of the elements in the meta-model

The basic paradox with creating a meta-model is that you want to have a model that can be used to represent reality, and all of its complexity. On the other hand, the model is more likely to be used when it is simple, with a minimum of elements. This is in contrast to a model that can deal with the complexity of reality, i.e. a model with many elements. The challenge is to create a model that finds a reasonable balance between complexity and simplicity. We have attempted

to balance the two, however further research is needed to judge if the presented meta-model is well balanced.

For the permissions you can see that there is a specialization for four specific permissions; Create, Read, Update, and Delete (CRUD). This specialization is inserted because both Mendix and Be Informed incorporate these permissions in their software. One might think of other permissions, but permissions can always be translated to CRUD permissions. They only need to be applied in the right location. The reason that this is possible is that we are talking here about information intensive processes and therefore information that needs to be handled. And there is a limit to what you can do to digital information. This is also the reason why these permissions are standard implemented in both Mendix and Be Informed.

An alternative permission for the sharing of information can be transformed into a CRUD permission. For example, role A is allowed to create, read, update and delete personal information. However role A is not allowed to share the information with other employees or people outside the organization. What you could do then is set the permission to the share functionality (in Mendix you would make a button which refers to a microflow) to 'not allowed' to create, read, update or delete that specific microflow.

A number of elements were considered, but not included in the meta-model, based upon the expert opinion of the interviewees. These excluded elements are discussed next. Thereafter the element task is discussed separately because of its special characteristics. With task it is not possible to grasp what needs to be done by just naming the element. For instance with role you can give the role a name and that is it, for task it is not this straightforward.

6.3.1 Excluded elements

One of the interviewees suggested attaching a user element to the role element. Then in this user element the information of the employees can be filled in. The association to the role element will clearly indicate which user has which role. However the user element is not selected, because in Mendix and Be Informed the users are added as soon as the system is running, in the implementation stage. Therefore there is no need to include users into the meta-model. Another reason is that the meta-model must be focused on cases, and thus operate independent of the involved employees.

Another element that was discussed often was the process element. Including a process element would allow the specification of the process a task or case belongs to. However, groups of tasks are often so specific that the group of tasks belonging to one process will have its own application. Consequently, each application is named after a process, which is of course useful and it does not require including a process element in the meta-model. In other words if you look into one process it is not relevant to have the option to distinguish to which process each task belongs, as all tasks belong to the same process.

6.4 Creating an instance

In the previous chapter the meta-model was presented. This chapter explains how the meta-model can be used to actually create an instance of an information intensive process (PIM). An example of a fully worked out instance is discussed in chapter 8, a full description of this example can be found in Appendix F. In chapter 7 a description will be presented of how the created instance can be transformed from PIM into Mendix and Be Informed. But first the focus is on finding the answer to the second research question R2: *How can the selected elements be used to create an instance of an information intensive process?* Therefore we turn to so called well-formedness rules as suggested by the Model Driven Architecture of OMG (Kleppe, Warmer, & Bast, 2003).

Since this thesis only concerns the specification phase, the rules are presented in words. To achieve the long term vision software must be created that allows the automated creation of an instance on the basis of the presented meta-model combined with the well-formedness rules presented in this chapter.

6.4.1 Well-formedness rules

The well-formedness rules (WFR's) as stated below are of a general nature as they are merely meant to give general guidelines for the use of the meta-model. In other words everything is possible, but these rules must be satisfied for the result to be valid.

1. The instance must be created in the UML format as prescribed by the OMG.
2. The associations that are set in the meta-model must be respected. For instance an entity can contain one to many data items and a data item can be part of only one entity.
3. For each element a name is created stating what it represents. For example status: 'processed', and status: 'handled'.
4. If a task is performed, the information about the execution is stored automatically. One can think of information about the employee who executed the task at what time.
5. Conditions can only refer to elements that are already described in another section of the model.
6. Conditions can refer to elements they are not directly linked to.
7. A data item can be of any type like 'string' or 'integer'.
8. Within each element it is possible to write further instructions in text.
9. It is not obligatory to make use of all elements when modeling a process.

The presented example in chapter 8 will clarify how these rules are applied to come to an instance.

To create an instance based on the meta-model you can start with any element. But obviously if you want to refer to another element, this element needs to be created first. For example the role Clerk is not allowed to view the case Problem when it is in status Finished. You will see that

in practice permissions and conditions can only be created when all other elements have been created since they highly depend on the other elements.

6.4.2 Tools

One of the functions of a tool is that it allows the consultant to create business models that can be interpreted by computers. The created model must thus respect the abovementioned WFR's. Several software packages are available today that can help in the creation of such a tool. Eugenia by Eclipse is an example. With this software the rules are implemented and when an instance is created the rules are automatically checked. If a rule is violated, the software will notify the builder that there is a violation and identify the rule that is violated.

The tool selected to do this must be easy to use because eventually consultants will use this tool to create the instances of information intensive processes. This model in turn will be automatically transformed into a software tool like Mendix and Be Informed with the click of a button, as stated in the long term vision. What actually happens when such a button is clicked is explained in the next chapter.

7 Black box

This chapter explains how the elements presented in the meta-model relate to Mendix and Be Informed. Answering research question R3, *How can an instance of the meta-model be mapped into software tools, like Mendix and Be Informed?* This mapping is important since the long term vision is to have a method that can turn PIM's (instances) into PSM's (applications in software tools) automatically. For the black box to fully function two steps must be taken, the first is managing the transformation from a PIM to a software tool (inside the scope of this project), the second is automating this (outside the scope of this project). The term black box is used because user of the EAD method will not know how the transformation occurs. Chapter 7.1 will give a peak into the black box, explaining what happens in general. What happens specifically in the case of Mendix and Be Informed is described in chapter 7.4 and 7.5. The information in this chapter is given in plain text as lies within the goals of this research. Eventually the presented transformation rules will be programmed, allowing automatic transformation.

After reading this chapter you know how in general the transformation from the PIM to an application is made. For Mendix and Be Informed specific transformation rules are presented, showing how for these two software tools the transformation can be achieved.

7.1 Black box explained

So far we have seen that with the use of a meta-model, and well-formedness rules an instance can be created (PIM). The next step is going via the black box towards the preferred application, as is indicated in figure 7; each application needs its own black box. Since the focus for this research is on Mendix and Be Informed, those two are made explicit, but various other software tools can be used as well. However for other software tools the black box needs to be developed completely from scratch before it can be used.

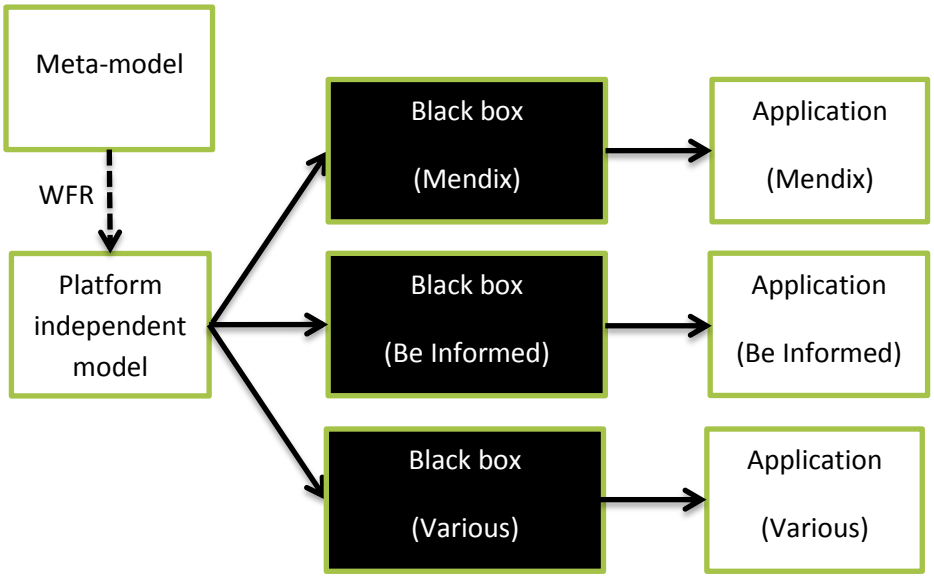


Figure 7 black box of the EAD method

So what does this black box do? On a meta-level, it transforms the PIM to a model within the environment of the selected software tool. Often application providers have a modeling feature, as is the case with Mendix and Be Informed. Since the PIM is written in a well-defined language (UML), the transformation by the black box can automatically interpret the information and transform it to the preferred application. During the development phase of the EAD method some programming is required to make this transformation go automatically. But for the specification phase the transformation is presented as specific rules in plain text.

The selected approach is to work with small examples. First the example is transformed to a PIM by use of the meta-model and well-formedness rules. Secondly the PIM is transformed by hand to Mendix and Be Informed. The actions taken in these small examples are then transformed into general rules. So each application will have its own set of transformation rules. This approach is selected because it can test the possibility to transform a PIM into an application. Stating a full list of rules that will make it possible to model the complexity of any possible scenario is no option. This would take a very long time. Instead starting small and slowly expanding the possible applications is a much better alternative. Think of photo editing: in the beginning this was only possible for specialists using special software. Today it is possible for every one with simple software to handle some aspects, like red eye reduction or light changes. However, advanced editing can still only be done by specialists using special software, but this software does include the easy red eye reduction feature as it is in the simple software. This simplification of small problems is a first step where everyone can benefit from.

7.2 Transformation rules

In order to bring logic into the transformation rules that will be presented two levels of separation are created, basic structure and customization. In the first phase transformation rules are presented which must be followed in order to create a working application. The relation between the elements in the meta-model and the software tools is rather straightforward. However, it is made explicit what needs to be done, and in what order. Eventually software should be able to make these transitions automatically. With this basic structure the application can be customized by adding elements. This customization will require manual intervention in order to create a working application.

In the first phase a logical approach is used, short statements make clear what must be done in order to create an application. If in an instance of the meta-model more elements of the same type are used, the specific transformation rule applying to that element must be repeated for each element. For instance there are three cases, then for each case the transformation rules concerning the case must be executed.

In the second phase, customization, a description gives a set of the possibilities of the software tools. For instance how in Mendix a change of status can be realized. After both phases a comparison between Mendix and Be Informed is made on what is specifically possible in both software tools. What are strengths and weaknesses?

A simple instance of the meta-model is presented here for clarification purposes. There after the rules for Mendix are presented, followed by the transformation rules for Be Informed are presented.

7.3 Example

Information intensive processes often concern personal information. Therefore this simple example is based on a case where personal information needs to be managed in a product ordering system. The application allows the creation of a new person, the deletion of a person, and changing the information of a person (for instance address). Name, street, postal code, house number, and bank account will be stored in the entity Person. The entity Product contains items and corresponding prices. Together Person and Item form the case Order. The role Registrator is allowed to create an order and the Handler role is allowed to change and delete the order.

Furthermore two statuses are introduced: Active and Inactive. Status change can only be made by the Handler (permission). Every new person is always in an active status and this can be changed to an inactive status.

The event order is added; when a person makes an order it must be filed in the system. The condition that applies to this order is that the total price of the order must exceed 20 euros (condition).

The model of this example can be found in Appendix D.

7.4 Mendix

First we look to the transformation rules that will form the black box of Mendix. Within Mendix there are a number of possible ways to create, add or connect elements:

1. Create: create a new Mendix element on the top level (an element that is separated completely from all other elements on creation)
2. Add: add elements to a another element; the creation of a connection to an existing element is already made (i.e. data items to case, or permissions to role)
3. Link: associate entities
4. Compose: compose a microflow

7.4.1 Phase one: basics

In Appendix E a number of screenshots of Mendix are placed. This will allow you to see how the different elements look in the Mendix modeling environment. As becomes clear from table 3, most elements out of the meta-model can be transformed into Mendix without much extra effort. Mainly the elements need to created and linked to the appropriate elements within Mendix.

Table 3 basic mapping in Mendix

Number	Meta-model element	Mendix transformation	Comment	Example
1.	Entity	Entity	Create an entity in the domain model for each entity	The entity product and person are created
2.	Data items	Attribute	Add to the case the associated data items as attributes	To person name (string), street (string), postal code (string), house number (integer), and bank number (integer) are added. To product item (string) and price (float) is added.
2.1.			In basic everything is a string, but the following can be set separately	
2.1.1.		DateTime		
2.1.2.		Integer		
2.1.3.		Float		
2.1.4.		Enumeration		
3.	Case	Entity	Create an entity in the domain model for each case	The case order is added
3.1.			Link each entity to its corresponding case	Person and product are linked to order
4.	Status	Entity-enumeration	Create an entity in the domain model for status	The entity for status is added
4.1.		Attribute	Add the different statuses to the entity by means of an enumeration attribute	Active and inactive are added in an enumeration attribute
4.2.			Link the status entity to the appropriate case	The entity status is linked to case order
5.	Form	Form	Generate a form, the name corresponds to the name of the form in the MM	A form is generated
5.1.		Data grid	Insert a data grid and link it to the associated case	A data grid is inserted and linked to the case order
5.2.		Automatic form	Generate a NEW/EDIT form automatically from the basic form	A new/edit form is generated
6.	Role	Security	Add user roles within the security page of the project*	The roles Registrator and Handler are added
7.	Permission	Security	Add permissions in the security section of the module	
7.1.		Module roles	In the first tab “Module	

			roles" the different roles are visible	
7.2.		Form access	In the second tab "Form access" per role it can be identified what form can be accessed	
7.3.		Microflow access	In the third tab "Microflow access" per role it can be identified what microflow can be accessed	
7.4.	Create, read, write, and delete	Entity access	In the fourth tab "Entity access" per role it can be identified which entity can be accessed, the possible settings are Create, read, write, and or delete. Within the entity you can also specify which attribute can be accessed options are none, read, and/or write.	The permissions for Registrator are set to create for all entities, the permissions for handler are set to read, write, and delete

* Mendix allows the use of different modules within one application. Each module can have its own security settings. For complexity reasons this option is left out of scope.

7.4.2 Phase two: customization

When the basic transformation rules are applied the basic of an application is created. Of course it is possible to modify the application in many ways in order to make it fit the requirements. The majority of customizations in Mendix can be realized by creating microflows. This section explains what the possibilities within Mendix are, table 4 explains how a basic microflow is constructed and table 5 explains what the possible action activities in a Mendix microflow are. An explanation how a microflow is built can be found in Appendix E.

The microflow can also be used to incorporate pre-conditions, post-conditions, and events. In any case the association to the appropriate task is straightforward, as the condition or event is implemented in the microflow as an action activity in the microflow that represents the task.

The tasks 'create, read, update, and delete' are automatically transformed into Mendix through the forms. They do not need specially created microflows. In each form with a data grid, four standard buttons are added: Search, New, Edit, and Delete. Default anybody is allowed to use all the buttons, however in the security section it is possible to set permissions such that not everything is allowed by each role.

For an example a microflow is made which will allow the status to be changed by the push of a button. Since you start in an overview form of a person, the associated status needs to be retrieved. After this first action activity a second action activity is added which will change the status to inactive. Hence, all persons begin active and can be turned into inactive, which is permanent in this simple example.

The functionality and settings of each possible action activity will be discussed next[4].

Table 4 creating a basic task in Mendix

Number	Meta-model element	Mendix transformation	Comment	Example
8.	Task	Microflow	A task is created by adding a button to a form which refers to a microflow	Add a button to the overview form that refers to a microflow
8.1.			The microflow is composed out of the following elements	
8.1.1.		Parameter	Add a parameter	Add a parameter
8.1.1.1.			Link the parameter to the required entity	Link the parameter to the entity (case) order
8.1.2.		Start event	Add a start event	A start event is added
8.1.3.		Action activity	Add an action activity	An action activity is added
8.1.3.1.			Link the activity to the appropriate entity	
8.1.4.		End event	Add an end event	An end event is added

Table 5 custom mapping in Mendix

Number	Microflow group	Microflow element	Functionality	Setting
9.	Object			
9.1.		Cast object	Cast Object can be used in combination with an inheritance split to use the specialized members of the object	Set the variable name to cast
9.2.		Change object	Change Object can be used to change the members of an object.	Set the variable that needs to be changed, and set what the new value of the object needs to be

[4] All information is retrieved from the Mendix website (world.mendix.com)

			This can be done with or without committing and with or without events	
9.3.		Create object	Create Object can be used to create an object	Select the entity for which a new value must be created (choose person if you want to add a person)
9.4.		Delete object	Delete Object can be used to delete an object	Select the variable to delete
9.5.		Retrieve	Retrieve can be used to get one (or more) associated objects of another object. Furthermore the activity can also get one (or more) objects directly from the database	Select whether to retrieve by association or from database, if applicable select the association. Enter the name of the output variable for further usage in the microflow
9.6.		Rollback object	Rollback Object can be used to undo changes (that have not been committed) that were made to the object in the part of the microflow preceding the activity. Furthermore it deletes objects that have been created but have never been committed	Select the variable to rollback
10.	List			
10.1.		Aggregate list	Aggregate List can be used to calculate aggregated values such as the maximum, minimum, sum, average and total amount of objects over a list of objects	After selecting the variable for calculation, the type of calculation must be selected (sum, average, count, minimum, or maximum). Give a name to the output, this can be used by other action activities in the same microflow (show the value in a pop up box, or for instance set the average income in the field average income of a person)
10.2.		Change list	Change List can be used to change the content of a list variable	Select the list variable, generate an expression that changes the list according to the requirements
10.3.		Create list	Create List can be used to create a (empty) list variable	Select an entity to list, give the list a new which can be used in this microflow
10.4.		List operation	List operation can be used to combine or compare two list with objects of the same entity	Select the first and second list for the operation, specify the required operation (union, intersect, subtract, contains, or equals). Give the output a name which can

				be used by this microflow
11.	Action call			
11.1.		Java action call	Java action call can be used to call a Java action. Arguments can be passed to the action and the result can be stored in a variable	Select a Java action, specify what the return type is and give it a name
11.2.		Microflow call	Microflow call can be used to call another microflow. Arguments can be passed to the microflow and the result can be stored in a variable	Select a microflow, specify what the return type is and give it a name
12.	Variable			
12.1.		Change variable	Change Variable can be used to change a Boolean, DateTime, Enumeration, Float/Currency, Integer/Long or String variable	Select the variable to be changed, generate the action. The variable can be changed into a constant, token or associated variable
12.2.		Create variable	Create Variable can be used to create a Boolean, DateTime, Enumeration, Float/Currency, Integer/Long or String variable	Set the data type and give the new variable a name. An initial value can be generated, again into a constant, token or associated variable
13.	Client			
13.1.		Close form	Close Form closes the form that is opened last by the user that calls the microflow where this activity is used in	No setting
13.2.		Download file	Download File can be used to enable the browser to download a specific file. The user, that calls the microflow where this activity is used in, gets a download popup or the file is shown directly in the browser	Specify the name of the input file
13.3.		Show form	Show Form can be used to show a form to the user that calls the microflow where this activity is used	Select the form that needs to be shown, specify how it is shown (in content, pop-up, or blocking pop-up) and select the form object

			in	
13.4.		Show message	Show Message can be used to show a blocking or non-blocking message to the user that calls the microflow where this activity is used in	Set the type of message (information, warning, or error), write down the text, and add possible parameters
13.5.		Validation feedback	Validation feedback can be used to display a red text below a widget that displays an attribute or association	Select the variable and its member which need validation. Enter a text for the validation and add parameters
14.	Integration			
14.1.		Call web service	Call Web Service can be used to call one of the imported web services. The content of the request can be edited. Furthermore the response of the web service can be mapped to entities, stored in a variable or be ignored	First select which operation of a web service is called. - Set the HTTP properties (location, authentication, user name, password) - Set the request header/body properties (mapping (simple, custom, or advanced)) - Set the output properties (response data type, and response usage)
14.2.		Export XML	With the export-XML action you can export the data stored in domain model entities into an XML document	Select the entities to export, indicate the parameter and give the output XML document a name
14.3.		Generate document	Generate Document can be used to create a document of a certain type based on a template	First select a file. • Set the language (current user, project default, or variable) • Select the document type (.html, .pdf, .docx, .doc, .rtf, or .odt) • Select a template • Set the margins, or leave them standard
14.4.		Import XML	Import XML can be used to save the data stored in a XML document in the structure defined in the domain model of the database	Give the name of the input document, select to which entities in the domain model the XML model refers
15.	Logging			
15.1.		Log message	With the log-message action you can create messages that appear in the log of your Mendix application	Select the level (info, trace, debug, warning, error, or critical), specify a log node name, the textual template and add parameters

7.5 Be Informed

Within Be Informed there are four possible ways of create, add or connect elements:

1. Create: create a new Be Informed element, creating a new element on the top level (an element that is separated completely from all other elements on creation)
2. Add: add an element to a top level element
3. Link: link an element to an existing element
4. Apply: apply permissions

Because the available Be Informed version is in Dutch, the Dutch terms are displayed in table 6. In order to prevent confusion, the information is presented in Dutch. A translation of the words can be found in Appendix C.

7.5.1 Phase one: basics

Again in Appendix E a number of print screens are added to show how the mapping will look in Be Informed.

Table 6 basic mapping in Be Informed

Number	Meta-model element	Be Informed transformation	Comment	Example
16.		Dossierservice	Create a dossierservice named dossierservice	
17.	Entity	Attribuut-groep	Create an attribuutgroep for each entity	For Person and Item an attribuutverzameling is created.
17.1.	Data-item	Element	Add elements (data items associated to the same entity	To person name (string attribuut), street (string attribuut), postal code (Postcode attribuut), house number (Nummer attribuut), and bank number (Bank nummer attribuut) are added. To product item (string attribuut) and price (Nummer attribuut) is added.
17.1.1.		Datum attribuut		
17.1.2.		Postcode attribuut		
17.1.3.		Bank nummer attribuut		
17.1.4.		String attribuut		
17.1.5.		Nummer attribuut		
17.1.6.		Keuze attribuut		
18.	Case	Dossiertype	Create a dossiertype named to the case	A dossiertype Order is created
18.1.			Link to dossierservice	
18.2.		Recordtype	Add recordtype	

18.2.1.		Attribuutgroep referentie	Add attribuutgroep referentie	
18.2.2.			Link to associated atribuutverzameling	
18.3.	Status	Status	Add status	For every possible status of the case add a status to the Dossiertype
19.		Gebeurtenistype	Create a gebeurtenistype	
19.1.			Link to dossierservice	
19.2.			Add atribuutverzameling referentie	
19.2.1.			Link to atribuutgroep	
19.3.	Pre-condition	Beginhandeling	Add beginhandeling	
19.4.	Post-condition	Eindhandeling	Add eindhandeling	
20.	Form	Formulier	Create a form	A form is created for managing the order
20.1.			Link to gebeurtenistype	
20.2.			Add Atribuutverzameling referentie	
20.2.1.			Link to atribuutgroep	
20.3.			Add question	
20.3.1.			Link to atribuutverzameling	
21.		Basisstijl	Create basisstijl	
22.		Gebruikergroep	Create gebruikergroep	
22.1.		Gebruikersprovider	Add gebruikersprovider	
22.2.	Role	Role	Add role	A Registrator and a Handler are added.
22.2.1.	Permission	Permission	Apply permissions	
23.		Dossierscherm	Create dossierscherm	
23.1.			Link to dossiertype	
23.2.			Add panel	
23.2.1.			Link to dossiertype	
24.			Create dossiertab	
24.1.			Link dossierservice	
24.2.			Add dossierscherm (referentie)	
24.2.1.			Link to dossierscherm	
24.2.2.			Add attribuutverzameling (referentie)	
24.2.3.			Link to attribuutverzameling	

25.			Create webapplicatie	
25.1.			Link to basisstijl	
25.2.			Link to gebruikersgroep	
25.3.			Link to gebruikersprovider	
25.4.			Add dossiertab	
25.4.1.			Link dossiertab	

7.5.2 Phase two: customization

As becomes clear from the previous overview of what needs to be done in order to create a running application in Be Informed is that it requires some effort. It is not possible to leave out of scope elements which you do not want to use. In practise this is not a problem since probably you always will use all elements. In each of the steps mentioned in table 6 you can customize the application. Therefore no specific explanation is presented here how customization can take place.

A basic application in Be Informed can be created by following the steps mentioned in table 6. Customizing the application can be achieved by changing configurations in each step.

7.6 Mendix versus Be Informed

Both software tools aim at making applications for information intensive processes, but they do this with a different angle. Mendix allows you to create an application in only a few steps, however limiting the options for customizing the application. Customizing an application in Be Informed is a bigger effort and needs a higher skill with the modeler, it will also be harder to make the transformation automatically. But it allows for more sophisticated results with Be Informed.

Part 4: Validation

In earlier stages validation was done by interviewing field and science experts. Especially during the composition of the meta-model this approach was used, in total 7 person were interviewed some of them several times (Appendix B). During these interviews the experts gave their opinion about the model and suggested changes. These suggestions resulted in the meta-model as displayed in chapter 6. The two other elements of the EAD method were also initially validated by interviewing experts. After this initial validation they were, together with the meta-model, put to the test. The role of the end user (consultant) was used to perform this part of the validation. A process description is interpreted by the consultant and transformed into a PIM. What choices the consultant made and the resulting PIM are displayed in chapter 8. For the consultant the work should end by clicking a button which will created the application in the selected software tool (i.e. Mendix), what has been referred to as the black box. Since for the black box only the transformation rules are created the application of the transformation rules is simulated by hand. What will prove that by applying the rules an application can be created. Of course this is a first validation, further validation rounds will most probably lead to adjustments of the different elements of the EAD method, this is for further research.

First an instance will be created on the basis of a process description, which can be found in Appendix F. This instance (PIM) is presented and discussed in chapter 8. For the creation of this instance the meta-model and well-formedness rules of chapter 6 are used. With this created instance running applications are created, which are presented in chapter 9.1 (Mendix) and 9.2 (Be Informed). Figure 8 captures an overview of the different steps, showing that all steps of the EAD method are followed during this validation.

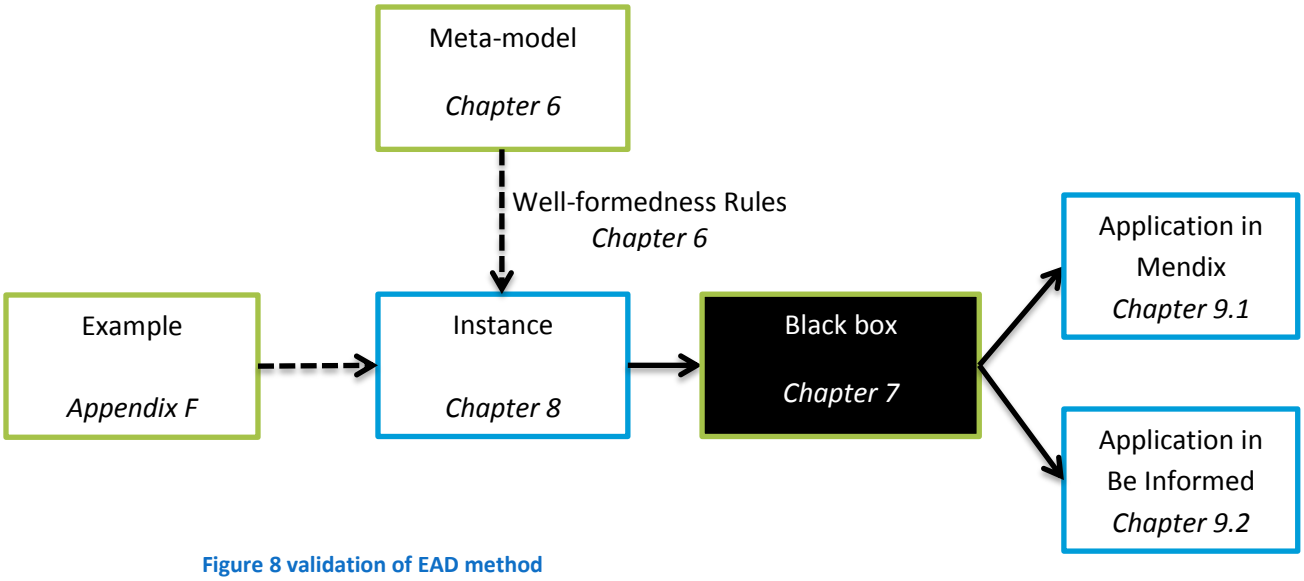


Figure 8 validation of EAD method

8 Instance of long example

For making an instance of the process description of Appendix F, first it must be decided (in practice by a consultant) what information corresponds to what element out of the meta-model. Secondly the relations must be selected. With this information a UML model can be created which captures all information. First we will clarify what information is of what type.

The fictive request of Rijkswaterstaat for a complaint application is used in the description of the example. The case in the description is the complaint (road related), the information that belongs to the claim are the following entities: location, discoverer, and impairment. Location consists out of the data items: province (enumeration), road number (string), street name (string), and postal code (string). Discoverer consists out of the data items: name (string), email (string), phone number (integer), street (string), house number (integer) and postal code (string). Impairment consists out of the data items: topic (string), date (date), explanation (string), urgency (enumeration), and type of complaint (enumeration). The case complaint can be in four statuses: received, handled, repaired, and archived.

Four roles are distinguished in the process: Clerk, Road worker, Manager (major rework department), and controller. The clerk is responsible for the tasks accept complaint, and research complaint. The road worker is responsible for the task sign of repair. In the process description receive complaint and make repair are also mentioned, but it is stated later that the fixing of the repair is left out of scope for this system, therefore the only tasks within the system for the road worker is to sign off on repairs. The manager major rework is responsible for the tasks decide on action. The controller is responsible for the tasks inform discoverer and archive settled complaint. Again the checking of the repair is a task that falls outside the system. However the controller must do this before he can inform the customer. Therefore the element pre-condition is used for the informing of the customer, this element states that the check must be done before the customer can be informed.

A number of forms are needed to present all the information and to perform the tasks. First of all a form where all the information represented is needed. Another form is needed form updating the information. A form is needed to give feedback to each discoverer.

The resulting UML instance is presented in figure 9. For this PIM all well-formedness rules are not violated, this is confirmed by hand.

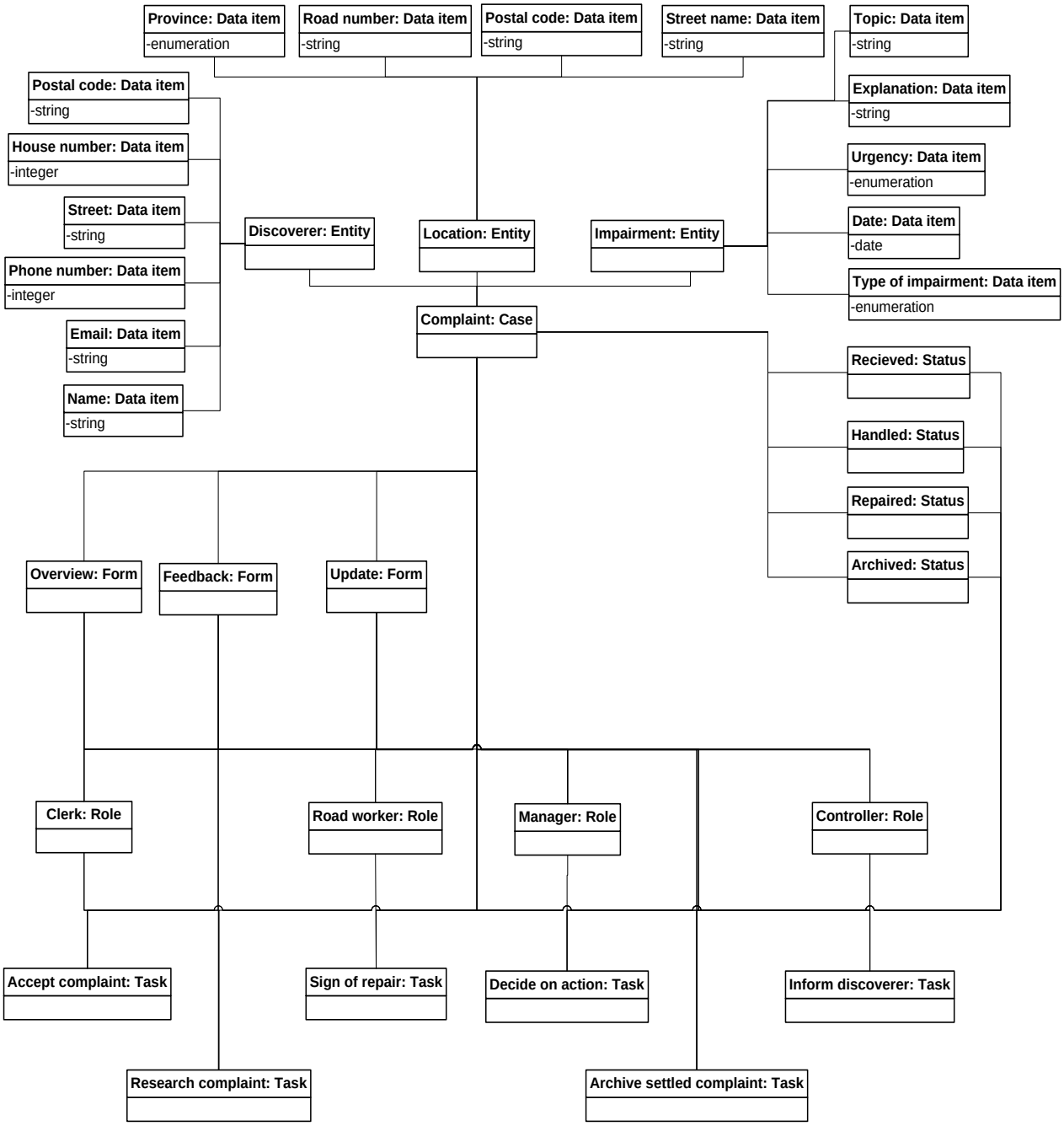


Figure 9 instance of long example

9 Applications

This chapter is used to make clear how the PIM of the Rijkswaterstaat example is transformed into Mendix and Be Informed. For this the transformation rules of chapter 7 are used as a basis, verifying the validity of these transformation rules. This chapter gives a peak into Mendix and Be Informed, showing how the software tools work. First we present what is done in Mendix in chapter 9.1, secondly the explanation for Be Informed is given in chapter 9.2.

9.1 Mendix

According to transformation rule 1 to transformation rule 4 entities in Mendix must be created for case, entity, and status. These must be connected as in the PIM. To the entities the associated data items are added as attributes. This results in the Mendix domain model of figure 10.

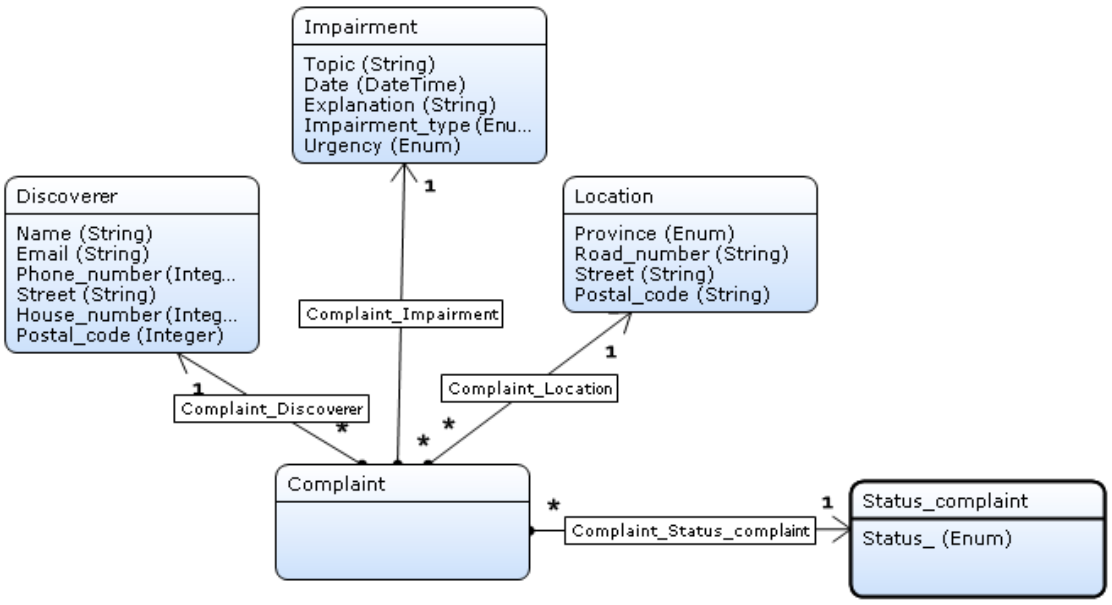


Figure 10 Mendix domain model

Transformation rule 5 describes that a form must be generated and the associated information must be linked. First the Overview form of the PIM is created (figure 11).

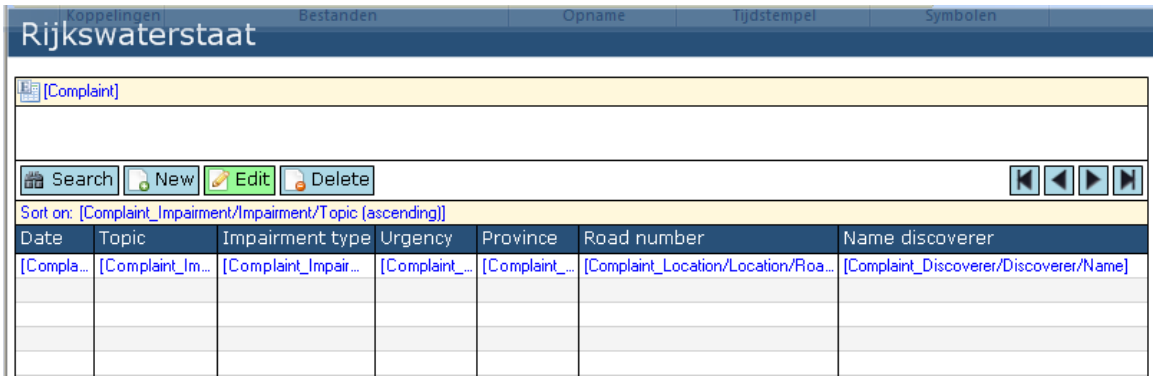


Figure 11 Mendix overview form

Now the Update form can be created, as prescribed by transformation rule 5.2 by automatically creating it (figure 12).

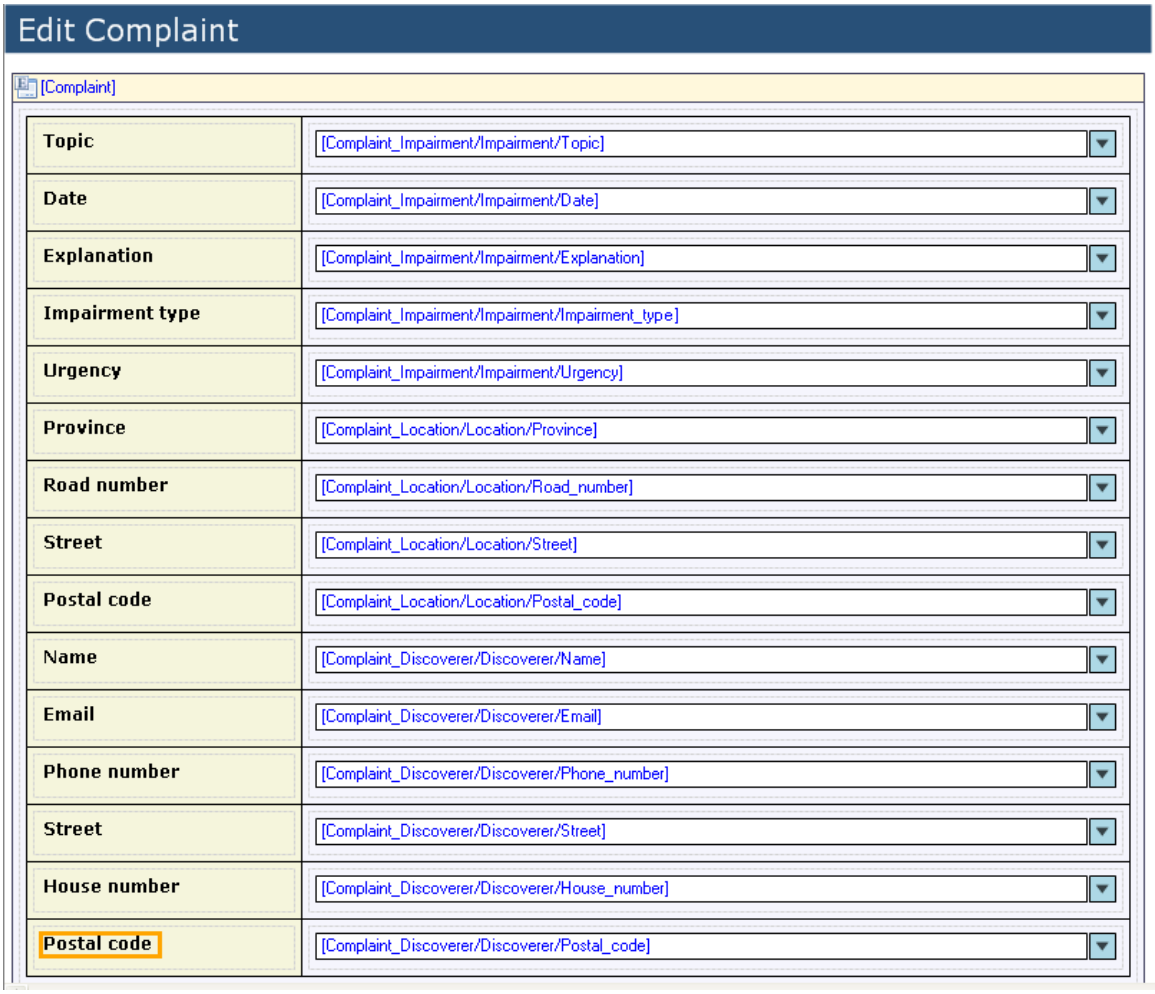


Figure 12 Mendix update form

The third form that is in the PIM is the feedback form (figure 13).

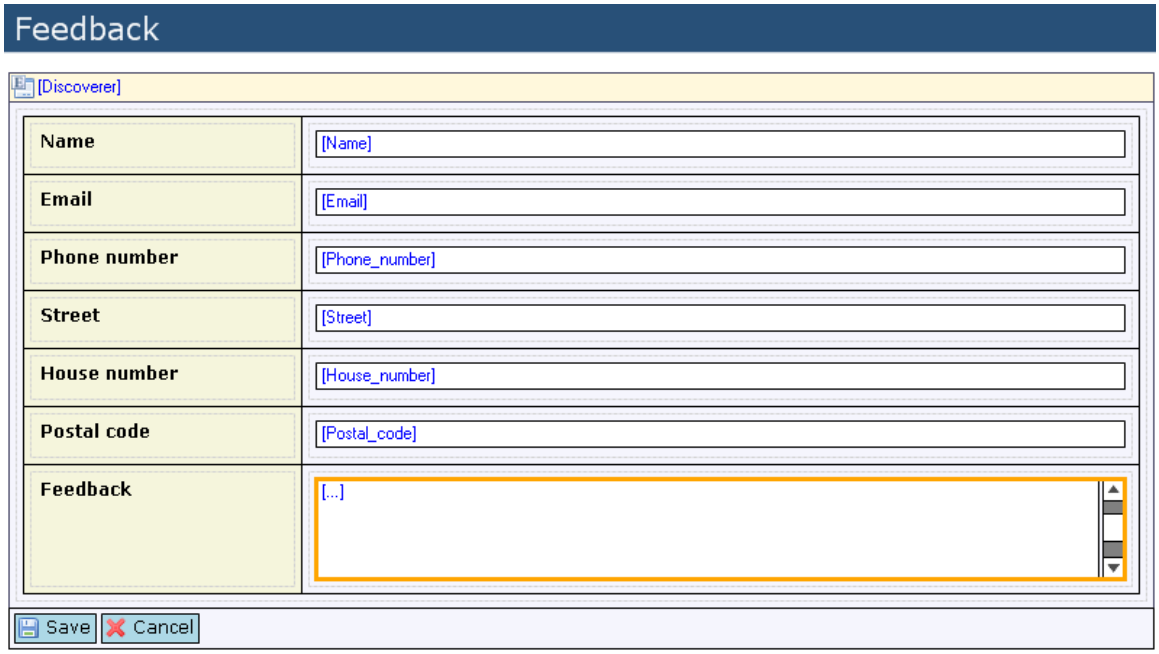


Figure 13 Mendix feedback form

Transformation rule 6 describes how roles can be added by opening the security section, the roles of the Rijkswaterstaat case are added as in figure 14.

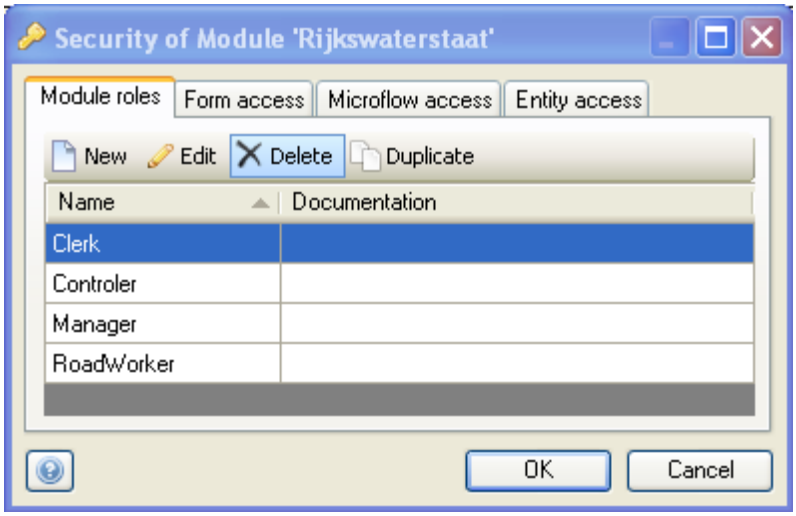


Figure 14 Mendix security section

The only element of the PIM that has not been dealt with so far is the tasks. The following six tasks are displayed in the PIM:

1. Accept complaint
2. Research complaint
3. Sign of repair
4. Decide on action
5. Inform discoverer
6. Archive settled complaint

Accept complaint and research complaint are done by the Clerk; these actions are done by updating the complaint case. Therefore no specific modeling is needed in Mendix; hence the update form already exists. The same accounts for the third and fourth task. Sign of repair must be performed by the road worker. This task is just updating the status field, which can be done in the already created Update form. Decide on action must be performed by the manager, when he has made decision; this is entered in via the update form.
The fifth task, inform discoverer, needs its own form which was already created. The sixed task is again just an update of the status, performed in the update form.

9.2 Be Informed

According to transformation rule 1 for Be Informed a dossierservice needs to be created (for translation of the Be Informed terms check Appendix C). This dossierservice is something specific for Be Informed which always needs to be created. Many other created elements are linked to this dossierservice, it is not associated with any element of the meta-model.
The next step is to create the entities and add the data items, the result is shown if figure 15. In the top row the three entities Discoverer, Impairment and Location are shown. In the right hand overview tab for the entity impairment the data items are shown. For the data item type of impairment the options field is shown in the center. For each data item such an option field must be filled in. This makes clear that Be Informed is more suited to create customized applications, and therefore is harder to create automatically.

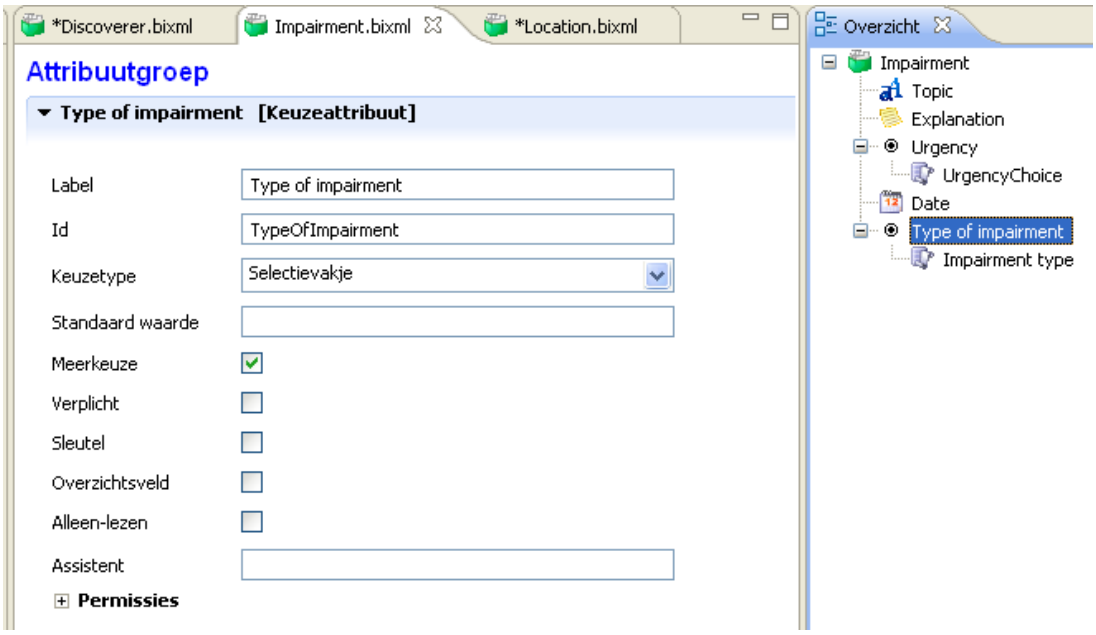


Figure 15 Be Informed entities

The next step is to create a dossiertype (case) where, according to the transformation rules, the different atribuut groepen (entities) and statuses are added to. In figure 16 the coloumn on the right shows the different statuses. The entities are added in a couple of steps, first an event type is create where attribute collection is created, to this attribute collection the different entities are added. In the dossiertype of figure 16 a record type is used to refer to this created attribute collection, again this is visible in the right column of the figure.

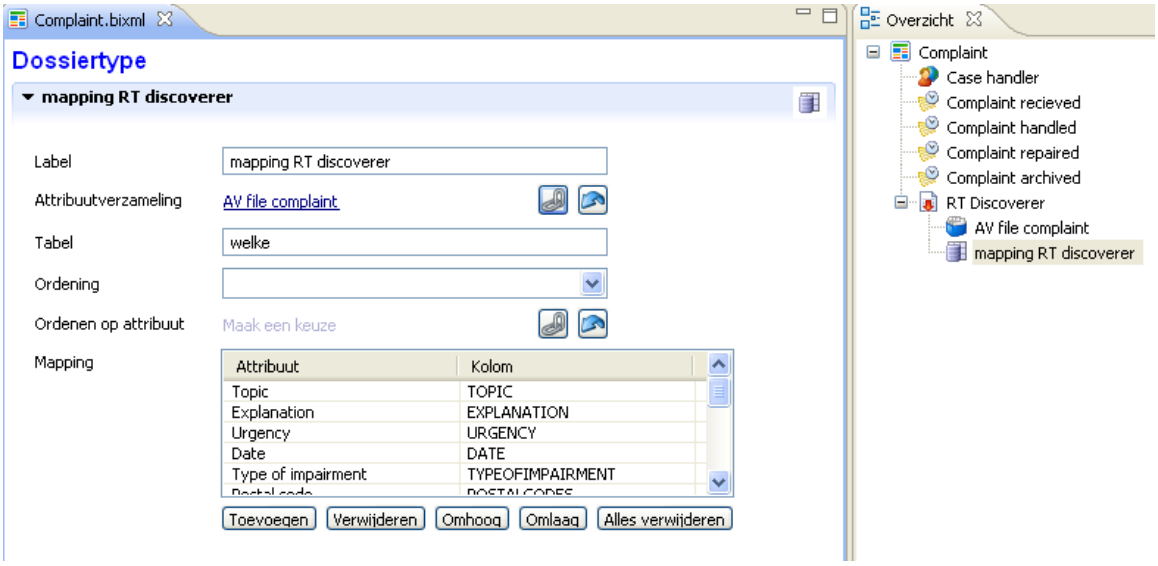


Figure 16 Be Informed dossiertype

According to transformation rule 20 a form needs to be created for each form in the PIM. Figure 17 shows the three forms described in the PIM.

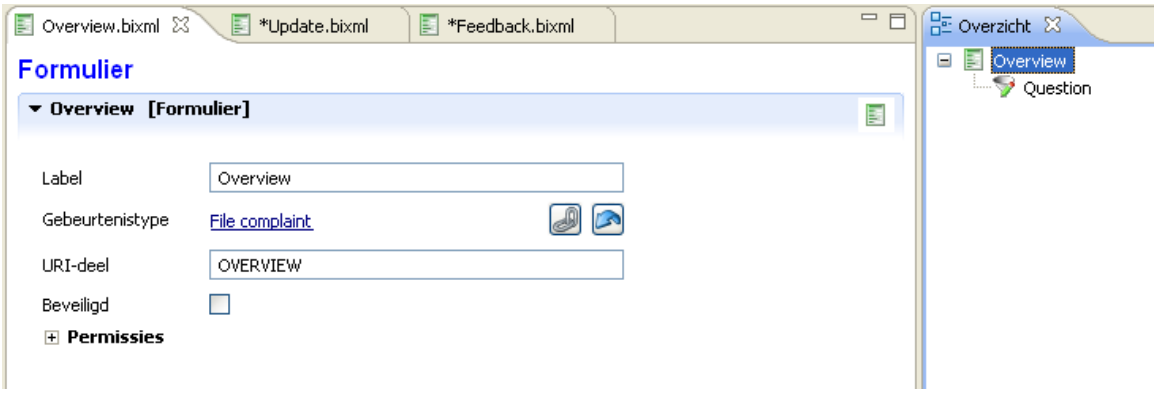


Figure 17 Be Informed forms

The last step to take, that directly links to the PIM, is to add the roles (figure 18).

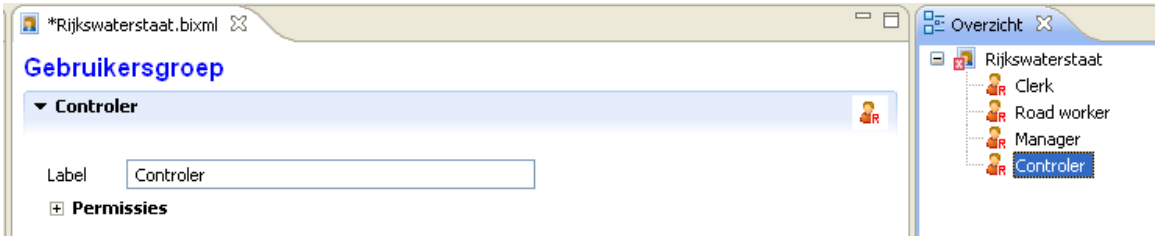


Figure 18 Be Informed roles

Other steps described in the transformation rules of Be Informed are steps that have to be taken to create a running application, but they do not directly relate to elements in the PIM. However these steps can be used to customize the final application. The transformation rules as they are will create a running application where settings are not changed. For instance the element basic style (transformation rule 21) can be used to create a style, you have to have this element. But you can leave it unchanged.

Part 5: Discussion and conclusion

In this part a reflection is presented on the research that has been conducted. First a discussion is held on the value of the result, on the implications of the limited scope, and on what could have been done differently. Second is a discussion about the conducted validation. Also the design science guidelines by Hevner et al. (2004) are discussed. Finally a conclusion on the outcome of this research is presented.

10 Discussion

The value of the presented work is very dependent on the stage of the development of the EAD method. This thesis contains only the specification phase, so in this stage this result has hardly any value to practitioners yet. For science it can be a basis for further research, but again the current value is limited. When however the EAD method would be brought to further development stages and eventually to a fully functioning method it can be a very helpful tool for practitioners. It will allow consultants to create easily, without specific software training, a process supporting application. This in turn will make the creation of these applications cheaper, what will make them accessible to a larger group of firms, which can all benefit from these applications. Experts are positive about the possibilities the EAD method offers, but they acknowledge that further research is needed to make it useful to them. Extending the EAD method by making it comply with other software tools would make the EAD method even more valuable and practically relevant to consultants.

There are some specific points to discuss concerning the EAD method. First of all an attempt was made to involve experts from Mendix and Be Informed in the development of the EAD method. Unfortunately Mendix did not respond to the request. Especially for the development of the black box for Mendix, cooperation of the software provider is desired. Some areas of the EAD method have been left open during this specification phase, and must be dealt with in the development phase. This concerns primarily the modeling of tasks, which has been already discussed in chapter 6. The point is that with tasks much can be done in practice, and the two discussed software tools Mendix and Be Informed both have many options for implementing tasks. It has been proven in chapter 8 that it is possible to implement several tasks. However, fully automatic transformation of a task described in a PIM into a software tool may remain difficult. Further research needs to be done to make this feasible in a later stage. For instance the way information flows automatically from and to other data sources is not dealt with in the current EAD method. And this connectivity to other data sources is often needed in business processes.

Finally what needs to be developed is the application that facilitates the making of a PIM on the basis of the meta-model. The application should guard the application of the well-formedness rules and be able to transfer the PIM to the required software tool. For this transformation the presented rules must then be translated into transformation rules that the application can work with.

Let us recall the vision: *The EAD method allows consultants to model an information intensive business process and use the created model to automatically generate process supporting applications.* The achievement of this vision will lead to a number of improvements for consultants. The consultant can create applications by himself for his customers. This will reduce the lead time and therefore the costs of creating the application. Furthermore the consultant can now directly communicate with the customer and alter the application to his needs, were currently he needs to translate the requirements of the customer to a programmer who has never talked directly to the customer. One translation less means that the likelihood that the application does what the customer wants increases. Something that goes wrong much too often, referring here to the infamous software design process tree-swing comic[5].

10.1 Discussion of validation

On the basis of the long example discussed in part 4 we can conclude that the transformation rules as stated in chapter 7 work for the creation of an application in both software tools (Mendix and Be Informed). One issue is that in the instance of the meta-model it is not possible to state what the values are of a specific enumeration. It is possible to specify that province is an enumeration, but what the possibilities are cannot be specified, e.g. the provinces of The Netherlands. Other process descriptions could also be used to further validate and improve the method, meaning that using just one process description for the validation has limited reliability. A reliable validation was not the objective; this first test just proves that it is well possible for this specific process description.

A limitation of this validation is that the black box is simulated by hand. The PIM is created with standard software (Microsoft Visio) that allows you to create anything. The well-formedness rules are thus not automatically checked, but they are evaluated by hand. Also the transformation of the PIM to the environment of Mendix and Be Informed is done by hand. The described rules are followed strictly, but this is different for a human than for a computer because all the rules are interpreted by the researcher who knows what is meant by them. Other people might find some rules unclear what could lead to different results.

10.2 Design science guidelines

On the basis of the design science guidelines of Hevner et al. (2004) a further discussion is held. Each guideline will be discussed separately.

Design as an artifact; the result of this thesis is the EAD method. Even though the method is not fully developed yet, the artifact is the result of this study. We can say that the guideline is followed.

Problem relevance; as is pointed out in the introduction the scale of savings for a service oriented country as The Netherlands can be considerable. These savings can be realized for a

[5] For instance: http://tamingdata.com/2010/07/08/the-project-management-tree-swing-cartoon-past-and-present/

majority of businesses in the western world. The EAD method contributes to making it simpler to become more efficient as a company. Therefore, if the EAD method would be fully developed, it would be very relevant. But at this stage the EAD method is not ready to be used. So the EAD method is potentially relevant.

Design evaluation; the EAD method is described and specified, but not created as an automated software program, let alone validated. This can be done in the third stage described by Sommerville (2004). What has been done is an attempt to validate the results of this specification phase by discussing it with experts and by working out an example. As it seems we are on the right track, but this validation should have been done in a more rigorous setting. For example, one can test application development by comparing two groups. People in both groups would have the comparable background knowledge in Mendix and Be Informed. Then people in the experimental group would make an application out of a business description by use of the EAD method. People in the control group would make the same exercise, but just using what they know. Now you can compare for instance speed and quality (number of errors and does the application what it is supposed to do). This more rigorous validation was not done because at this stage of the development of the EAD method the results will lead to limited proof. When the EAD method is fully developed such a comparison can really prove the potential gain consultants and businesses can experience.

Research contributions; as stated before, the contribution of the EAD method is depending on the further development of the method. However we believe that the first step made in this thesis is a solid basis for further development, and if further developed the EAD method is contribution to existing methods. A note that can be made here is that it was expected that workflow models described in literature would be ahead of developed software tools. This turned out not to be the case Existing software like Be Informed is already a powerful tool for creating case centric applications.

Research rigor; the specification of the EAD method is based upon MDA by OMG. Therefore the resulting EAD method is satisfying standards in the field of information systems. Also scientific literature is used and iterations were conducted. Therefore the construction of the EAD method can be stated as rigorous. The evaluation however was not very rigorous. As mentioned before evaluation can only be done meaningfully in a later phase.

Design as a search process; the available means (scientific literature, existing methods as in MDA, existing software tools, experts from practice and experts from science) where all used in the development of the EAD method. By following the MDA framework the rules of the field are satisfied, which is a good thing for practical implications of the EAD method.

Communication of research; this thesis will be published in the library of the Eindhoven University of Technology, thereby creating the communication of the EAD method to the scientific field. By also presenting the report to the consultancy firm O&i the EAD method is brought to practice. Reporting the results in a magazine that is read by other consultants would put the results in reach of a larger audience. This will not be done, because it is considered too early to publish the EAD method, not being fully developed.

11 Conclusion

The described EAD method for the development of an application for information intensive processes has gone through the specification phase. The vision is stated and first steps on the route to the achievement of that vision are taken. As is outlined in the discussion, strong points are:

1. The designed model is a clear and well based start for further development.
2. The EAD model is a major contribution to existing methods.
3. When further developed and tested, the model-based applications will have strong relevance for enhancing efficiency of all information intensive businesses.

Weak points are:

1. In the scope of this thesis it was not possible to create an actual automated software program. Therefore the ultimate value of the EAD model cannot be evaluated at this stage.
2. At this stage scientific communication is premature and therefore limited.

When in the future the EAD method is ready for consultants to use, they can experience easy application development. This results in faster development times and with that faster return on the investment. When the creation of an application is easy, it also becomes less expensive, and therefore more companies can benefit from process supporting applications. Therefore the potential practical value of this thesis is considerable.

11.1 Further research

The discussion outlined what elements need further research. Before an application can be developed that supports the EAD method, the task element needs to be further researched. What are all possible tasks? How does each task translate to a specific software tool? When this is done a first application can be developed, which must then be validated. A good thing to research first is what must the application of the EAD method by able to do, and how should it operate? When this is done, other software tools can be added in the evaluation phase.

It would be interesting to compare applications created by the EAD method to applications created in the conventional way on quality of the result and development time and costs.

References

Ader, M. (2000). Workflow comparative study. Workflow and Groupware Strategies.

Bhattacharya, K., Hull, R., & Su, J. (2009). A data-centric design methodology for business processes. 503--531.

Bradley, J. (2008). Management based critical success factors in the implementation of Enterprise Resource Planning systems. *International Journal of Accounting Information Systems, 9*(3), 175--200.

Cohn, D., & Hull, R. (2009). Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull, 32*(3), 3--9.

Garcia, H. (2011). Evaluation of Data-Centric Process Modeling Approaches.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *Mis Quarterly, 28*(1), 75--105.

Hull, R. (2008). Artifact-centric business process models: Brief survey of research results and challenges. *On the Move to Meaningful Internet Systems: OTM 2008*, 1152--1163.

Kappel, G., Rausch-Schott, S., & Retschitzegger, W. (2000). A framework for workflow management systems based on objects, rules and roles. *ACM Computing Surveys (CSUR), 32*(1es), 27.

Kleppe, A., Warmer, J., & Bast, W. (2003). *MDA explained: the model driven architecture: practice and promise.* Addison-Wesley Longman Publishing Co., Inc.

Marchetti, A., Tesconi, M., & Minutoli, S. (2005). Xflow: An xml-based document-centric workflow. *Web Information Systems Engineering--WISE 2005*, 290--303.

Nigam, A., & Caswell, N. (2003). Business artifacts: An approach to operational specification. *IBM Systems Journal, 42*(3), 428--445.

Object Management Group. (2003, 06 12). *MDA guide.* Retrieved 05 10, 2012, from Object Management Group: www.omg.org/mof

Object Management Group. (2011, 08 05). *Unified Modeling Language V2.4.1.* Retrieved 05 10, 2012, from Object Management Group: http://www.omg.org/spec/UML/2.4.1/

Pesic, M., Schonenberg, H., & van der Aalst, W. (2007). Declare: Full support for loosely-structured processes. *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, (pp. 287--287).

Pine, B., & Davis, S. (1999). *Mass customization: the new frontier in business competition.* Harvard Business School Pr.

Redding, G., Dumas, M., ter Hofstede, A., & Iordachescu, A. (2010). A flexible, object-centric approach for business process modelling. *Service Oriented Computing and Applications, 4*(3), 191--201.

Reijers, H., Limam, S., & van der Aalst, W. (2003). Product-based workflow design. *Journal of Management Information Systems, 20*(1), 229--262.

Ross, R. (2012, 04 23). *Business agility vs. agile in software development: not related!* Retrieved 05 20, 2012, from Ron Ross on business rules: http://www.ronross.info/blog/2012/04/23/business-agility-vs-agile-in-software-development-not-related/

Sommerville, I. (2004). An introduction to software engineering.

Stohr, E., & Zhao, J. (2001). Workflow automation: Overview and research issues. *Information Systems Frontiers, 3*(3), 281--296.

Tsourveloudis, N., & Valavanis, K. (2002). On the measurement of enterprise agility. *Journal of Intelligent & Robotic Systems, 33*(3), 329--342.

van der Aalst, W., & Pesic, M. (2006). DecSerFlow: Towards a truly declarative service flow language. *Web Services and Formal Methods*, 1--23.

van der Aalst, W., Pesic, M., & Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science-Research and Development, 23*(2), 99--113.

Voss, C., Tsikriktsis, N., & Frohlich, M. (2002). Case research in operations management. *International journal of operations \& production management, 22*(2), 195--219.

Zachman, J. (1987). A framework for information systems architecture. *IBM systems journal, 26*(3), 276--292.

Appendix A: Literature overview

The techniques are selected because they give a good representation of what is available in the field of information intensive processes. A division in the techniques can be made between object lifecycle, data driven, document based, and bridging methods (Garcia, 2011). BALSA and FlexConnect are object lifecycle methods, product based workflow design is a data driven technique, XFLOW is a document based technique, and $TriGS_{flow}$ is a bridging technique. Not all techniques are supported by an execution engine, they are just designed on paper. Table 7 summarizes the techniques that will be discussed, the first row gives the name of the technique, the second row the reference to the article that describes the technique. The third row gives the modeling language that is used, not all techniques are linked to a specific language. The fourth row gives execution engines that can be used for the technique, two techniques are only supported by a prototype. The fifth row shows to what category each technique belongs.

Table 7 artifact centric workflow techniques

Name	Described in	Language	Execution engine	Group
BALSA	Bhattacharya et al., 2009		IBM WebSphere	Lifecycle
FlexConnect	Redding et al., 2010	UML	FlexConnect	Lifecycle
PBWD	Reijers et al., 2003	PDM, YAWL	ProM, ExSpect	Data driven
Xflow	Marchetti et al., 2005	XML	Prototype	Document based
TriGS$_{flow}$	Kappel et al., 2000		Prototype	Bridging

BALSA

BALSA is method that is object lifecycle method developed by Bhattacharya et al. (2009), it is an acronym where each letter stands for an element that is present in the final model. The elements are Business Artifact with Lifecycle, Services, and Associations. This is transformed into four elements discussed in more detail below.

Business artifact information model

In general you specify all the artifacts and their elements in a *business artifact information model*. The artifact data should incorporate the information needed to (i) capture business process goals, and (ii) allow for evaluating how thoroughly these goals are achieved (Bhattacharya et al., 2009). The data in an artifact usually come from the outside world, or are a result of executing a task.

Often information in the artifact is linked to another artifact. In the case of ACP the sender is an element of the package artifact, but more information about the sender is an element of the sender artifact. Usually artifacts are linked by an identification number.

Business artifact lifecycle

The business artifact lifecycle describes which stages are possible for the artifact, from

introduction to final disposition and archiving. This information is important since it sets limitations to the possible stages an artifact can be in. Without defining the possible states the artifact can be anything, resulting in a situation where an artifact will never be finished. As mentioned before the life expectancy of an artifact can vary. Permanent artifacts have different lifecycles than short lived artifacts (Hull, 2008). In the running example of ACP the package artifact is typical short lived artifact, it comes to live when an order is created and end when the delivery is performed. The customer artifact would be long lived. As long as ACP has a relation with the customer that artifact is valid. A permanent artifact is an artifact that states for instance inventory levels. It changes often, but the artifact will always be there.

Services

A service (i.e. a task) makes a change to one or more artifacts. The word service is used to emphasize that the artifact centric workflows are used in service oriented environments. The changes are however usually restricted by constraints, for instance business rules or flowcharts stating in what order the service must be conducted. Two aspects can be identified in which the work is meaningful to the whole business process. First, the potential changes made by the service should reflect a measurable step of progress towards the business goal. Second, the division of the business process into some collection of services should be able to accommodate administrative organization structures, IT infrastructures, and customer-visible status (Hull, 2008; Bhattacharya et al., 2009). When you have defined services, you can create a workflow by associating the different services. For the running example of ACP services can be: receive transport request, send invoice, receive payment, determine route to recipient, and deliver package.

Associations

The services make changes to artifacts that are restricted by constraints. The constraints can come from a procedural specification like a flow chart, but also from relationships among the services, between services or external events. Associations are given in a declarative way, using Event-Condition-Action (ECA) rules (Bhattacharya et al., 2009). For example when an order arrives, and the customer has paid, than deliver the package.

BALSA method is constructed with the support of the IBM Watson research laboratory. It is possible to model BALSA in the execution engine IBM WebSphere.

FlexConnect

FlexConnect is an object-centric process modeling framework and notation, designed to support highly flexible processes. Case-by-case variations and exceptions are today's norm; FlexConnect

supports this type of flexibility. The framework consists out of three predefined abstract types of business objects (Redding, Dumas, ter Hofstede, & Iordachescu, 2010):

1) COROB: Coordination object that is responsible for the creation and synchronization of tasks needed to complete a process, managing the execution of a process and referring out of scope work to other COROB's.
2) JOB: Job Objects responsible for the execution of a task, and report on task completion to its parent object (a COROB).
3) ROB: Referral object that allows a COROB to refer out of scope objects to another COROB.

A FlexConnect process model consists of a number of COROB, JOB, and ROB elements and their relations. Each object type has a lifecycle, which is captured by means of a state machine. They may be connected with signals. Two types are distinguished: static signals and dynamic signals. Static signals are automatic; they react on events that occur during execution. They can either be a start, finish, or massage signals. Dynamic signals occur at a certain point in time, their occurrence is entirely controlled by users. Four types are distinguished: delegation, creation, referral, and nesting.

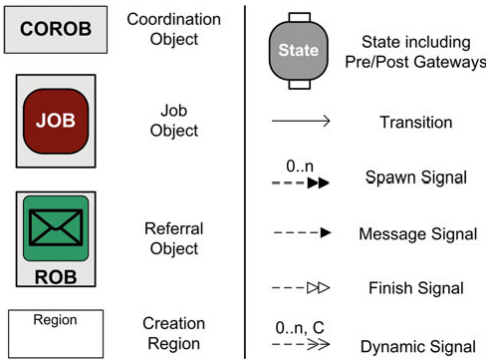


Figure 19 FlexConnect core model elements (Redding et al., 2010)

FlexConnect is thus modeled with the elements shown in figure 16, but behind those elements a UML diagram is created. The semantics are defined by means of an interpreter specified as a colored petri net. A FlexConnect tool is created that allows one to make graphically a FlexConnect model.

Product based workflow design

The data driven method Product Based Workflow Design (PBWD) is a technique that is suitable for the (re)design of administrative processes (Reijers, Limam, & van der Aalst, 2003). Obviously administrative processes are processes where one can find many artifacts. The idea comes from the manufacturing world; there a bill of materials (BOM) is used to specify what elements must be in which quantity in the final product. In PBWD a similar schema is made to that of the BOM, but now specifying the data elements that are required to make the final judgment. The model consists of the following elements (Garcia, 2011):

- A set of data elements with a special top element i.e. the final data element or final product.
- Production rules indicating the different ways in which a data element can be produced based on the value of other information elements.
- A set of constraints that apply to each production rule: a production rule can only be applied if its correspondent constraints evaluate to true.
- The cost of using each production rule.
- The time that takes to use a production rule.
- The probability that each production generates an acceptable result.

For the modeling a clean sheet approach is used, not considering any existing models, but starting from scratch. The method distinguishes four phases (Reijers et al., 2003):

1. Scope: The workflow that will be subject to the (re)design is selected; objectives and limitations are identified.
2. Analysis: A study of product specification leads to its decomposition into data elements and their logical dependencies. If there is an existing workflow, this might be a source for finding data elements.
3. Design: Based on the objectives identified in phase one, the product specification decomposition and their logical dependencies, one or several alternative workflows are derived. A workflow structure consists of tasks that retrieve or process data elements.
4. Evaluation: The created workflow structures are verified, validated with end users, and their estimated performance is analyzed in more detail. This information is used to select the design to be implemented.

Product Data Modeling (PDM) and Yet Another Workflow Language (YAWL) are languages that support PBWD. PROM and ExSpect are tools where it is possible to model a PBWD (Garcia, 2011).

Xflow

The document based method Xflow is described by Marchetti, Tesconi, and Minutoli (2005). They state that a document workflow is a particular workflow where all activities made by the actors are document compilations. In many businesses today a document runs through different departments where different actors have to fill in a section of the document. Xflow aims at

streamlining this type of processes. A state diagram describes which are the possible statuses a document can be in, compilations by actors can change the status of a document. Four different statuses are identified:

1. Creation; a new instance of document is created, with several information attached.
2. Pending; waiting to be compiled.
3. Processing; an actor compiles his part of the document.
4. Freezing; the actor is not able to compile the document at the current moment.

The information of each document is isolated in a XML document.

In order to fully describe a document workflow in Xflow four elements must be described (Marchetti et al., 2005):

1. Schema; an overview of all documents involved in the document workflow.
2. Role chart; a set of all internal and external actors that operate in the document workflow.
3. Document interface description; an interface that can be used by external actors to access the documents.
4. Document workflow; defining for a document all paths that it can follow in its lifecycle, the activities and policies for each role.

Even though Xflow is only supported by a prototype engine, the fact that it works with XML documents for communication between systems makes it a method with the potential to be used in the field. Further developments on Xflow might bring the technique to usage in practice, however the paper was published in 2005 and since then no such effort has been taken.

TriGS$_{flow}$

The bridging method *TriGS$_{flow}$* is an object-oriented framework for the construction of WfMS. The focus is upon the balance between reusability and adaptability. To reach this objective, *TriGS$_{flow}$* integrates three basic techniques (Kappel, Rausch-Schott, & Retschitzegger, 2000):

1. An object-oriented database system is used to build a generic workflow model providing both database functionality and possibilities for modeling, reusing, and customizing complex business domain objects.
2. To cope with changes in the personnel, a role model has been integrated into the object-oriented environment in order to decouple activities from particular persons.
3. Event/Condition/Action rules (ECA rules) are used to allow for a flexible coordination of activities as well as of resources needed to perform these activities.

TriGS$_{flow}$ is only supported by a prototype that is created by the authors of this first paper on the model.

Declarative Workflows

A declarative workflow is a workflow that does not state what must be done, but it states what must not be done. In the figure below you can see that under situation (b) the procedural workflow defines a control flow that lies within the allowed area. Situation (c) declarative workflow does not define the actual control flow, but defines the limits by constraining the allowed area. This will create much more flexibility in executing the workflow.

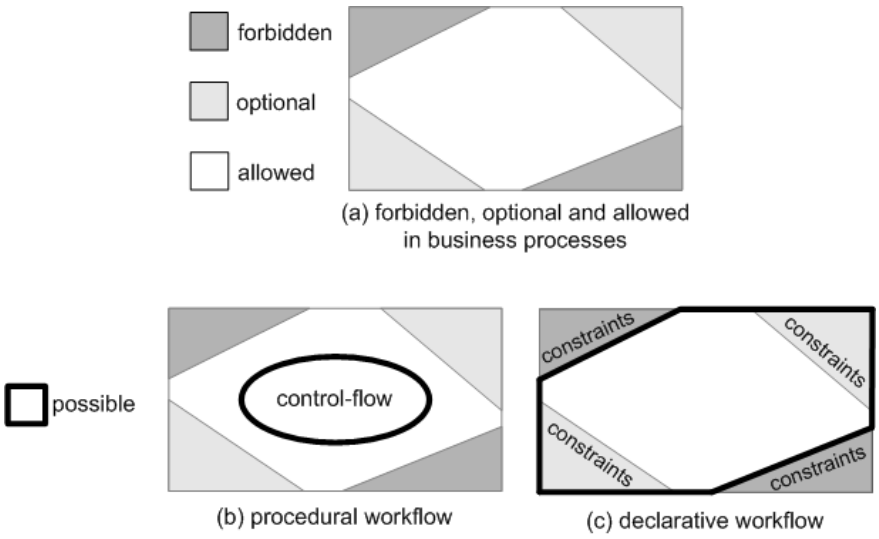


Figure 20 procedural vs declarative workflow (van der Aalst, Pesic, & Schonenberg, 2009)

In workflow management systems there is a paradox: you want process support and flexibility. Because processes support can improve your run time in the short run and flexibility can improve run time in the long run. With the current systems the combination is hard to achieve, since information systems can vary from structured to unstructured. In the latter you will have more flexibility, in the first you will have much better process support. With declarative workflows it is possible to balance support and flexibility (van der Aalst, Pesic, & Schonenberg, 2009).

The only possible way to model a declarative workflow is to use DECLARE. You can do this by using the modeling language ConDec or DecSerFlow. YAWL and PROM are execution engines that support these languages (van der Aalst et al., 2009). Today there are no other techniques described in literature that fully work on a declarative basis.

DECLARE

DECLARE is developed as a constraint-based system and uses a declarative language based upon temporal logic for the development and execution of process models. DECLARE is a declarative system, however it also offers features of traditional WfMSs like model development, model verification (finding errors in models), automated model execution, changing models at run-time, analysis of already executed processes, and process decomposition (Pesic, Schonenberg, & van der Aalst, 2007). DECLARE uses three elements which can be found in many WfMSs:

1. Framework, a modeling tool used for system settings and process model development.
2. Designer, a tool for process enactment, also used for communication with YAWL and ProM, and changing models at run time.
3. Worklist, a tool for process execution.

Figure 21 shows a standard architecture for DECLARE. As can be seen two others tools are used as well. YAWL is used to model the structured parts, and DECLARE is used to model the unstructured parts. The process mining tool ProM is used for analysis of the past executions of DECLARE.

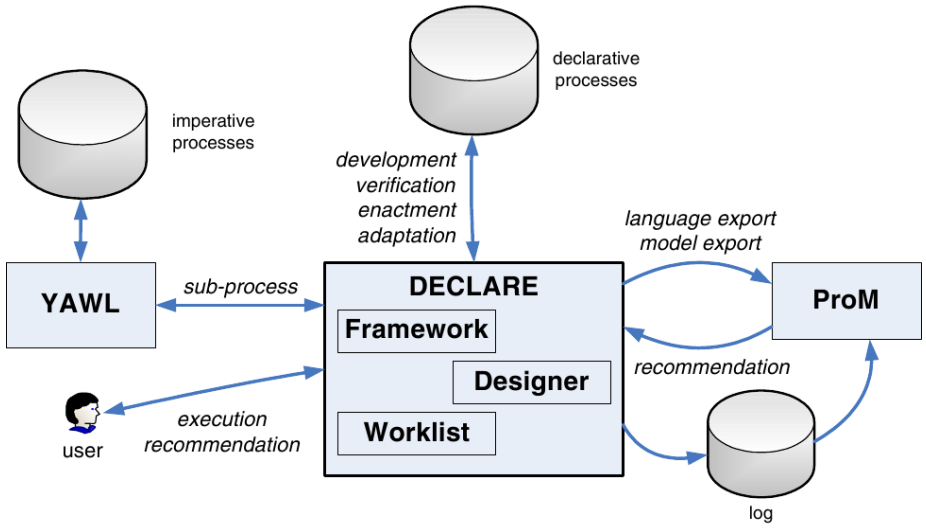


Figure 21 DECLARE system architecture (Pesic, Schonenberg, & van der Aalst, 2007)

In DECLARE Linear Temporal Logic (LTL) is used to specify the constraints (Figure 22). Since LTL is difficult language to understand for non-specialists DecSerFlow (Declarative Service Flow Language) can be used. DecSerFlow is an executable language that can be used to specify hard and soft constraints, which is very useful when modeling in a declarative style (van der Aalst & Pesic, 2006).

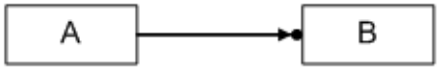
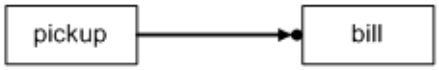
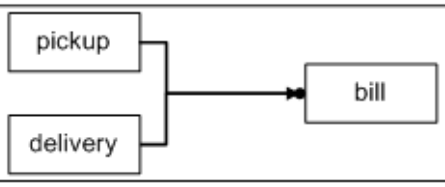
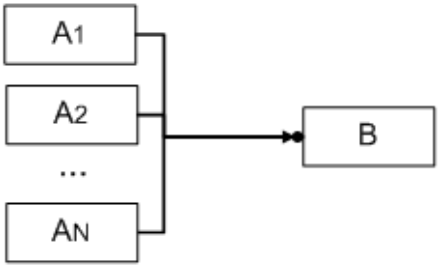
	GRAPHICAL	LTL FORMULA
TEMPLATE		*(! A) W B*
'PLAIN' CONSTRAINT		*(! bill) W pickup*
BRANCHED CONSTRAINT		*(! bill) W (pickup V deliver)*
BRANCHED CONSTRAINT TO MULTIPLE TASKS		*(! B) W (A1 V A2 V … V AN)*

Figure 22 LTL formula (van der Aalst & Pesic, 2006)

Appendix B: Interviews

Name	Function	Modeling experience	Date	Topic
Charlotte Bouvy	Sr. consultant	16 years	10 April 2012	Meta-model, phase one
Eyad Sabbah	Managing consultant	4 years	2 April 2012	Meta-model, phase one
Marjan Kerkhofs-Gerritsen	Information analyst	4 years	16 April 2012	Meta-model, phase one
Marco Comuzzi	Assistant Professor	-	Multiple occasions	Meta-model, phase one
Philip de Lang	Sr. consultant	17 years	9 May 2012	Meta-model, phase two
Pieter van Gorp	Assistant Professor	-	6 June 2012	Well-formedness rules
Eyad Sabbah	Managing consultant	4 years	Multiple occasions	Meta-model, phase two, WFR, Rules
Marco Comuzzi	Assistant Professor	-	Multiple occasions	Meta-model, phase two, WFR, Rules

Appendix C: Translation of Be Informed terms

Be Informed	English
Attribuutgroep	Attribute group
Attribuut	Attribute
Attribuutverzameling	Attribute collection
Basisstijl	Basic style
Beginhandeling	Start action
Dossierservice	Dossier service
Dossier tab	Dossier tab
Dossiertype	Dossier type
Eindhandeling	End action
Formulier	Form
Gebeurtenistype	Event type
Gebruikersgroep	User group
Keuze attribuut	Enumeration
Recordtype	Record type
Referentie	Reference
We applicatie	Web application

Appendix D: Short example (PIM)

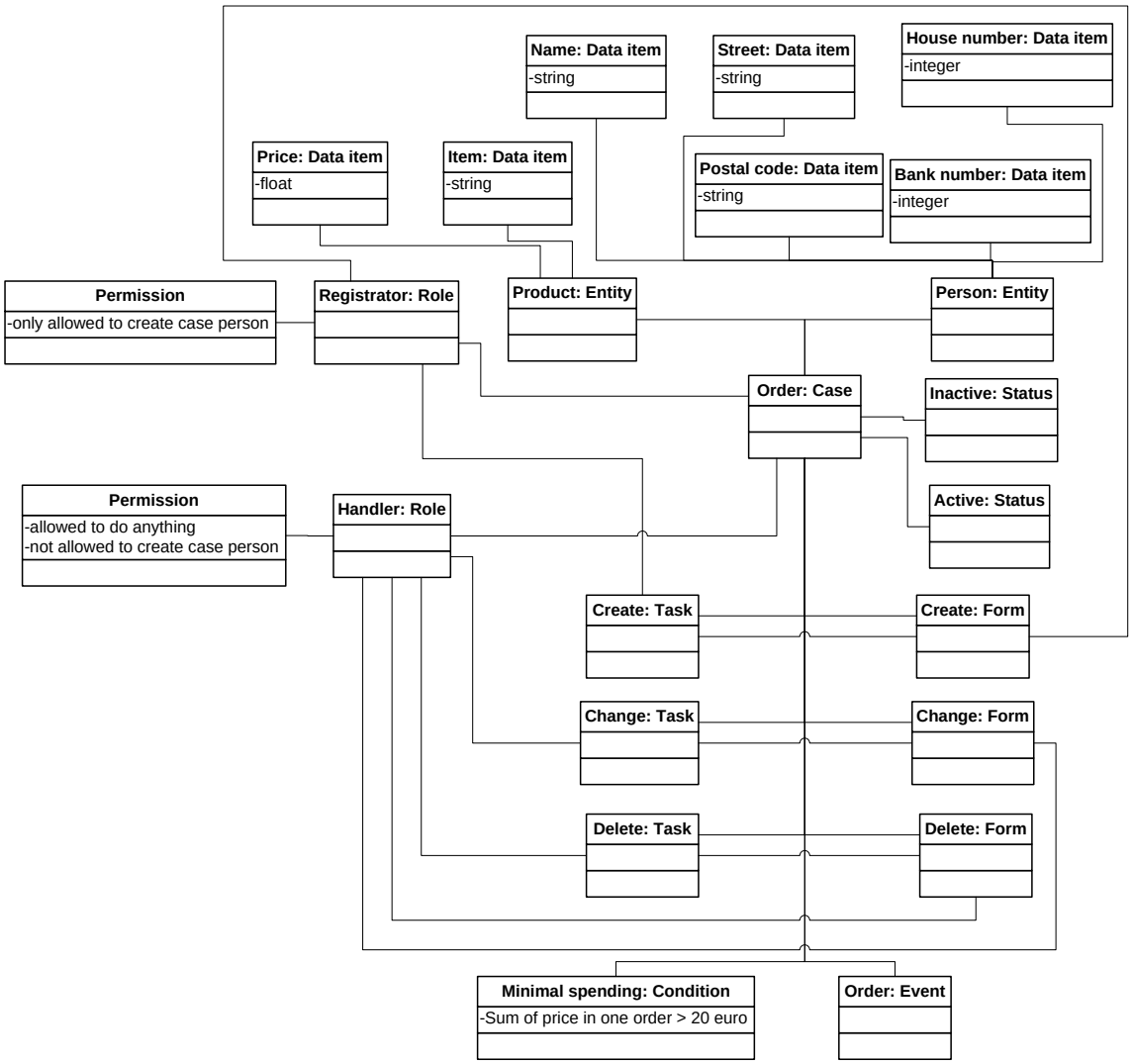


Figure 23 PIM of short example

Appendix E: Short example (Mendix and Be Informed)

Mendix

Each microflow basically contains a start and an end event where one or more action activities are placed in between. In order for the microflow to relate to information from the domain model, a parameter is added and linked to the appropriate entity in the domain model, hence this is the association between case and task in the meta-model. There are 7 possible types of activities (figure 1), each with a couple of alternatives. The possible types are: Object, List, Action call, Variable, Client, Integration, and Logging. Often the microflow is started with the object activity Retrieve, this imports the required information into the microflow and it is now for instance allowed to make calculations or to manipulate the data.

One more option within the microflow is to split the retrieved information, for instance when you want to calculate the average spending's of just the males in your data set. This is done with a split, if a split is used a merge needs to be inserted at some point in order to bring each flow back together.

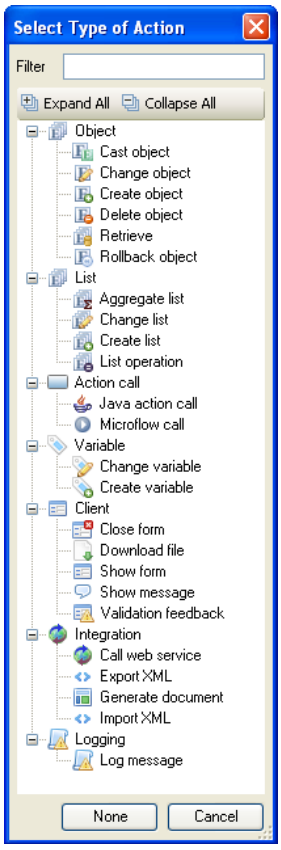


Figure 24 possible action activities in Mendix

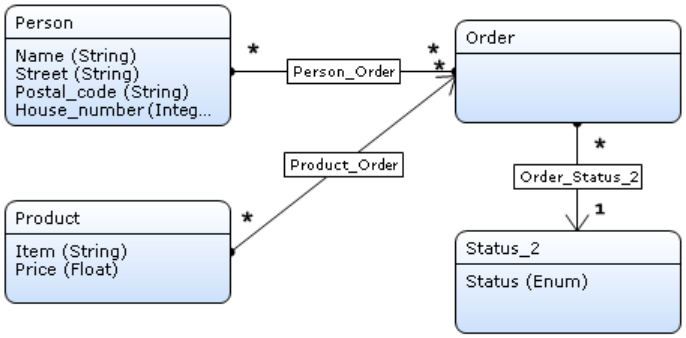


Figure 25 the domain model in Mendix, containing the Case, Entity, Data items, and Status

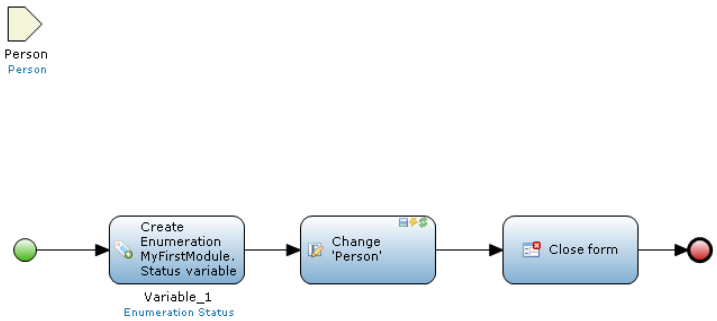


Figure 26 a microflow

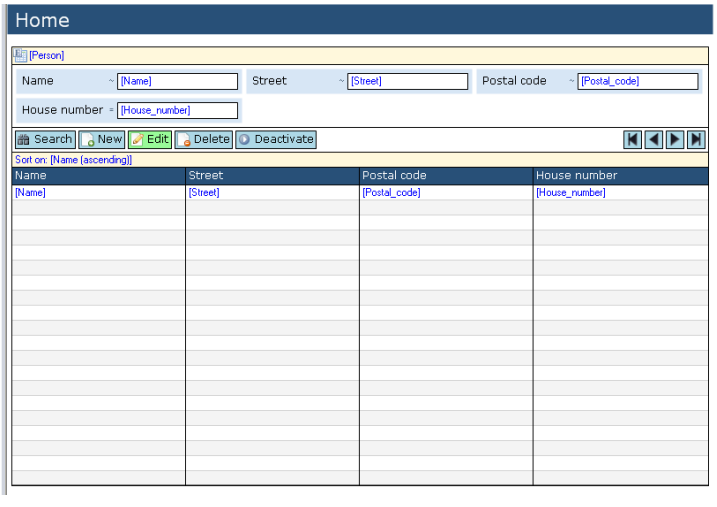


Figure 27 home form with the buttons Search, New, Edit, Delete, and Deactivate

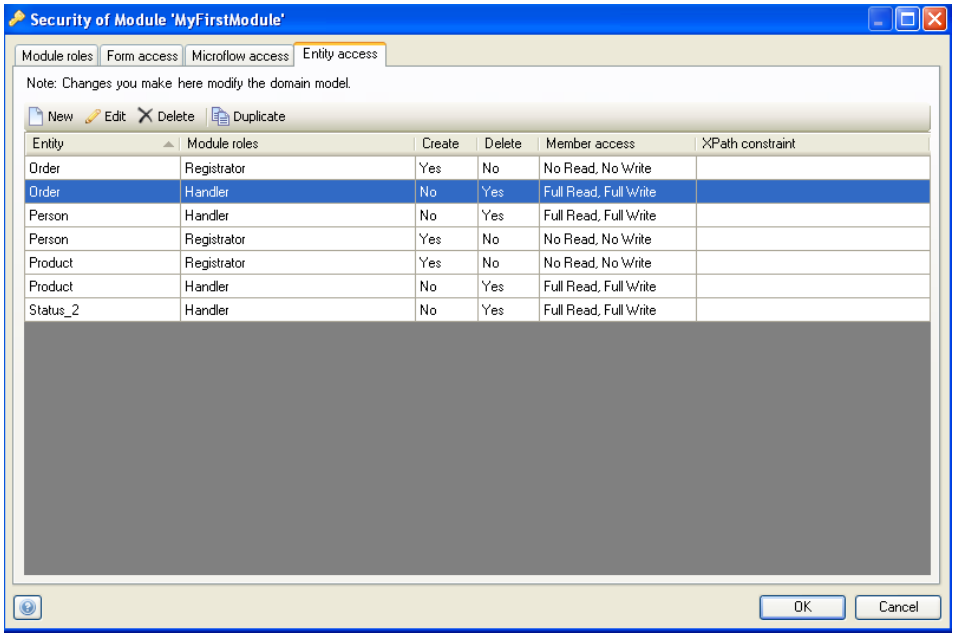


Figure 28 permissions for Registrator and Handler

Be Informed

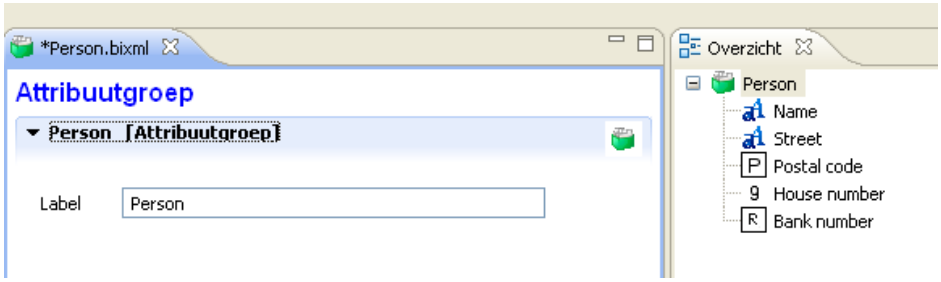


Figure 29 person entity with its data items

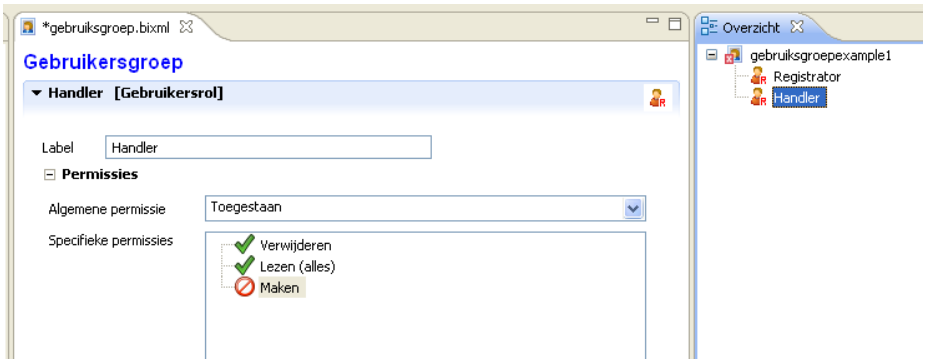


Figure 30 role handler with its permissions

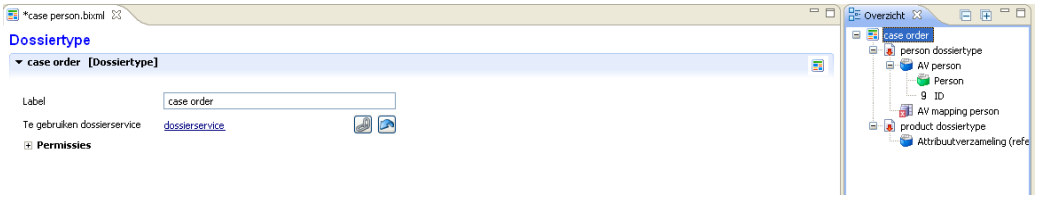


Figure 31 the case order

Appendix F: Long example: process description

A complaint handling process.

Problem Solved (PS) is a company that delivers tailored software solutions for business who want to receive feedback from their customers. Recently PS was asked by Rijkswaterstaat to create a feedback system for the Dutch highways. The users of the Dutch roads must be able to give feedback on bottlenecks, bad roads, broken signs and all other road related items. The feedback system must thus be a passive system, i.e. road users must be able to express complaints or defaults. The idea is that all road users see more than the employees of Rijkswaterstaat who also survey the roads. Another positive point is that some impairments may seem harmless, but do annoy road users, these can now be addressed and fixed faster.

The complaint handling process consists out of several phases which each contain a couple of steps:

- Receive complaint
    - Accept complaint
    - Register complaint
    - Circulate complaint
    - Inform discoverer
- Handle complaint
    - Research complaint
    - Determine urgency of complaint
    - Give feedback to discoverer
    - Circulate complaint (if necessary outside of Rijkswaterstaat)
- Settle complaint (Road, Sign or Barrier)
    - Receive the complaint
    - Make the repair
    - Sign of on the repair
- Settle complaint (Safety or other)
    - Decide on action
    - Inform discoverer
- Check repair
    - Check the repair (if not sufficient, send for rework)
    - Inform discoverer
    - Archive settled complaint

For each phase and step roles are defined that can perform the task, for each phase a role is appointed that is responsible for the process.
When a discoverer has filed the complaint a clerk receives and handles the complaint. If it is about a road, sign or barrier the complaint is forwarded to a road worker. If the complaint is about an unsafe situation or any other impairment a manager of the major rework department

will decide what to do next. If a repair has been executed a controller checks the repair. If it meets to the standard the discoverer is informed. Hence, how the repair is resolved is not recorded in the system, just that the complaint is resolved or not.

The data that is recorded for each complaint is: information about the impairment (topic, date, explanation, type of complaint, urgency, location), information about the discoverer (the name of the person who registered the impairment) (name, email, phone number, street, house number, postal code). It must be possible to add a picture to the notification. For location the following elements can be stored: province, road number, road/street name, and postal code. It must be possible to group complaints concerning the same location. Notice that the public does not create an account for making a notification, the fear of Rijkswaterstaat is that this would create an obstacle for making notifications.

The system will be promoted for all roads in the Netherlands for communication reasons, but not all the roads are the responsibility of Rijkswaterstaat, therefore if a notification applies to an impairment that must be fixed by a municipality the notification will be forwarded to the appropriate municipality. The discoverer is also informed by email that the notification is forwarded.

If more notifications come in about the same impairment they must be grouped in order to prevent doing double work. When a larger number of notifications come in on one location, the urgency level must be upgraded. The possible urgency levels are as traffic lights; green, orange, and red, where red is the highest urgency level.

The possible types of impairments are:

- Road (holes, cracks, or other pavement related impairments)
- Sign (missing, broken, outdated, or other road signs related impairments, including traffic lights)
- Barrier (problems with crash barriers, sound barriers, fences, or other material near the road)
- Safety (general unsafe situations, for instance unsafe junctions)
- Other (any other type of impairment)

After a decision has been made on the urgency level, the type of impairment, and the location have been determined the notification needs to be forwarded to the right party which must fix the impairment. This can be a municipality, Rijkswaterstaat road work department for small repairs, or the department for major rework and maintenance.

A notification can thus be in four different statuses: received, handled, repaired, and archived.

The discoverer of an impairment receives an update by email when the status of a notification changes. With the final step a letter is sent the inform the discoverer that the work has been handled or that the impairment will not be resolved in the near future, if relevant, a picture can be send to the discoverer showing the fixed impairment.

When an impairment has been fixed this will be reported by the person responsible for the repair. Before the discoverer can be informed a controller checks if the repair was sufficient. If this is not the case the rework of the impairment is required. If the repair is fixed correct this will be registered in the system. The final step of informing the discoverer and archiving the complaint can be taken.