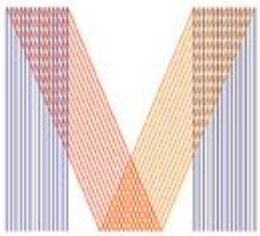
Implementing A Case-handling System: Supporting the Design Process

Master Business Informatics – Thesis



Vellekoop & Meesters

informatie, organisatie en it

Universiteit Utrecht

Evert Eckhardt – 3469689 – A.E.Eckhardt@uu.nl

4 August 2015

Abstract

Case handling is a popular paradigm within municipalities and governmental organizations in The Netherlands. However, there is a significant gap between scientific literature and the implementation of Case-handling Systems at these organizations. Based on a literature study and semi-structured interviews at organizations, this research presents several best practices and introduces the Declarative Modular Case Modelling technique in order to support organizations with their implementation of a Case-handling System. The technique is tested in a case study performed at the Nederlandse Voedsel- en Warenautoriteit, which is at time of writing in the process of implementing such a system. The technique presents a generic modular approach to the modelling of processes and case types, focusing on the reusability of process fragments between different case types. The modules are gathered in a repository on which the specific implementations are based. With the introduction of the modelling technique, best practices and a definitive definition of the case handling paradigm, this research bridges the gap between the academic and organizational aspects of case handling.

Acknowledgements

"Satisfaction lies in the effort, not in the attainment, full effort is full victory."- *Mahatma Gandhi*

I wish to express my thanks to all people involved in my thesis project, my supervisors from Utrecht University and Vellekoop & Meesters for providing insights and feedback on the project and my fellow students for their suggestions during the colloquium presentations.

A special mention goes out to my family, friends and girlfriend for their love and support during the duration of this research project, thank you all.

Contact information

A.E. (Evert) Eckhardt
Graduating student
3469689

A.E.Eckhardt@uu.nl

Utrecht University
Department of Information and Computing Sciences
Buys Ballot Building
Princetonplein 5, De Uithof
3584 CC, Utrecht

dr. ir. J.M.E.M. (Jan Martijn) van der Werf
First internal supervisor

J.M.E.M.vanderWerf@uu.nl

Utrecht University
Department of Information and Computing Sciences
Buys Ballot Building, office 584
Princetonplein 5, De Uithof
3584 CC, Utrecht

dr. S.J. (Sietse) Overbeek
Second internal supervisor

S.J.Overbeek@uu.nl

Utrecht University
Department of Information and Computing Sciences
Buys Ballot Building, office 584
Princetonplein 5, De Uithof
3584 CC, Utrecht

I. (Ivar) Vennekens
First external supervisor

ivar.vennekens@vm-advies.nl

Vellekoop & Meesters
Drs. W. van Royenstraat 3
3871 AN, Hoevelaken

J. (Jos) Overbeek
Second external supervisor

jos.overbeek@vm-advies.nl

Vellekoop & Meesters
Drs. W. van Royenstraat 3
3871 AN, Hoevelaken

Table of Contents

List of Figures

List of Tables

Introduction

1.1. Background

Case handling is a popular paradigm emerging within the municipalities and governmental organizations within the Netherlands (Dekker, 2009). By using a case-based approach to workflow management, organizations can gain better and transparent insight in their business processes, allowing them to analyze the process from end-to-end instead of looking at it step-by-step.

Figure 1. Different Case-handling Systems in municipalities (Zaaksystemeninbeeld.nl, 2015)

The implementation of Case-handling Systems in municipalities started in 2009 in the municipality of Drechtsteden (Dekker, 2009), which showcased the benefits of using case handling. Within case handling, the focus lies on the *product* that is delivered when the case is successfully handled. A frequent example in the domain of municipalities and governmental organizations is applying for a permit. The case is initiated when a person starts the application process. The process can be initiated using a web-form, by phone or in person. If the case is accepted, the processes needed to procure the permit are executed by the representative and the case is closed when the permit is either approved or declined. The processes and activities that are executed in the case are linked to certain *data objects* that are the by-product of executing a specific step. These data objects can serve as a pre-condition for the next process step, meaning that the step cannot be executed before the data object becomes available. In the case of procuring a permit, the permit cannot be send out until all mandatory data objects are present and completed within this case (van der Aalst, Weske, & Grünbauer, 2005).

According to the map depicted in Figure 1, there are 19 different Case-handling Systems that are currently in use, spread across 79% of all municipalities (Appendix A). In an interview with Blom (2015), he argues the numbers of the actual Case-handling Systems: *"Halve of these systems have some aspects of a Case-handling System, creating links between documents, but most of them are just glorified Document Management Systems".* The actual number of systems that contain all aspects of case handling will therefore be lower than described in Appendix A. According to *Zaaksystemeninbeeld.nl (2015)*, 16% of municipalities are not using a system with case handling properties and for 5% it is unknown which type of system is used within the organization.

The positive reactions and benefits of Case-handling Systems have reached the ears of the *Netherlands Food and Consumer Product Safety Authority*, hereafter referred to as *Nederlandse Voedsel- en Warenautoriteit (NVWA)*. The NVWA is currently in the process of switching to a Case-handling System to replace their legacy systems that resulted from the merging of three smaller governmental organizations. The organization regulates 23 different domains, including animal health, crop protection, food and drinks, restaurants, subsidies, import and export. The regulation that is applied differs per domain, such as approving permits for restaurants, inspecting companies based on health criteria and providing subsidies. The domains are regulated by five divisions within the organization, which currently still use the legacy systems they used before the merge. As a result of this fragmentation, the organization and her employees have no transparent view of the processes and work of other divisions. The goal of the NVWA is to create a uniform and transparent system that can be used throughout the organization and streamlining their work processes.

1.2. Problem statement

The popularity of Case-handling Systems started as stated above roughly in 2009, however, looking at scientific literature there are no articles that describe the concepts and implementation methods of said systems after 2008. The articles that are available give a clear description of the case-handling paradigm and the elements that come with it, but fail to describe the actual definition of case handling and Case-handling Systems. Furthermore, there are no descriptions of best practices that organizations can use to avoid common pitfalls and issues during the implementation process. Due to the rising popularity of the paradigm, it is curious to see this gap in scientific literature. The descriptions of best practices could help organizations tackle problems early in the implementation process, comparing this to the existing literature and available documentation.

Adding to the problem of implementation is the size of organizations. The average workforce of a municipality in The Netherlands consists of roughly 200-300 employees that work with the system on a daily basis. The implementation of a Case-handling System might differ when it has to be rolled out in a larger organization, such as the NVWA. In comparison, the NVWA employs roughly 2500 employees that need access to the system in order to finish their work. Current literature does not present a method to tackle the implementation, adding to the scientific gap in this domain.

The problem statement of this thesis can be summarized as follows:

> *"Despite the rising popularity of Case-handling Systems within municipalities and governmental organizations, there is no direct overlap to be found between scientific literature and state of the art that is currently applied within organizations. The definition of the paradigm is unclear and there are no best practices available for the implementation of a Case-handling system, which can be a big disadvantage for organizations starting the design phase of their system."*

In order to bridge the gap between scientific literature and the organizational view of case handling, this thesis strives to create a better understanding of the case handling phenomenon by defining best practices as described by organizations that have implemented a Case-handling System, creating a uniform definition of case handling and supporting organizations in the design phase of their implementation. The support will be presented in the form of a modelling technique that organizations can use to create a uniform and transparent overview of their current and desired state of their processes.

1.3. Thesis outline

Chapter 1 described the context and domain of this thesis, resulting in the problem statement and research objective. Chapter 2 will describe the research approach taken in order to achieve the research objective and present the reader with the research questions and methods applied in the approach.

The theoretical background on concepts used throughout the thesis will be detailed in chapter 3, describing the specifics of case handling, modelling techniques and state of the art. In chapter 4, the results gathered from several interviews are described and summarized to extract best practices for organizations.

Chapter 5 introduces the modelling technique and the related concepts based on literature and data gathered from the interviews, resulting in the application of the technique to a case study in chapter 6, performed at the NVWA.

Finally, Chapter 7 contains the discussion points on the thesis and her approach, drawing conclusions and presenting opportunities for future research in Chapter 8.

2. Research Approach

This chapter describes the research approach of this thesis. In section 2.1, the research objective is described. Section 2.2 defines the research question and related sub-questions, and expands on the added value of these questions. The research process is then described further in section 2.3, explaining the performed literature and documentation study, interview process and case study design. Section 2.4 finally focusses on the validity constructs of the performed research.

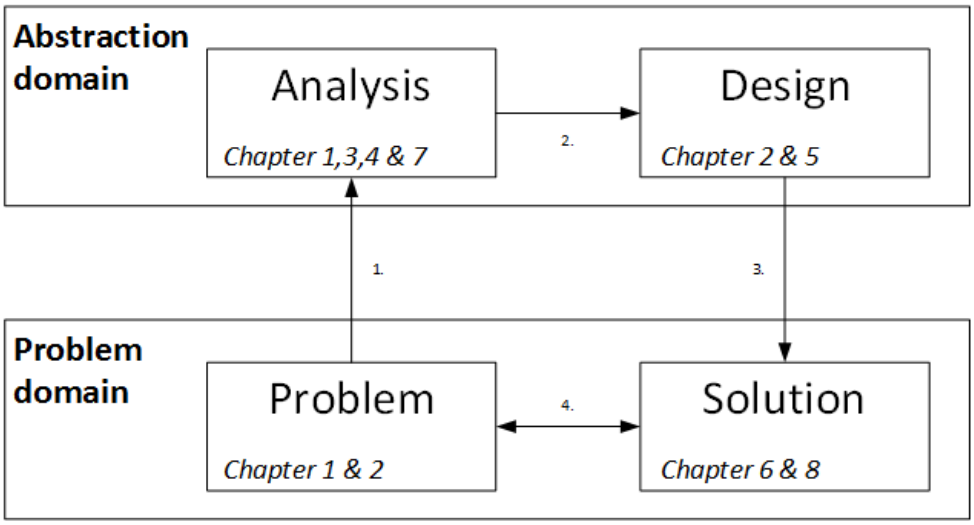


Figure 2. Design science cycle (based on Hevner, March, Park, & Ram, 2004)

The approach of this thesis is based on the design science cycle (Hevner et al., 2004; Polya, 1945). The cycle starts with the first step in Figure 2, the definition and understanding of the problem, as described in section 1.2. After the analysis, the second step is to make a plan for the design of the research to tackle the problem, which are described in this chapter of the thesis. These two steps can be categorized in the *abstraction domain* of the cycle, focused on making sense and coming up with a possible solution to tackling of the problem.

The *problem domain* contains the actual problem and the solution to the problem. After the design of the research, the solution is defined and tested on the actual problem. The fourth step is the evaluation and feedback of the solution. This allows the solution to be tested in practice and edited where needed to better suit the problem. The cycle does not necessarily have to be followed in the exact order as described above. During the implementation, changes to the design can be made in the situation that the insights gained during this stage have influenced the understanding of the problem, and therefore the analysis.

2.1. Research objective

The main objective of this thesis is to bridge the gap between available literature and the experience from organizations that have already implemented a Case-handling System in order to provide organizations that are planning to implement a similar with best practices and a uniform understanding of case handling. The scope of the research is limited to the design phase of the system, the phase in which companies inventory their current landscape and prepare for the restructuring of their processes in order to implement a Case-handling System. As described in section 1.2, the problem of this paradigm is that there is no uniform definition of case handling nor are their best practices for implementation based on experiences from actual implementation.

In order to tackle these problems, a literature and document research will be performed to analyze the scientific and organizational views on case handling. To define the basic best practices for implementation, interviews will be conducted at organizations, taking their view of case handling and comparing this with the views provided in literature. The interviews will also form the basis to define a modelling technique that can support organizations in the design phase of a Case-handling System.

2.2. Research questions

In order to support organizations in the design phase of implementing a Case-handling System, a main research question is defined. This research question is defined as follows:

RQ **How can the design phase of implementing a Case-handling System be supported using a modelling technique?**

In order to give a complete and comprehensive answer to this question, and to create a uniform understanding of case handling as a whole, the main question is answer via five sub-research questions:

SRQ1 *What is the definition of case handling?*

The first sub-question focusses on creating a uniform definition of case handling. As described above, there is currently no official definition of case handling to be found in scientific literature, and although there is an overlap between basic elements of case handling, organizations and researchers might disagree on the specifics of a definition. This research question strives to combine the views from scientific literature, organizational documentation and data gathered from interviewees to create a complete definition of case handling.

SRQ2 *How does case handling compare to conventional Workflow Management and Business Process Management?*

Case handling can be seen as a new form or view on Workflow Management and subsequently Business Process Management. This question will describe the differences and similarities of these concepts in comparison to case handling as defined by scientific literature.

SRQ3 *How can a Case-handling System be implemented in a comprehensive and unambiguous way?*

In order to gain an understanding of the implementation of a Case-handling System, information gathered from interviewees will be used to determine which way they have implemented their respective Case-handling Systems. Through this question we will obtain the current practices of organizations in order to compare them with the changes they would make if the implementation started now (SRQ4).

SRQ4 *What are the most occurring problems during the design phase of the implementation process and what would the interviewed organizations do differently if they would start the implementation now?*

From this question, information from the interviewees can be summarized into several best practices and pitfalls to avoid when implementing case handling in an organization. Interviewees will be asked to describe what they would execute the same and what they would change if they could start the implementation all over with the knowledge they have acquired through the previous process.

SRQ5 *How can the processes of a Case-handling System be modelled, while including all relevant data but avoiding unnecessary complexity of the model?*

To support the design phase with a modelling technique, a modelling convention needs to be defined and described. Using the knowledge of the interviewees and existing literature, such a modelling technique will be created and tested on a use case supplied by the NVWA.

2.3. Research process

This section describes the research process as shown in Figure 3. The steps performed are a literature and document review, followed by interviews and their analysis. After the interviews have been analyzed, a modelling technique is defined and tested using a case study.

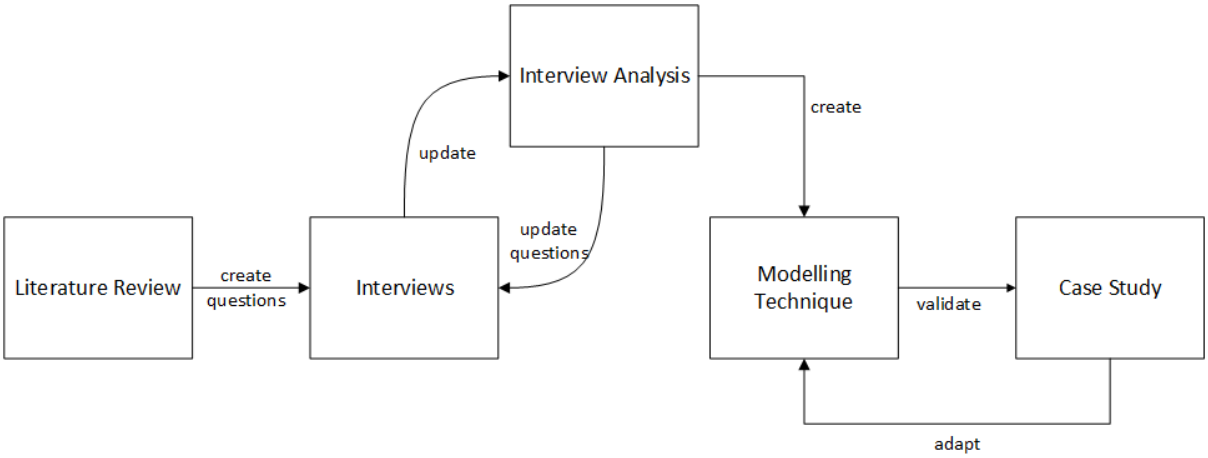


Figure 3. Research process visualized

2.3.1. Literature & document review

The second sub-research question can be answered by conducting a literature research into the concepts of case handling, business process management, workflow management and other related documents supplied by the NVWA. In order to answer the first sub-research question, context gathered from interviewees will have to be combined with the concepts of the literature in order to create a coherent definition. To assess the literature written on the topic of case handling, a narrative review will be performed (Paré, Trudel, Jaana, & Kitsiou, 2014). The narrative review selectively assesses the literature on a topic and is summarized as a narrative, described in chapter 3 of the thesis. References from the initial selection are included, using a snowballing technique, if deemed important enough in the context of case handling.

The initial impression of the availability of scientific literature on this topic was found to be scarce. A small group of researchers (van der Aalst, Grünbauer & Reichert, 2015) lay down the groundwork for this paradigm and the formal representation in the form of Petri nets. As mentioned before, the scientific research on the subject of case handling stops in 2008, while gaining more and more traction with organizations since 2009. Other scientific literature used includes the effectivity of workflow systems, organizational change in the public sector and comparisons between workflow management systems and case-handling systems. For the modelling technique, comparable techniques were used to create an easy to use technique for describing the processes, based on notations as Process Deliverable Diagrams and Business Process Model and Notation.

Documentation supplied by the case company was used to gain an understanding of the position of the company and the goals they have set. Other documentation the organization used to prepare

themselves for the public tender were taking into account while answering the research questions and performing the case study.

2.3.2. Interviews & analysis

Sub-research questions 3 and 4 will be answered by conducting interviews with organizations that have implemented a Case-handling System. The goal of these interviews is to gain an understanding of the current practices within the field of case handling and discovering best practices and pitfalls the organizations have had during their implementation.

Some examples the organizations can give insights on are:

- What method they used to write down their current practices and processes and what changes that needed to be made in order to support case handling;
- how long an organization of their size took in making the complete switch;
- how employees were informed of changes and how smoothly the transition went for them;
- how the ways of working were influenced;
- what benefits the organization gained from switching to a Case-handling System;
- what the negative effects of the switch included.

The interviews were conducted as semi-structured as to gather relevant information from each interviewed organization. The reason for using a semi-structured approach to these interviews is to let the interviewees speak freely on their experiences with this way of working. Hove and Anda (2005) describe encouraging interviewees to talk freely, asking relevant questions and following up on interesting topics as the main benefits of conducting a successful semi-structured interview. Problems that can occur during a semi-structured interview are silent interviewees, so that the only points that are discussed are the questions that were prepared beforehand, and interviewees that talk too much about irrelevant topics. These problems can be addressed during the interviews by continuing to ask relevant questions and to steer the interviewee back to the core of the interview. The questions used to guide the interviews can be found in Appendix B of this thesis.

The interviewed organizations include three municipalities, one province, the NVWA and the consultancy firm that is supporting the NVWA in the implementation process. The contacts for the municipalities and a province were acquired through this consultancy firm, which, according to Blom (2015) are a good sample of the current state of the art. Further information on the interviews can also be found in Appendix B.

To support the transcribing and summarizing of the interviews, the interviewees will be recorded during the sessions. The expected interview time will be between 30 and 60 minutes in order to gather enough information to answer the relevant sub-research questions. The data gathered from the interviews are summarized into best practices and pitfalls to avoid, and further information is used to define the modelling technique used in the case study.

2.3.3. Modelling technique & case study

The information gathered from the interviews will be used to find the baseline for the definition of the modelling technique. With the established baseline, comparable modelling techniques will be studied in order to determine the applicability of their concepts for this new modelling technique. The concepts of this technique will afterwards be applied in a case study, remodelling one of the processes at the NVWA (Yin, 2009).

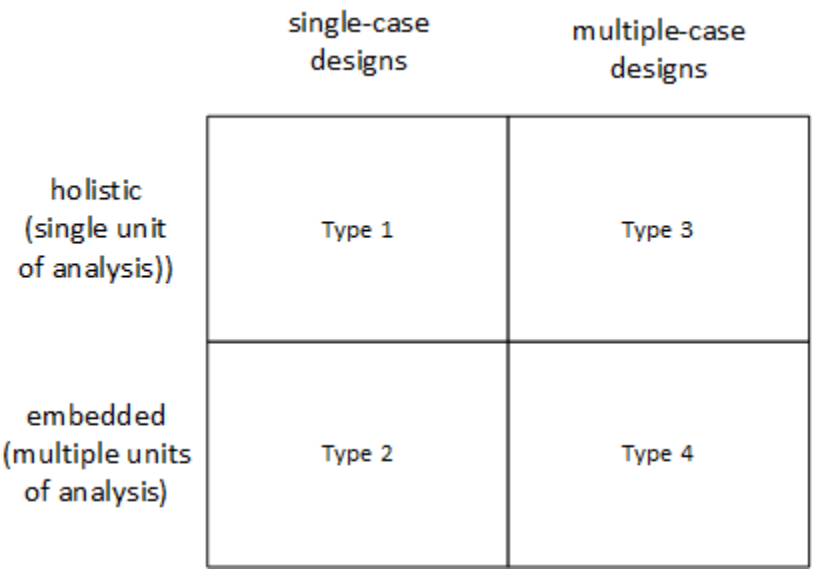


Figure 4. Types of case study design (Yin, 2009)

According to Yin (2009), there are four different types of case studies. The *holistic* studies focus on a single unit of analysis, either using a single case (type 1), or multiple cases (type 3). The *embedded* form of a case study can also be split into single (type 2), or multiple (type 4) cases and focusses on multiple units of analysis. For the case study performed in this thesis, the first type of case study design is chosen. According to Yin, *"A single case can be used to determine whether a theory's propositions are correct or whether some alternative set of explanations might be more relevant."*, meaning the applicability of the modelling technique can be tested using a single-case holistic design.

The main reason for selecting a single-case holistic design is the availability of case study material at the NVWA. Since the organization is in the early stages of redefining their processes, there is only one reworked process available. The case study would benefit from a multiple-case holistic design, since the applicability of the modelling technique could then be verified over more variations of process models, this is however not possible in the time frame of this thesis project and will be suggested for future research.

2.4. Research deliverables

The deliverables of the thesis will be the results of the research process described in section 2.3. The deliverable from the interviews with organizations will contain a summary of the approaches taken by the organizations, describing problems they encountered in their implementation process. From these problems, best practices will be defined based on how organizations would tackle the process if they can use the knowledge they have now to a new implementation process of a Case-handling System. The final and definitive definition of case handling will be created using the interviewees view on case handling and comparing this with available literature and documentation.

Another deliverable of the research will be the modelling technique and definition of the modelling syntax. The technique will be based on comparable modelling techniques, which are fitted to the baseline as determined from the interviewees. The technique will be verified and tested using a use case as described in the previous section. After the evaluation, feedback on the technique will be processed, making changes to the technique where necessary. The evaluation of case study can help the NVWA in revising future processes that have yet to be revised within the organization.

3. Theoretical Background

This chapter focusses on the literature review performed as described in section 2.3.1. The focus of the literature is on case handling and Petri nets, as this form of modelling is used later on in this thesis.

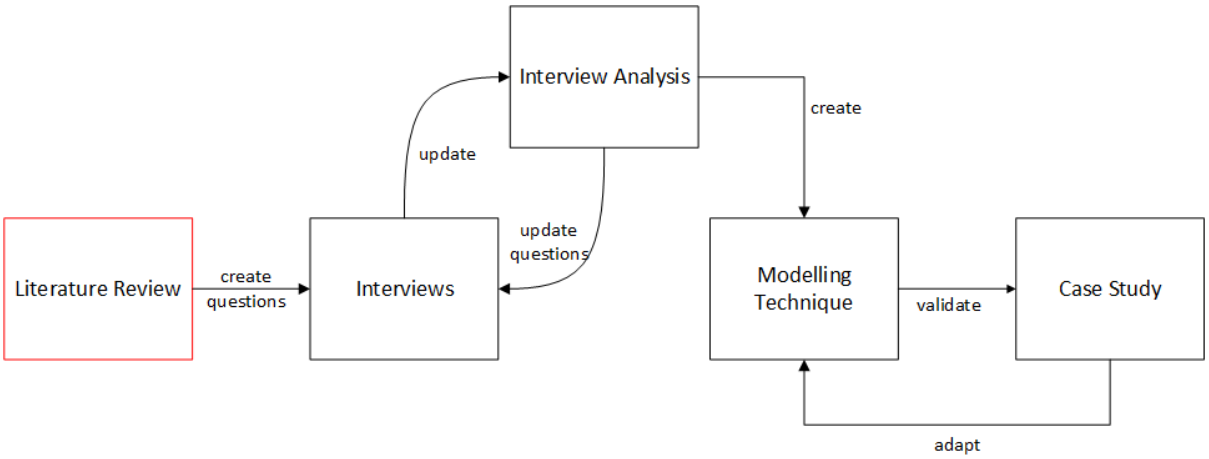


Figure 5. Research process – Literature review

3.1. Case handling

According to Aalst et al., (2005), case handling is a *data-driven* approach to Business Process Management (BPM), whereas with conventional workflow systems the focus lies on the processes that drive the workflow. Aalst, Hofstede & Weske (2003) define a Workflow Management System (WfMS) as:

> *"A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications."*

By using a WfMS, jobs can become more structured and factors like cycle time, throughput, communication and productivity increase significantly. In an effectiveness research by Reijers and Aalst (2005) on six Dutch organizations, the performance indicators *lead time (*time between start and finish of a case*), service time (*time spent by resources), *wait time (*idle time of resources) and *utilization of human resources* were all positively influenced with the use of a WfMS. However, because of the rigidly defined activities and lack of opportunities for creativity, a WfMS tends to be inflexible (Kueng, 2000).

According to Aalst and Berens (2001) the problems of a WfMS can be condensed into four core problems:

1. Work needs to be straight jacketed into activities, resulting in clustered activities. The actual work, however, is performed at a more atomic level;
2. Routing is used for both distribution and authorization;
3. The context of the case is pushed to the background because of the control flows in the WfMS;
4. The routing focuses on what should be done instead of what can be done.

A Case-handling System (CHS), a system using the concepts of case handling as a basis, can be placed in the category of Business Process Management Systems (BPMS), which are *"generic software*

systems that are driven by explicit process design to enact and manage operational business processes" (van der Aalst et al., 2003).

The main difference between the conventional WfMS and a Case-handling System is based on the third problem, the lack of *"context tunneling"* (Mutschler, Weber, & Reichert, 2008). Only the data needed to execute a certain activity is provided, whereas in a case-handling system all data related to the case is freely available to all actors, assuming they have the proper authorization. The activities that are performed result in *data objects.* These data objects are further distinguished by Aalst et al. (2005) in two categories, *free* and *mandatory* objects. Free objects can be changed during the runtime of the case, whereas mandatory objects are required before an actor is able to perform a new activity. *Forms* can be used to present different views on data objects within a certain case, which can be linked to activities to mark the most relevant data objects. By using data objects to lead the case, instead of waiting for the previous steps to be completed, it allows actors to work more freely and gives them the possibility to complete multiple data objects of a certain case. It changes the conventional 'in tray', a collection of work items for a certain actor are collected to a more open querying mechanism. A visualization of a case-handling structure is given in Figure 6.

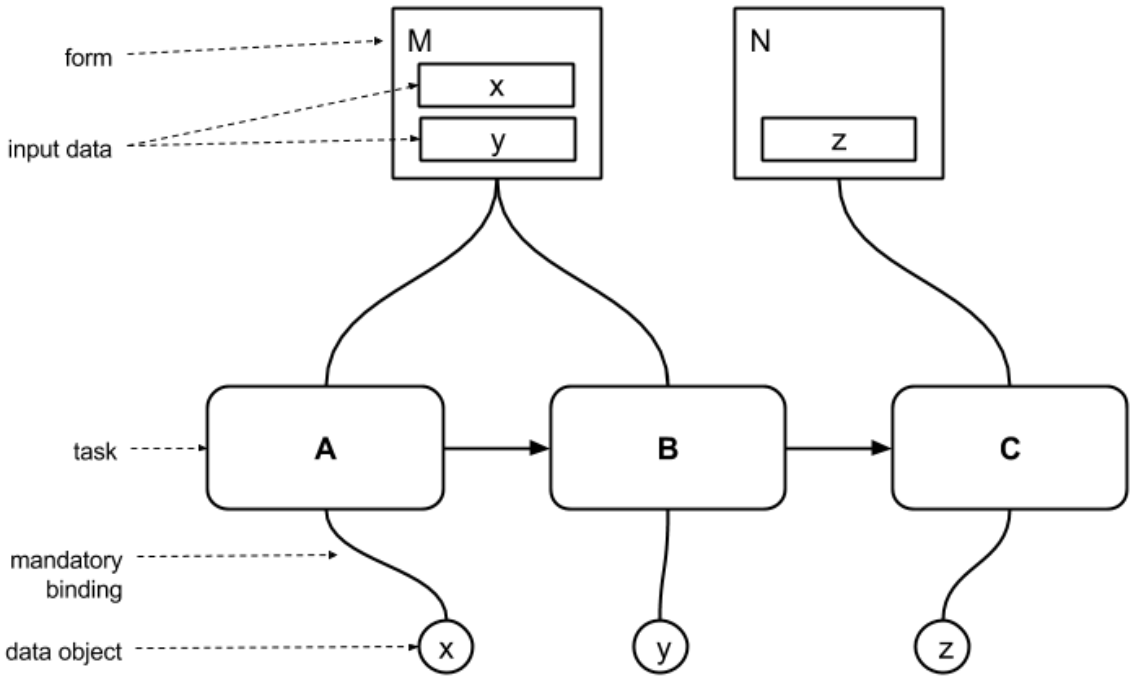


Figure 6. An example of a case-handling structure (Günther & Reichert, 2008)

The information displayed above can be seen as an answer to the second sub-research question, "*How does case handling compare to conventional Workflow Management and Business Process Management?"* To further distinguish the system, several comparisons have been made. Hee, Keiren & Post describe the three main components of a Case-handling System as (2008)

1. A workflow manager for executing the processes;
2. a document manager for handling the documents related to the case, and;
3. a Database Management System (DBMS) to manipulate the data.

Hee et al. (2008) created a proof of concept with the workflow engine YasperWE, based on the smart process editor Yasper (van Hee, Oanea, Post, Somers, & van der Werf, 2006), document manager

Infopath and a MySQL database to create a formal approach to case handling using Petri nets to model and verify the processes. In a number of experiments where students had to build a web shop prototype using this proof of concept, the students discovered several problems, which only came to light through the formal modelling and verification of the system.

Günther and Reichert (2008) have compared case handling with Adaptive Process Management (APM). Where case handling focusses mainly on using data as the main driver instead of processes, APM focuses on dynamic changing environments. With cases, there is always a possibility that changes in the environment occur which may influence the cases being processed. APM uses block-structured processes which can be executed in any order. When a change in the environment occurs, such as a law influencing the way the process is handled, only the block that process is related to has to be redone, instead of redoing all related processes as well. Günther and Reichert (2008) propose the integration of this APM approach into a CHS. By doing so, the case can be altered during the runtime instead of waiting for the case to finish.

When a case, defined as *"a coherent amount of work with a defined motivation and a defined result, in which quality and cycle time have to be guarded"* (Dekker, 2009), is initiated in a CHS, it is usually is the result of a certain action, like applying for a permit. This initial action can be described as the *case type.* The case type, applying for a permit, forms the core of the case, the case ends when the permit is either approved or declined. During the process, it is possible that certain activities require the start of a *subcase* that allow for the activity to be completed. This is the reason that the processes of a case are standardized across an organization, but can vary from case to case. For example, during the application for a permit, it can come to the attention of the employees that the applicant in question has already tried to get this permit approved before. This can result in the start of a *subcase* to investigate the reasons for dismissal in the previous situations, which can influence the result of the case. This subcase would not be initiated if this is the first time the applicant started this particular permit process. The continuous evolution and variability of processes is one of the major problem areas in information systems (Mutschler, Reichert, & Bumiller, 2008). Case handling tries to tackle this problem and that of hard-coded process logic by using a database that contains all case-type processes, in order to support their customization.

The adding of subcases to a case can result in complex processes that contain mistakes and delay the execution of the case. The case and relevant subcases most likely share certain activities, such as *decide* or *archive*, which, if complete processes are just added to the main case, have to be executed multiple times. The question in this situation is, how subcases can be added to the main case without obstructing the execution of the case as a whole in terms of time and complexity. The processes need stay formally correct during the runtime of the case and be able to adapt to the situation presented in the case.

In order to present an answer to the first sub-research question, *"What is the definition of case handling?"*, we will present a preliminary definition based only on the literature described above:

> *"Case handling is a data-driven approach to Business Process Management that provides all case-related data to involved actors, focusing on the case as a whole."*

This definition will be completed with the data gathered from interviews in chapter 4, resulting in a definition with both scientific and organizational aspects.

3.2. Modelling techniques

This section describes the modelling techniques that show relevance to the modelling of a Case-handling System and case types within an organization, which will support in finding an answer to the fifth sub-research question.

3.2.1. Petri nets

Aalst et al. (2003) give the following reasons for using Petri nets in the specification of workflows: *"Petri nets are formal, have associated analysis techniques, and are state-based rather than event-based."* To model Petri nets, the tool Yasper, on which the workflow engine YasperWE used by Hee et al. (2008) is based, can be used.

Petri nets can be used to model and analyze processes (van der Werf, 2011). Petri nets were invented by Carl Adam Petri in 1962 and have since then been expanded with dimensions as time, hierarchy and colors to allow for more complex processes to be modeled. According to Aalst (1998), Petri nets can be used for modelling workflow processes because the semantics of the elements have been formally defined. Petri nets are graphical, which allows for a small learning curve and communication with end-users. The formal definitions of Petri nets means that it is based on a mathematical foundation and therefore allows for several analysis techniques to be conducted. Some examples of the techniques are finding deadlocks, measuring response times, waiting times and occupation rates.

With the use of Petri nets, the processes in the case can be *validated*, *verified* and *analyzed* for performance. The validation tests if the process behaves as expected, the verification tests the formal correctness and the performance analysis checks if the runtime of the case meets the requirements that have been set for that particular case type.

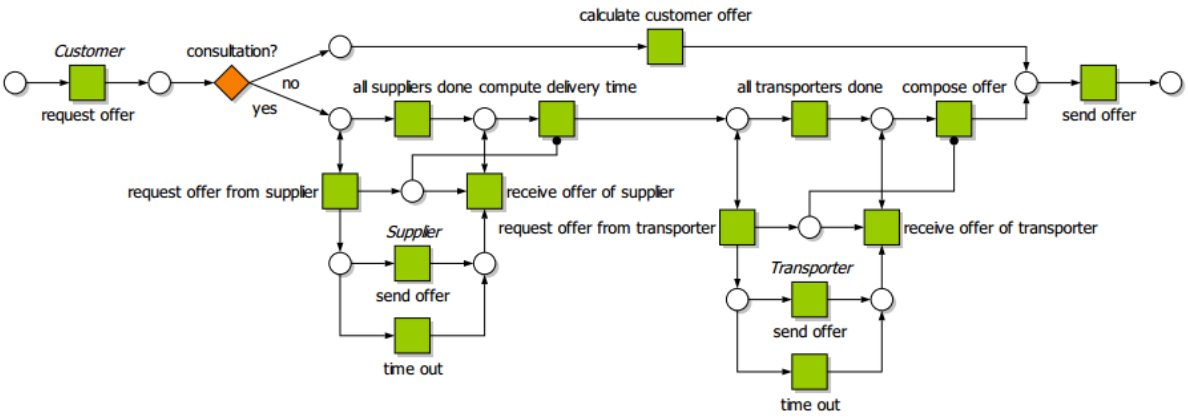


Figure 7. Example of an offer request Petri net (van Hee et al., 2008)

As depicted in Figure 7, a Petri net are represented by circle and rectangle nodes, where the circles are the so called *places* and the rectangles are the *transitions.* These nodes are connected via directional *arcs,* and connections between two nodes of the same are not allowed. The Petri net starts on the input place, and follows the directional arcs to the output place. The process of moving from one node to another is done by the *firing* of so called *tokens*, depicted by black dots in the nodes. If a transition node *fires* a token, it consumes one token from the input place and produces one in the output place. In Figure 4, the transition *request offer* can consume a token from the input place on the left and then produce one in the place on the right, following the direction of the arcs. The following diamond-shaped transition represents a choice in the process, as is a regular expression in many modelling languages such as UML.

Petri nets can be extended further with the dimensions color, time and hierarchy. A colored Petri net contains tokens that have an assigned value to it, the '*color*' or *type* of the token (Günther & Aalst, 2005). This allows for transitions to have specified preconditions that vary per type of token, making it possible to get different results from a single transition. An example in the context of case handling is that the tokens are given a case ID as the identifier, which allows for multiple cases to be executed in parallel. The time dimension is a way to model delays or exceptions such as a *time out.* In Figure 7, a timeout is modeled as a transition that allows the circumventing of the transition *send offer*. If a process model is too large to get a comprehensive overview, a *subnet* can be used to introduce hierarchy in the Petri net. The *subnet* consists of a subset of the places, transitions and arcs of a Petri net and ensures the readability of the modeled process. If a more detailed representation is required, the net can be zoomed into the subnet to show its inner nodes.

3.2.2. Business Process Modelling Notation

A frequently used technique for modelling processes is the *Business Process Modelling Notation* (BPMN). The primary goal of this standard was to provide an easy to understand notation for all involved parties, from business users to technical developers. White (2004) divides the basics of BPMN in a *Business Process Diagram (BPD)* in four categories: *flow objects*, *connecting objects*, *swimlanes* and *artifacts*. The objects describe the activities performed and determine the flow from activity to activity. An example of a basic process modelled in BPMN is shown in Figure 8.

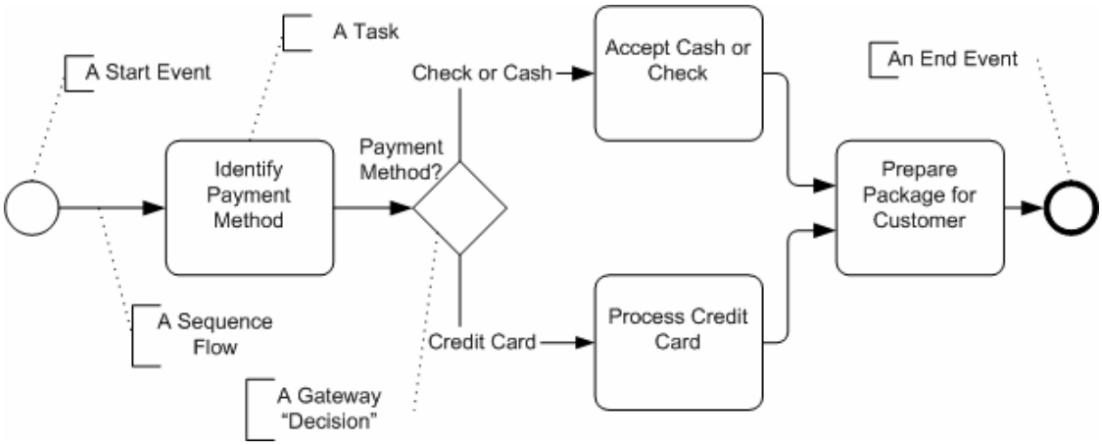


Figure 8. A basic BPMN process

The start and end of the process are depicted by *events*, which will either have a cause or an impact depending on the location of the event. The activities are depicted by the rounded rectangles, and describe the tasks that are executed in the activity. Sub-processes can be modelled by adding a plus sign in the bottom center of the rectangle, which allows for high level modelling in one diagram, and specific modelling in a separate diagram. This allows different users to gain understanding of the process as suit their needs, instead of always being presented with a detailed process diagram. The diamond shaped gateway is the choice node that splits the process based on the properties determined in earlier activities.

To identify different actors within one BPD, the notation makes uses of swimlanes and pools to differentiate between actors. Every pool contains the activities of one participant, where swimlanes can be used to categorize activities to create an easier to read process diagram.

3.2.3. Declarative Modelling

Another relevant technique in the context of case handling and remodelling of processes is *declarative modelling* (Pesic & van der Aalst, 2006; Reijers, Slaats, & Stahl, 2013). Pesic and van der Aalst (2006) notes that declarative modelling allows users to specify the *'what'* without describing the *'how'* of the process. Imperative languages, such as Petri nets, focus on describing the processes to the very core of the process, since the correct execution of the process relies on the execution of the modelled process. According to Pesic and van der Aalst (2006), these imperative languages tend over-specify, instead of focusing on creating a generic approach.

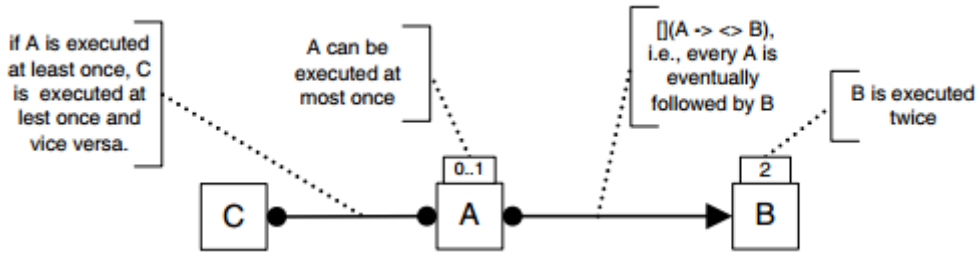


Figure 9. Declarative language ConDec (Pesic & van der Aalst, 2006)

In comparison with BPMN, the declarative language ConDec (Figure 9) describes the execution constrains of a certain task, depicted by the number in the rectangle on top of the activity. The flows are determined by the flow arcs. A filled circle at the end of an arc means the activity on that end has to be executed at least once, before executing the connected activity.

Reijers et al. (2013) identified the main use case for declarative modelling in the area of *process evolution*. Managing processes over time and making changes to processes is easier to be done on a higher level of abstraction, compared to the imperative and specifics of a modelling technique such as Petri nets and BPMN. Declarative modelling focusses on constraints, and in an ever changing environment removing or adding constraints is considered easier than removing or adding an activity to a process model. The second use case for declarative modelling is the basic visualization of processes in a process model, however only in the context of high level modelling. Reijers et al. found that declarative modelling was not found to be useful for specifying low level process specifics.

3.2.4. Meta-modelling technique

Another high level modelling technique that has some overlap with the implementation of a Case-handling System is the meta-modelling technique as described by van de Weerd & Brinkkemper (2008). The technique focusses on the creation of a *Process-Deliverable Diagram* (PDD), describing the method for designing and implementing an information system. The development of information systems is supported by reusable *method fragments*, gathered from existing established methods. This allows companies to specify an implementation method for their organization without building this method from scratch. The PDD consists of *activities* and *concepts*, where concepts are the result of the performed activities, as shown in Figure 10.

The reusability of the meta-modelling technique shows promise for the modelling of case types, as the processes in an organization can show overlap over different processes. If an organization can reuse fragments of other case types, it can benefit them on implementation and design time. The meta-modelling technique however, is just as declarative modelling, based on a higher abstraction level than the imperative languages described above. The technique is also focused on modelling methods instead of processes, meaning the concepts of the technique are interesting in the scope of this research, but the modelling syntax is not.

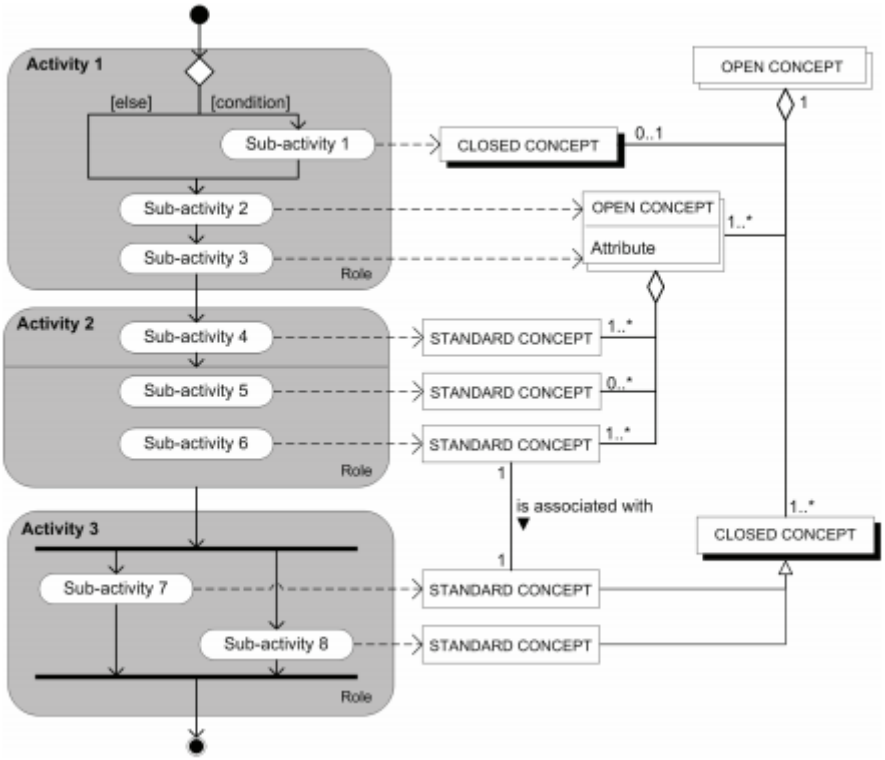


Figure 10. Meta-modelling - Process-Deliverable Diagram (van de Weerd & Brinkkemper, 2008)

4. Current State of Case-handling Systems

This chapter reflects on the results of the interviews and the analysis of gathered data from the interviewees. Further details of the interviews can be found in Appendix B.

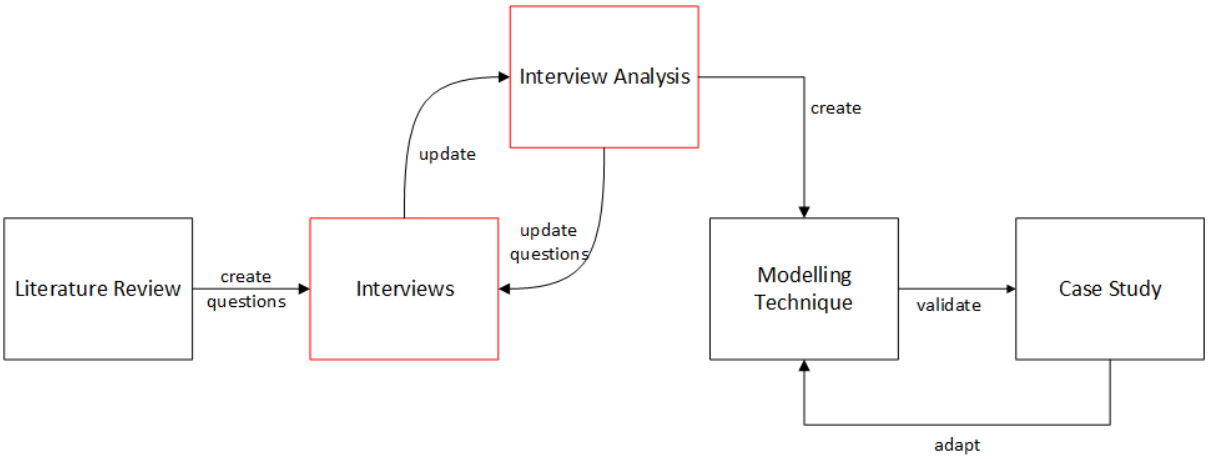


Figure 11. Research process – Interviews and analysis

To gain a wider understanding of Case-handling Systems and how they are being applied at different organizations, a series of interviews have been conducted. The interviewees were all familiar with Case handling and Case-handling Systems, either as a consultant or project leader. Further details of the interviewees have been summarized in Appendix B.

As described in section 2.3.2, the interviews were semi-structured to give the interviewees the possibility to present their view on Case handling and on how their organization had implemented a Case-handling System. The four organizations vary in size, implementation time and implementation method. Table 1 shows that three of the organizations chose for an implementation per team or cluster of teams based on how the organization was structured. Of these three, two organizations are still in the transformation process, either still converting teams to the new situation or fixing old cases to match the current systems.

This chapter further focusses on the implementation approaches in section 4.1, how they generated traction in the organization in section 4.2, how criteria and rules are described in section 4.3, summarizing the data with best practices in section 4.4.

Organization	System	Tender	In use	# of Employees	Implementation approach	Finished?
1. Municipality of Bunnik	Mozard Suite	2010	2013	125	Per cluster	No
2. Municipality of 's Hertogenbosch	Verseon E-Suite	2009	2010	1200	Per team/cluster	No
3. Municipality of Molenwaard	Mozard Suite	2011	2013	200	Per team/cluster	No
4. Province of Noord-Brabant	Corsa Nxt	2014	2015	1000	Big Bang	Yes

Table 1. Overview of systems at interviewed organizations

4.1. Implementation approaches

The fourth organization, the province of Noord-Brabant, differed from the other three organizations in terms of the implementation approach. The organization worked in close contact with the supplier of their old Document Management System (DMS), Corsa, to transform their product into a Case-handling System. The organization was under pressure to transform their system before the current administration period would end, which resulted in an implementation that effectively only took one year. The old system was in use since 2010, but could not adhere to the case handling principles. Instead of starting a public tender, they decided to upgrade their current DMS to a Case-handling System.

The approach that the organization took was a Big Bang, plugging out the old system and turning on the new version of the system, Corsa Nxt. A Big Bang approach differs from a clustered approach in the terms of time of the switch. Instead of upgrading parts of the system, the whole organization is transferred in one go. According to Harry Post, the Case handling project manager, they first tried the cluster approach as a pilot. The pilot took about four months and during that time, they had transformed and modeled the case types of two of the sixty teams in the organization. These results signified the importance to use a radical different approach and to keep the implementation as generalized as possible. *"By abandoning the specific modelling and focus on generalizing, we created freedom for ourselves as a project team, and the organization as a whole."* - Post (2015).

The project team at the Province of Noord-Brabant started by analyzing the standard case types that other provinces were using in order to generalize their processes on a high abstraction level. Provinces have about 200 different case types, whereas municipalities have about 400. The difference in the amount can be amounted to the difference in operation scale. A municipality serves all their citizens directly, whereas the overarching province focusses more on other organizations. The focus of the organization was to strip these 200 case types down to their most general form. Using workshops with people from all relevant clusters, the organization succeeded in creating 30 *case forms*, an umbrella case type from which all case types are derived. *"By creating these general case forms, we sped up the implementation tremendously. All case types that belong to a certain form, could be derived from that form by adding the specifics for that type to the skeleton that the form provided."* - Post (2015).

Harry Post notes that this approach might not be applicable to every organization. If the case types have too much variations, it might result in a slight reduction in case forms, but not enough to reduce it to a small amount of generic case types. However, the case form approach is very effective at this organization. After defining the 30 case forms, the 180 case types that were derived from them, were modeled in four days. The organization is currently monitoring the effectivity of all case types and making changes where needed. "*Even when it is possible to model every single exception in a case type, the result is always unneeded complexity and high maintainability. By keeping it generalized, we do not have to change our case types when there is a miniscule change in procedure."* - Post (2015).

The total project took about 1 year and 4 months, demonstrating the effect of a generalizing processes. The focus on case forms resulted in the removal of redundancy and spared the organization the time of modelling every single exception in the case types.

A similar approach was taken by the municipality of Bunnik, were a *mother case type* was used for modelling all case types. Instead of a Big Bang approach, this organization chose a phased approach, converting the organization per cluster. Bunnik spent a long time in the tender phase of the implementation, visiting other organizations and looking at partnerships with other municipalities. After three years the choice was made to make the switch from their legacy DMS to an *all-in-one* Case-handling System, the Mozard Suite.

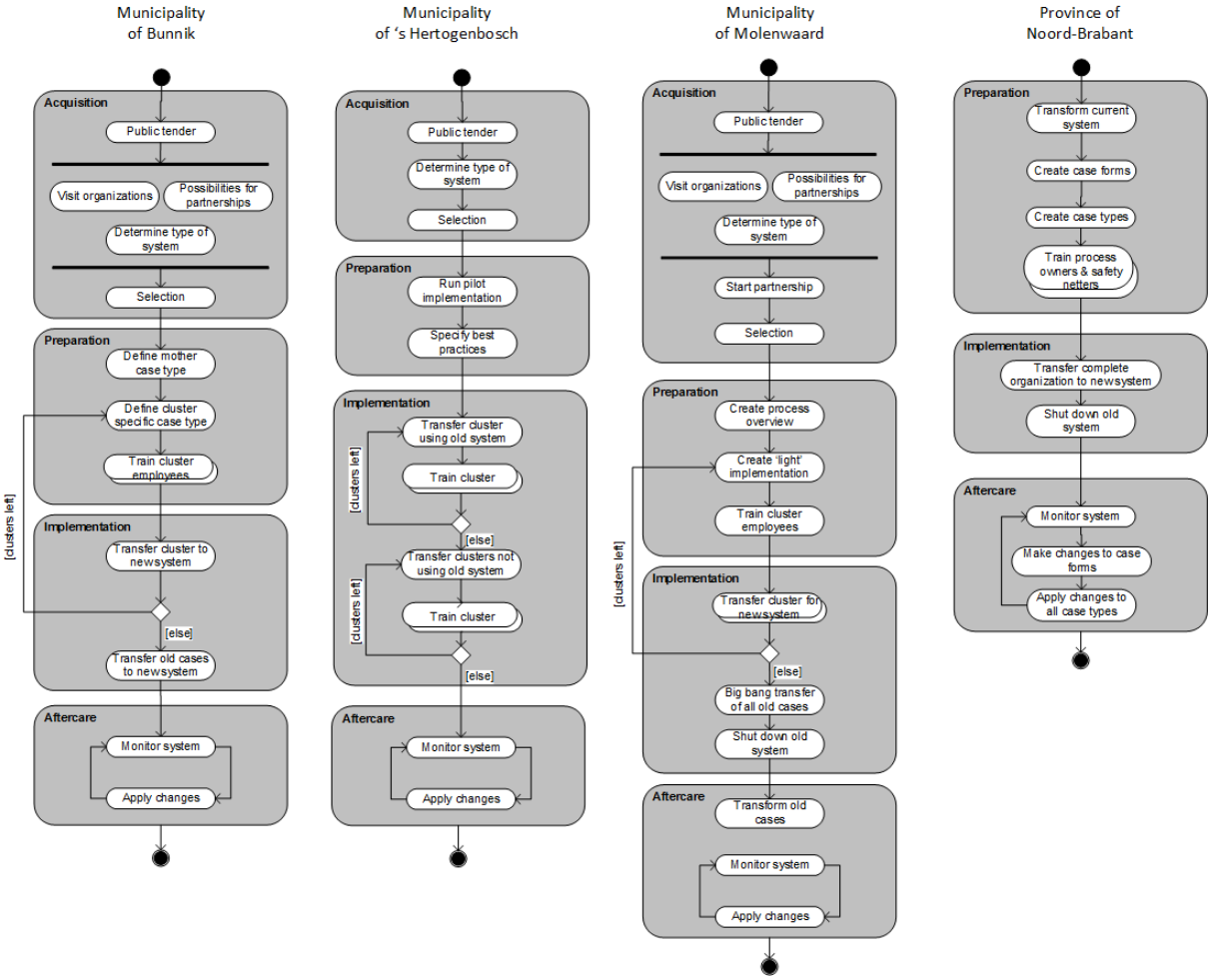


Figure 12. Implementation methods visualized in a Process-Deliverable Diagram

The organization started with a 'light' implementation of the case types, where the project team created the case types including the relevant templates and documentation for a single cluster. The use of a light implementation of the system allowed the organization to move to the new system faster, all clusters were transferred to the new system in roughly 11 months. Clusters were given a training on the new system on the day of the implementation, where all new cases were inputted in the new system. Old cases had to be handled in the old system, which resulted in the use of two systems for a significant amount of time. Some cases have a lead time of several months, resulting in inconvenience for the employees, who had to switch back and forth between systems to handle their cases. *"The negative side of this approach is the change is not instantaneous, employees are sometimes stuck with two different systems for a few months. This results in of friction and a bad experience until 95% of the cases are being handled in the new system."* – de Keijzer (2015).

After all clusters were transferred to the new system, the second phase in the implementation was started. This professionalization phase uses the same approach as the first phase, focusing on one cluster at the time. *"We see the implementation of a Case-handling System as a 'growth model'. We start switch at a certain point and move through the whole organization. Afterwards we look at the results of the 'light' implementation and make changes where they are needed, taking the wishes and needs of the employees into account."* – de Keijzer (2015). Some of the results of this phase are the

generalization of certain case types and removal of case types that are not used more than 10 times in a year. The organization hopes to finish this phase at the end of 2015.

The municipality of Molenwaard also took the phased approach in their implementation of the all-in-one Case-handling System *Mozard Suite*. The clusters in the organization were switched to the new system in about 1 year and 10 months. The project team was focused on generating quick wins for the civilians and employees, resulting in a services-based approach instead of a technical-/system-based approach. The Case-handling System was implemented as a light version, making the generalized switch to the new system as clear as possible. The project team focused on each individual team and their processes, modelling the general basis of the processes in the current situation. Afterwards the processes were transformed into case types, adding the needed specifics per type. Some cases were still being handled in the legacy DMS, while others were added to the new system.

One of the problems Molenwaard encountered because of this approach, was the remains of the 'light' cases in the legacy system. As a solution for this, a Big Bang approach was used on the remaining cases. *"The old cases were lagging behind in the old system, resulting in double system costs and maintenance. As a solution for this, including the wish for cutbacks from management, we decided to move all old cases to the new system in a Big Bang approach. I wish we hadn't, we've been stuck for 2 years trying to clean up and transform these cases." – Gerats (2015).*

The organization is still working on the aftercare of their Big Bang approach. According to Gerats (2015), the 'light' cases were put on hold for 1 year and 5 months, resulting in a halt in the implementation during that time. The team is currently working on transforming the remaining cases to the new Case handling format. Gerats (2015) would focus only on the phased approach if the organization would start the implementation with their current knowledge: *"The Big Bang resulted in reduced costs for the maintenance of two systems, but also resulted in the project being stretched out for over two years. If we had stuck to our phased approach, I think we could have worked more methodically and have been done with transforming all cases at least 18 months earlier".*

Organization	# of cases/year	# case types	Implementation time
1. Municipality of Bunnik	16.000	400 - 500	Jun 2013 – Dec 2015
2. Municipality of 's Hertogenbosch	50.000	400 - 500	Feb 2010 – Sep 2015
3. Municipality of Molenwaard	25.000	400	Jan 2011 – Dec 2015
4. Province of Noord-Brabant	20.000 (estimate)	180	Dec 2013 – Apr 2015

Table 2. Number of cases and case types compared to implementation time.

The municipality of 's Hertogenbosch has been using a Case-handling System the longest of the four interviewed organizations. The focus of the implementation at this municipality was not on improving services, but rather on improving the technology and systems. The legacy systems were mostly based on a custom-tailored approach. The trigger for moving to a Case-handling System was because of the lack of knowledge of the legacy systems and cluttered IT landscape. The Case-handling System however, was chosen on the amount of customizability and ability to create links between systems. *"The goal of the new system was to avoid a custom-made system, but our choice resulted in a system that had the possibilities for customization. We are now once more stuck with a lack of knowledge of the current system."* – Klerks (2015).

The organization took a phased approach, starting with the fitting of the system on the current infrastructure. This resulted in custom links to be made to other required systems, such as the personal records database. According to Klerks (2015), the organization spend too much time trying to get the system to work and linking it to other systems. The focus was on the technology, instead of gaining an understanding of the processes in the organization and creating transparency. After a few pilots, the project team had acquired some best practices for their implementation and teams to showcase the case handling concept to the rest of the organization.

The old system used by 's Hertogenbosch was a Workflow Management System with a separate Document Management System. About fifty percent of the organization was working with this system, while the other teams only used the DMS. The teams using the WfMS were transferred to the new CHS first, since these teams had more experience with the concepts. After all these teams were using the new system, the project team started with those who had never used a WfMS before. The organization is currently rolling out the system to the last few teams this year.

There is a high degree of decentralization in this organization, every team decides their own course and have their own view on how certain processes should be run. The culture in the organization was not open to change, resulting in a long implementation time of almost five years. The decentralized character of the organization resulted in almost no generalization or standardization of case types. Klerks (2015): "*We have four states: Intake, Pending, Handled & Archived. The pending state is when the case is being handled by a team, where we trust our employees to do what they were hired to do. The other states are standardized throughout the organization."*. There are no case types defined in the system, resulting in a hard to use system where every team uses fields for different reasons.

The reason Klerks (2015) gives for this choice is that it would have cost them many more years if all processes would have been modeled into case types and capturing all intangible information. Their approach focusses on the system as a tool for the employee to use, not as a supporting mechanism. The organization is currently entering a new public tender phase, in which they want to pick a system that does not allow too much customization and focus their approach on the process side of the system, instead of the technical side.

4.2. Generating traction in organization

All interviewees stress the point of getting the organization and employees to support the switch to a Case-handling System. The aftercare of the implementation is one of the most time consuming processes, especially if the managerial support is low.

The organizations have chosen different approaches to introducing their employees to the new system. The municipality of 's Hertogenbosch decided to pick a few teams to run a pilot and use these systems as a showcase for the other teams. Bunnik gave their teams thorough training of the system in small groups, allowing all employees to ask their questions beforehand. The most important factor that the interviewees agreed on was to show the employees the added value of the new system in order to make the transformation process a bit easier for them. De Keijzer (2015) described the change in the organization as a nine month process, with a tipping point after three months*. "In these first three months, the employees only see the new system as more work for them, and transparency in the system is something to get used to. After three months, they see the benefits of this social control and the system as a whole."*.

According to Gerarts (2015), the best way to get the whole organization to support the switch is to start at the top with the managers. After they have been won over, the organization will slowly start to move towards the change. The next step is to persuade the process owners in every team and to

provide them with a timeline of the transformation process. After all the case types have been defined in collaboration with the process owners, the remaining employees should be given an extensive tour of the system and training, after which the switch should be made as soon as possible.

Blom (2015) notes that ninety percent are not willing to change: "*Learning to work with a new system is extra work and the employees are already overloaded with work. Change is hard.*". When this is the case in a larger organization, it can become harder to generate traction for the change, like what happened at the municipality of 's Hertogenbosch. Blom (2015) stresses the importance of involving as many people with the project as possible, preferably in different parts of the organization.

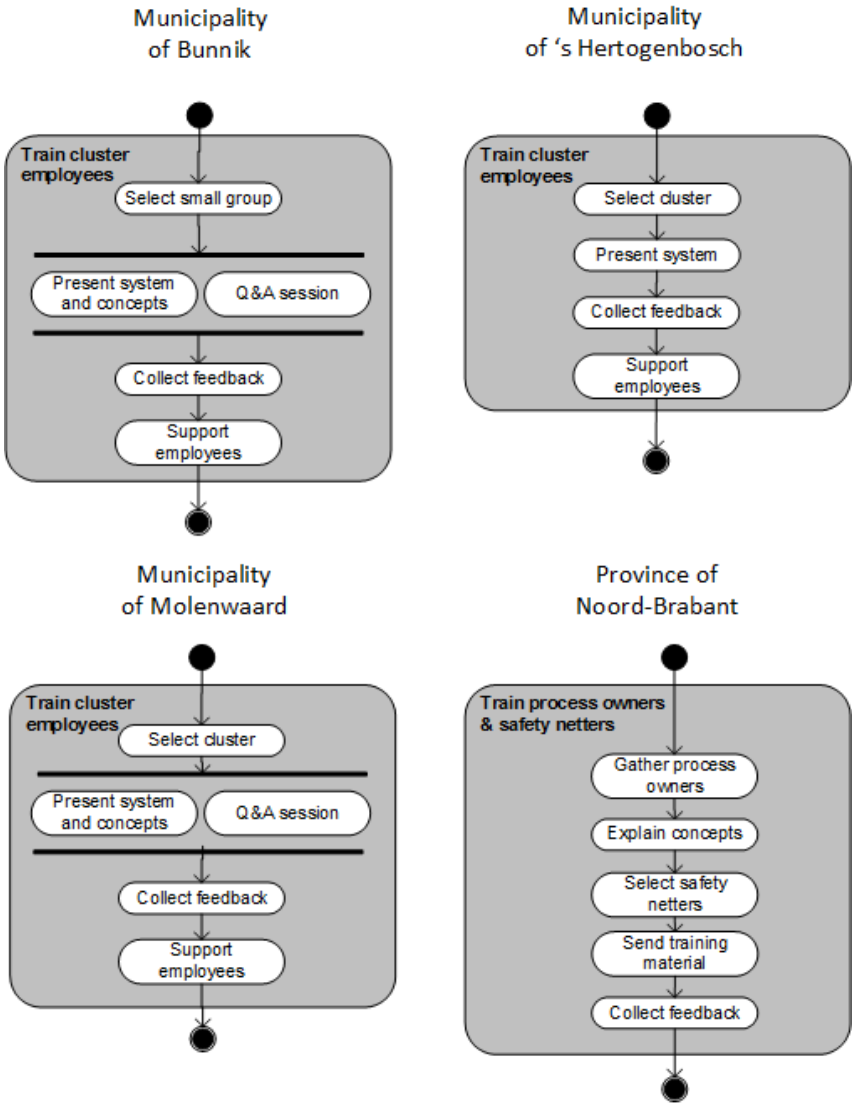


Figure 13. Training methods used at organizations

At the province of Noord-Brabant, there were no extensive trainings for employees, since the organization was not implementing a complete new system. The project team has knowingly chosen to keep their involvement with the internal business of the teams as low as possible. The approach they chose was to use the cluster representatives and so called 'safety netters' as the points of communication, instead of accompanying the employees themselves. This resulted in several central points for feedback, and allowed employees to press their concerns and questions to a familiar person.

Getting the organization to support the switch to a new system and supporting the employees after implementation is one of the biggest pitfalls the interviewees agreed upon. Klerks (2015) summarization of the matter captures the general impression given by the interviewees: *"Willing people can make failing systems work, unwilling people can make working systems fail".*

4.3. Modelling case types and relevant criteria

In order for organizations to switch to a new Case-handling System, mapping their IT landscape and processes is an important step. The interviewees gave the creation of clear case types as an important step to implement the system. Two of the four organizations used an umbrella case type as the basis for modelling all their case types. The other two either did not focus on modelling the individual case types of the teams or approached the modelling of the case types as an individual approach per team, without using an umbrella case type as the basis.

A focus point that was given was to 'keep it simple, keep it generalized'. By modelling the case types as basic process flows, the specifics of the processes can be removed, resulting in a clear, uncluttered case type. To achieve this throughout the whole design phase of the implementation process, it is key to create transparency in the organization on how the work is done. The best example from the interviewees was from the province of Noord-Brabant, with the creation of the case forms. After the unsuccessful pilot of modelling per cluster, the process owners of comparable processes were gathered in a workshop to create a case form which covered all basic and general aspects of their processes. By involving all relevant owners in this process, it is possible to avoid discussions afterwards.

By stripping the processes down to their bare functionalities, redundant process steps can be removed, improving readability of the case types and improving the possibility of re-using elements of the process steps in other case types.

Another important factor is the way the underlying rules of the system and process steps are documented in the system. The interviewees had several different approaches which were used in their system. In general, every case type is linked to certain documentation and documents that need to be filled in to complete the process and generate the product assigned to the specific case type. At the municipality of Bunnik, the checklists and criteria are placed in one of these documents. With the creation of a new case, all relevant documents are placed in the case dossier, accessible for all employees. Going through the process steps, employees can consult the document for the needed criteria. *"Most of our employees will open the file once in a month, to see if there are any changes, but since one handler focusses on one type of case 90% of the time, he does not need to check it, because they know what the criteria are by heart."* – de Keijzer (2015). The system uses several states that can be checked if a certain process step is open, being worked on or closed, allowing colleagues to jump in if needed.

The municipality of Molenwaard, where the same system as Bunnik is used, uses additional checklists in the system itself to support their employees in the steps. The checklists can be seen as certain business rules that need to be done before the handler can set the state of the process to closed. There are some formal checks in place to check the case dossier if certain documents that are required for the step are created, but it is still possible to create an empty document with the correct naming convention and close the step. *"It is possible, but we trust our employees to do their job correctly. If they decide to create empty documents, they are held responsible for their actions."* – Gerats (2015). Noord-Brabant uses the same principles as Molenwaard, but have not specified too many criteria yet. The organization is currently monitoring the effectiveness of their generalized approach and will specify further when needed. *"Since we have just gone live with the system, it is hard to say whether*

a generalized approach is effective enough. We will monitor the situation and will make changes where needed, but as little as possible." – Post (2015).

Looking at the municipality of 's Hertogenbosch, there is almost no transparency in the organization in respect to the process modelling. Teams using the same ordering process might have different approaches and use different criteria within the system when running this process. The decentralized focus of the organization severely hinders the transparency factor of case handling. Case type documentation and related criteria are managed by the teams themselves, nothing of this is registered within the system.

On the topic of changes of criteria, all four organizations leave the tracking of these changes to the specific teams and clusters. If changes need to be made, the initiative lies with them to either change the documentation themselves or notify the project team to implement the criteria change. The ability to change case types and the related documentation lies mainly with the project team, in order to ensure the correctness of the types.

4.4. Summarizing gathered interview information

The first sub-research question stated in this thesis was "*What is the definition of case handling?"* In section 3.1 we presented the preliminary definition as:

> *"Case handling is a data-driven approach to Business Process Management that provides all case-related data to involved actors, focusing on the case as a whole."*

After the interviews, it becomes apparent that the main reason of implementing case handling is to improve their services to their customers. The products that are delivered as a result of a case are the deliverable real incentive to finish the case. As described by van der Aalst et al. (2005), data is an important part in case handling, but is used as support for the employees to finish the case. To add to this, we compare the preliminary definition with the one as defined by the NVWA (van Klink & Hartelman, 2013):

> *"Case handling is the logical structuring and controlling of cohesive activities and tasks within an organization to improve the quality of service and the resulting product. The product and services are the focus point of this approach. By standardizing processes, information services and products, the execution and quality becomes more predictable."*

As per this definition, it becomes apparent that the focus of organizations is on the improvement of the final product and the standardization of the processes across the organization. In the context of this thesis, we define case handling as follows:

> *"Case handling is a standardized, transparent and product-driven approach to Business Process Management that provides all case-related data to involved actors, focusing on the case as a whole by removing the separation between processes and data."*

This definition summarizes the aspects found in literature and as described by the interviewees. The aspects of a standardization, transparency and product-driven nature are added. We are positive that this definition covers all aspects of case handling in one clear definition.

4.4.1. Best practices

Sub-research questions 3 and 4 focus on the best practices that could be extracted from the interviews, *"How can a Case-handling System be implemented in a comprehensive and unambiguous way?"* and "*What are the most occurring problems during the design phase of the implementation process and what would the interviewed organizations do differently if they would start the implementation now?".* After the analysis of the interviews, these questions can be answered and described as best practices for the implementation of a Case-handling System.

One of the main aspects the interviewees focused on was on creating a generic overview of the organizational processes, in order to create a coherent overview that all involved parties can understand. By doing this, the overview can be used in sessions to explain the necessity of a new system and allow involved process owners to share their view on the processes. The approach taken by two of the four organizations was to create a generic case type that encompasses the process steps of several comparable processes. In this approach we see some overlap between the concepts of declarative modelling and the meta-modelling technique, which focus on a higher abstraction level instead of modelling all specific process steps. By using this approach, the interviewees noted that they could easily communicate their current state and wishes to management and employees alike, allowing them to add to or change parts of the model which they did not agree upon.

In order to implement a Case-handling System in a comprehensive and unambiguous way, we define the first best practice: *Create a generic overview of the current state of processes in the organization to support the generation of traction amongst management and employees and to create a coherent view of the processes on which all process owners agree.*

This best practice is also the solution to the problem area of understanding the processes within the organization. As noted by the interviewees (de Keijzer, 2015; Post, 2015), the implementation sped up significantly after a generic model was defined. It allowed the organizations to specify the specific case types quicker based on the predetermined generic model. Process owners have in-depth knowledge of the processes and tend to jump to the exceptions within a process very quickly. If process owners do not agree on steps, it will result in many different forms of the same process fragment, even if the basics of the steps overlap. To avoid the problem of over-specification and redundant process fragments, organizations can use the generic overview to align the views of the process owners and other involved actors.

Another important aspect is the training and preparation of employees for the use of the new system. Although case handling shares some aspects with a conventional WfMS, it still adds many new concepts to the way an employee does their job. Transparency is a key concept of case handling, and results in the visibility of all data to all involved actors. This means that employees can view the status of cases from their colleagues, adding the aspect of social control and managerial analyses to the system. As shown in Figure 13, the organizations have used different approaches to their training sessions. When using a phased approach, the organizations focused on training employees per cluster and switching to the new system after the training was completed. These sessions were led by the project teams and resulted in many sessions and the team being the main point of contact in the aftercare part of the implementation. A different approach was taken by the organization that used the big bang approach, as they focused on the process owners and safety netters in a team. They presented the system and its concepts in a few sessions with the same group, allowing them to become the internal experts on the system, serving as the main contact point in one team. The only time the project team is involved in the problem solving area, is when the safety netters and project owners cannot solve the problem. This approach results in more time for the project team to focus on

the actual implementation instead of the aftercare for the whole organization. We expect this approach to be especially useful in a large organization, as the problem domain becomes larger and larger for a small project team to handle.

The second best practice can be defined as follows: *The training and aftercare of an implementation is best handled by a point of contact within a team. By training internal experts, the project team can focus on implementation of the rest of the organization, while avoiding answering questions that can easily be answered by an expert.*

This best practice will support the creation of traction in the organization, by leading the employees towards the new concepts introduced in the system and supporting them throughout the aftercare phase.

The organizations suggested different approaches when presented with the question what they would do different if they would start the implementation process now. A reoccurring problem was the lagging in one phase of the implementation. While the phases differed per organization, one of the lessons learned for them was to create a tighter schedule in order to close the gap between implementation and the finalization of the system. Although this is a logical step to take, planning will always be an important part of any information system implementation, which is therefore not a best practice specific to the implementation of a Case-handling System.

These two best practices summarize the lessons learned from the interviewed organizations, answering the third and fourth sub-research question. The first best practice also serves as a bridge to providing an answer for the fifth research question, how the processes of a Case-handling system can be modelled.

5. Towards a Modular Modelling Technique

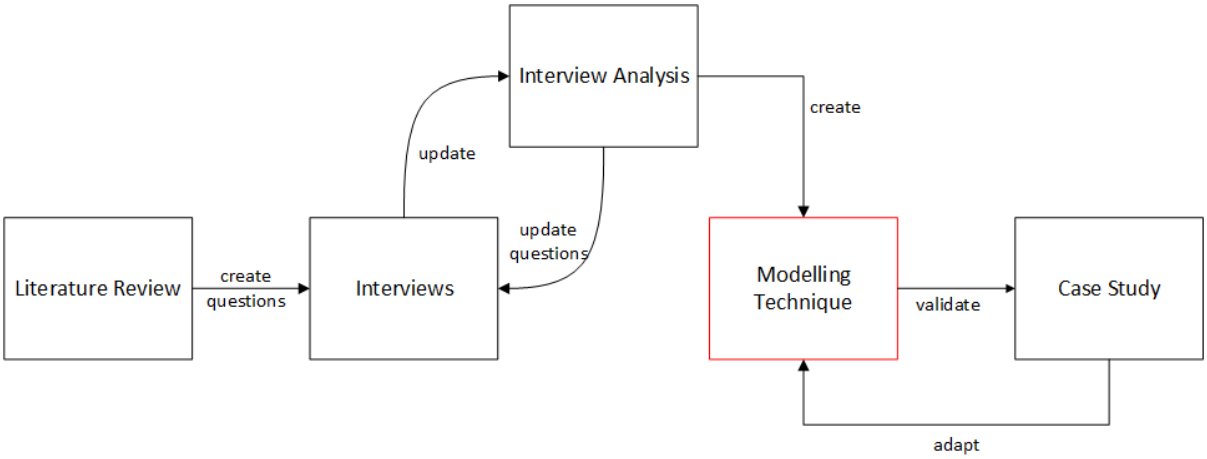


Figure 14. Research process – Creating the Modelling Technique

In order to give an answer to sub-research question 5, *"How can the processes of a Case-handling System be modelled, while including all relevant data but avoiding unnecessary complexity of the model?,* the results from the interviews and literature can be combined to create a modelling technique applicable to the design phase of a Case-handling System. As described in section 4.3, it becomes apparent that the interviewed organizations support the idea of generalized modelling to avoid specifics and to improve implementation time.

5.1. The generalization of processes

When modelling processes, the focus of process owners switches to specifics at a rapid pace, as described by the interviewees. The challenge provided with modelling is to avoid the specifics and generalize as much as possible. Generalization includes abstracting the given information and removing redundant or unnecessary information to make the model as broadly applicable as possible.

With a broad applicability of process steps, it becomes possible to use different parts of the models in different case types. This approach would match a modular way of modelling, creating general pieces of process steps that are used by all or a subset of the case types in the organization. The organization should focus on creating a transparent and generalized overview of the processes in order to support the redesigning process throughout the implementation of the new system. By focusing the modelling on a high abstraction level in the beginning of the process, the organization will gain a wider understanding of the processes that show overlap and can be implemented in the same way. To illustrate, Figure 15 depicts an example of a standard process, as it would be modeled on a process specific level.

The problem with a process like shown in Figure 15, is the redundancy of the last steps. Since step X, Y and Z all have the possibility to create a different end result based on the choices made in these steps, the result must always be sent to the customer, after which the case has to be closed and archived. This results in the cluttering of the model and lowers the readability of the overall process. The differences in the steps might be the channel used to send the result to the customer, or the evaluation criteria used based on the case type selected. This process model is specifically tailored to one process, and when starting the modelling of the next process, the logical step is to remodel this process from scratch as well. In order to avoid redundant work and promote the reusability of certain process fragments, we suggest to create a modular structure of certain sets of processes.

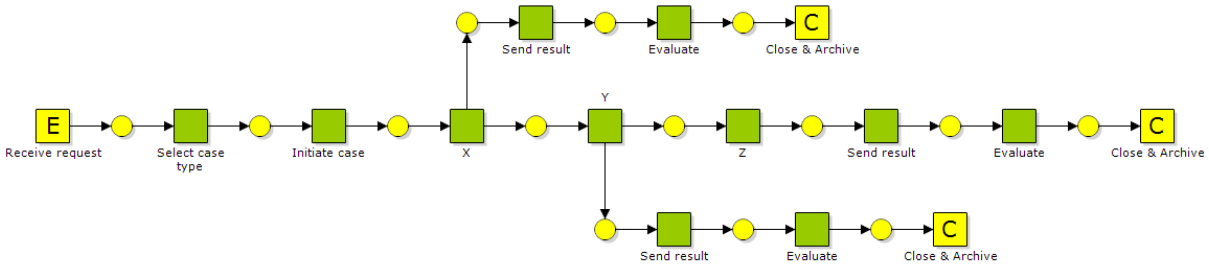


Figure 15. Example of standard modeled process

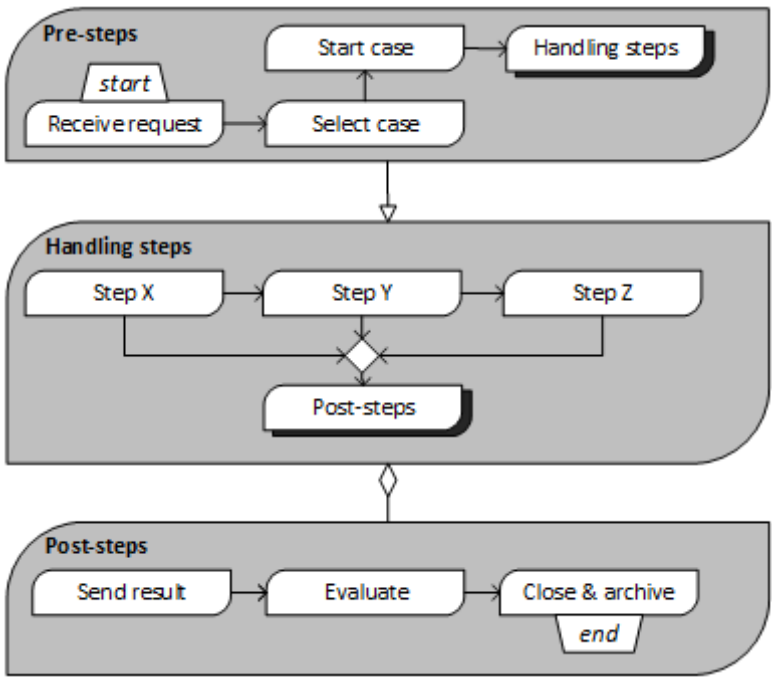


Figure 16. Modular process modelling approach

As described in section 4.3, a method for achieving this kind of generalization is the creation of case form or umbrella case types. By using a case form as an agreed upon skeleton for all related case types, modelling and implementation of case types in the system can be sped up significantly. Case types that are linked to a case form have the ability to use relevant pre-defined modules in their process model. Lindert and Deiters (1999) already suggested a similar process fragment approach in 1999. They focused on the inter-organizational aspects of WfMS, where fragments would benefit the information exchange between organizations.

The main reason behind creating *yet another modelling language*, is to create a generic technique that covers the high level reasoning for process models and specifically case types. When looking at the techniques discussed in section 3.2, the main difference can be described as imperative, specific languages on one hand and high level languages on the other. Organizations are familiar with the easy to use BPMN language, usually applying a certain form of this within their organization. The meta-modelling technique as described by van de Weerd and Brinkkemper (2008) uses a comparable easy to understand syntax to model implementation methods on a high abstraction level. Declarative modelling also focusses on high level modelling, but Reijers et al. (2013) noted that a pure declarative approach was not perceived as the solution. They propose the use of a *hybrid approach*, combining the high level characteristics of a declarative languages and the specifics of an imperative language to

create a language that can be used in all aspects of the implementation, from documentation to execution.

To apply a hybrid approach as proposed by Reijers et al. (2013) to the field of case handling, we therefore present a combination of the concepts of the meta-modelling technique and concepts of imperative languages such as BPMN and Petri nets in the form of a Declarative Modular Case Modelling (DMCM) technique. Figure 16 depicts the basic idea of the generalization of processes into modules, the model depicted in Figure 15 is remodeled and split up in three basic sections: Pre-steps, Handling steps and Post-steps. By using this approach, the pre- and post-processes can be created in cooperation with all process owners in the organization. This results in a transparent and generalized model that all teams use and agree upon. The handling module will then differ per case type, and can be specified to the needs of the process.

Figure 17 shows the intended use of generic modules in the modelling of case types in a Case-handling Systems. On top at the most generic level is the case form, created as the umbrella case type containing the steps all related case types have in common. From this case form, a number of generic modules are extracted, for example the three modules depicted in Figure 16. When constructing the specifics of the case types, the generic modules can be used as the basis for the case type. The case type created can use the modules to speed up the modelling and to create an organization-wide standardized view of those process steps.

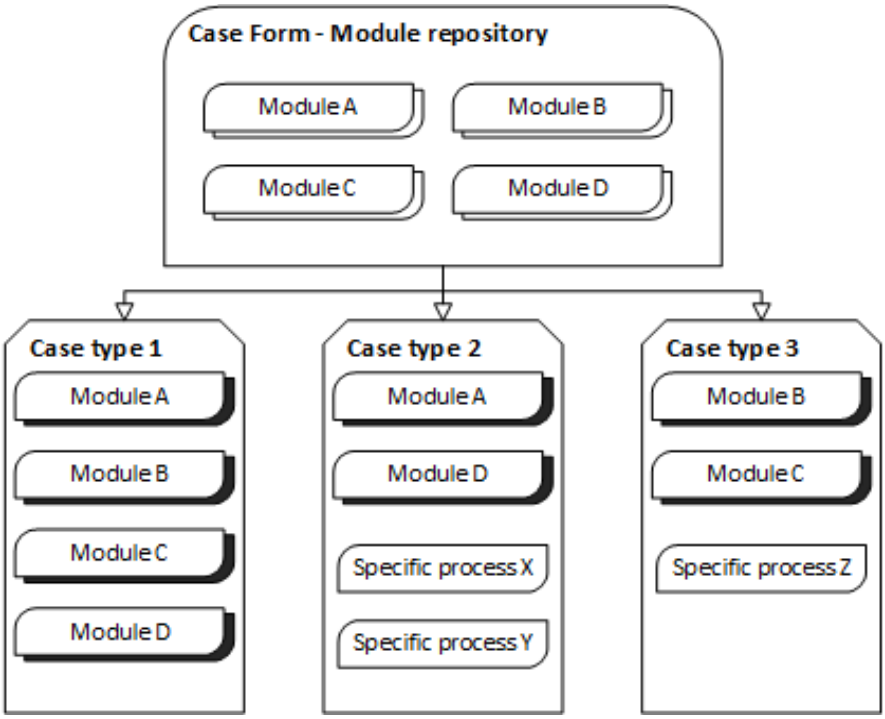


Figure 17. Position of case form and generic modules in relation to case types

As shown in Figure 17, case types can select the modules relevant to their processes. Where case type 1 uses all four modules of this specific case form, others redact certain steps and add the needed steps for that case type. Relevant information to these modules, such as the criteria and documentation, can and will most likely differ per case type. It is however in the interest of consistency and standardization to include generalized criteria in the generic modules, serving as a base line for the modules.

5.2. Declarative Modular Case Modelling syntax and semantics

In order to visualize the way of modelling the generalized process modules, a modelling technique needs to be defined. The technique presented below is based on fragments of meta-modelling for situational analysis, as described by van de Weerd and Brinkkemper (2008). They present their meta-process modelling technique as an adaption to the UML activity diagram, which is used to model the flow from and to activities in an information system.

The meta-modelling technique focusses activities on a higher abstraction level, omitting the specifics of the activity. The basic model consists of *standard activities, open activities* and *closed activities*. A standard activity contains no additional activities, where an open activity does contain additional activities and can be depicted in two ways. The first is a rounded rectangle with a white shadow that indicates that the activity is modeled elsewhere and only the overview is visible in the current context. The second way is the expanded version of the open activity, depicted by a gray rounded rectangle with relevant (sub) activities. The last activity form is the closed activity, which are not modeled elsewhere because the (sub) activities are unknown or not relevant in the current context.

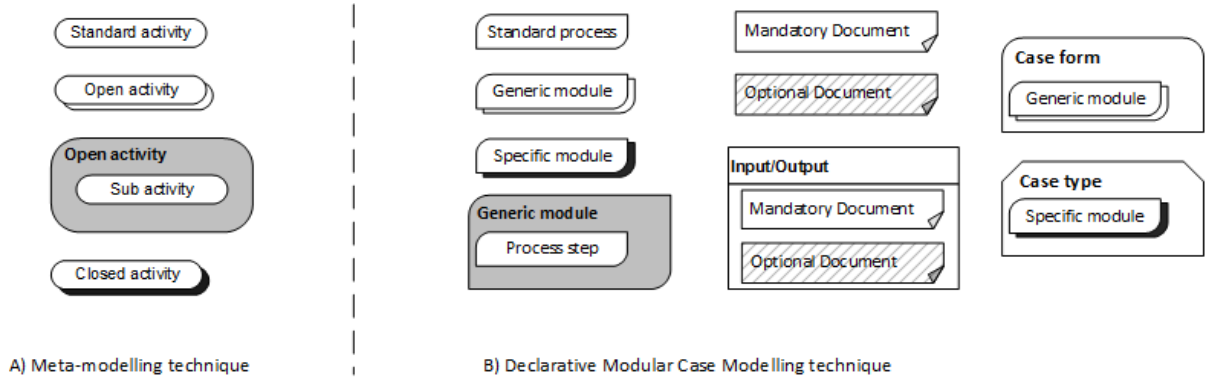


Figure 18. Meta-modelling and Declarative Modular Case Modelling

As shown in Figure 18, the syntax for the DMCM technique has a similar look and feel to the one depicted in meta-modelling. The model is depicted as a rounded rectangle with diagonal corners, with a *standard process* and a *generic module* modeled two ways, to support different levels of modelling and to allow the modules to be specified outside of the module. The contents of modules can be defined and modeled as per the wishes of the company. The *specific module* is a generic module specifically modeled for one case type. The specific module follows the structure set by the generic module and modifies this to the needs of the case type. The generic modules are encapsulated within a *case form*, depicted by a rectangle with rounded top corners, where the specific modules are a part of the case type, depicted by a rectangle with diagonal top corners. The final elements in the modelling syntax are *related documentation* and *document*. These elements can be used to link the documentation related to a module within the model.

A *closed module,* in respect to the closed activity of the meta-modelling technique, is omitted in the DMCM syntax. The reason for this is that we strive to create a technique that helps organizations understanding and generalizing their landscape. A closed module would mean that there is no context or knowledge of the module, defeating the purpose of organizational transparency.

As described in chapter 3, the modelling methods of organizations might differ. Where one organization might use BPMN, the other might have an internal standard that is used throughout the organization. By allowing the specifics to be determined by the user, the applicability of the DMCM rises. Another benefit of leaving the modelling specification to be determined per module, is the

possibility to use different modelling languages in one case type, for example allowing for a case type consisting of a BPMN module for process flow, Business Rules module for selection criteria and Petri net module for analysis of processes.

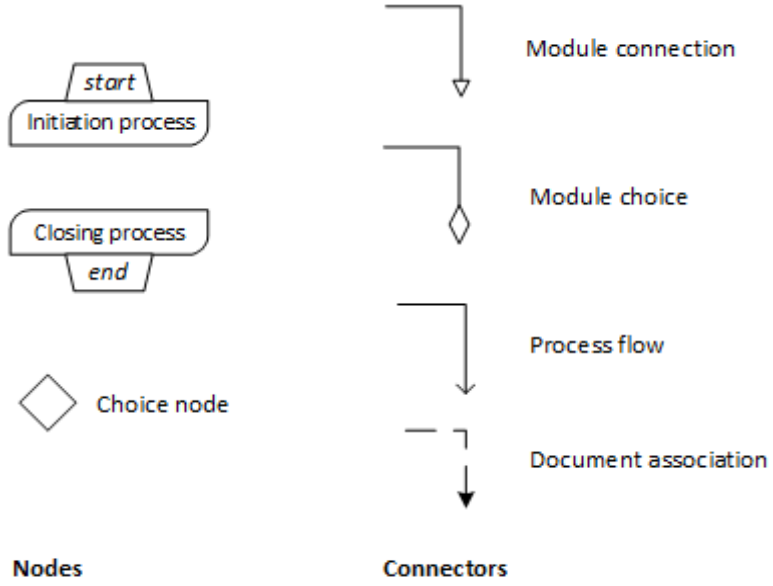


Figure 19. Nodes and connectors in the Declarative Modular Case Modelling technique

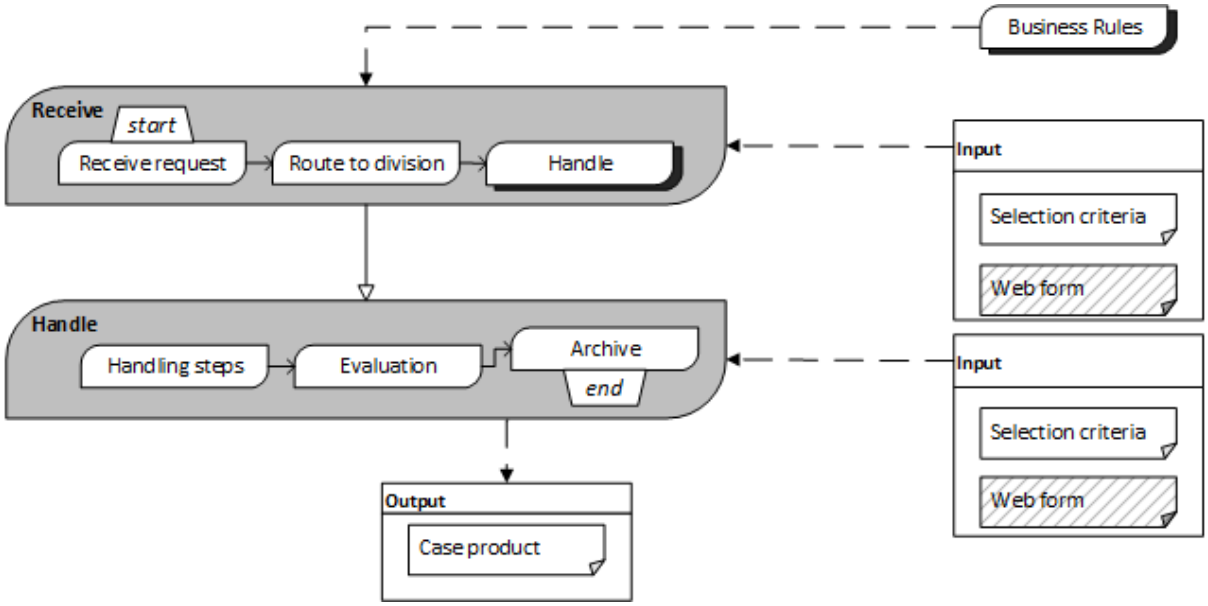


Figure 20. Example of modelling technique with related documentation

The elements of the DMCM are connected via several connectors, the *standard module connector* is used to show the link between different modules. The *open module connector* allows (sub) processes in the module to initiate the next module individually. In Figure 16, the module *Handling steps* is connected to the module *Post-steps* via an open connector. The sub-processes *X, Y* and *Z* are allowed to initiate the next module if the process requires the case to be finished.

Processes are connected via the *process flow connector*, showing the order of execution of the process steps. When no flow connectors are modeled, the processes can be executed in any order. The nodes depicted in Figure 19 are used to indicated the states of the model, the *start* and *end node* show the beginning and ending of the case type respectively, adapted from the declarative modelling technique

(Reijers et al., 2013) and a *choice node* depicts choices in which process to execute based on a *condition*. To support the link between modules and documents related to the module, a *module association* is defined. This association links the modules to the related documentation. An example of the model is given in Figure 20.

The meta-model in Figure 21 depicts the intended semantics for the DMCM technique. The main element is the *module*, which can link to any number of other modules. The *activities* represent the process steps within the module, where a module has a one-to-many relationship with the activity and are tied to each other using a flow. The documents related to the *in-* and *output* of a module are linked to the module with a zero-to-many relation, and documents can be either *optional* or *mandatory.*

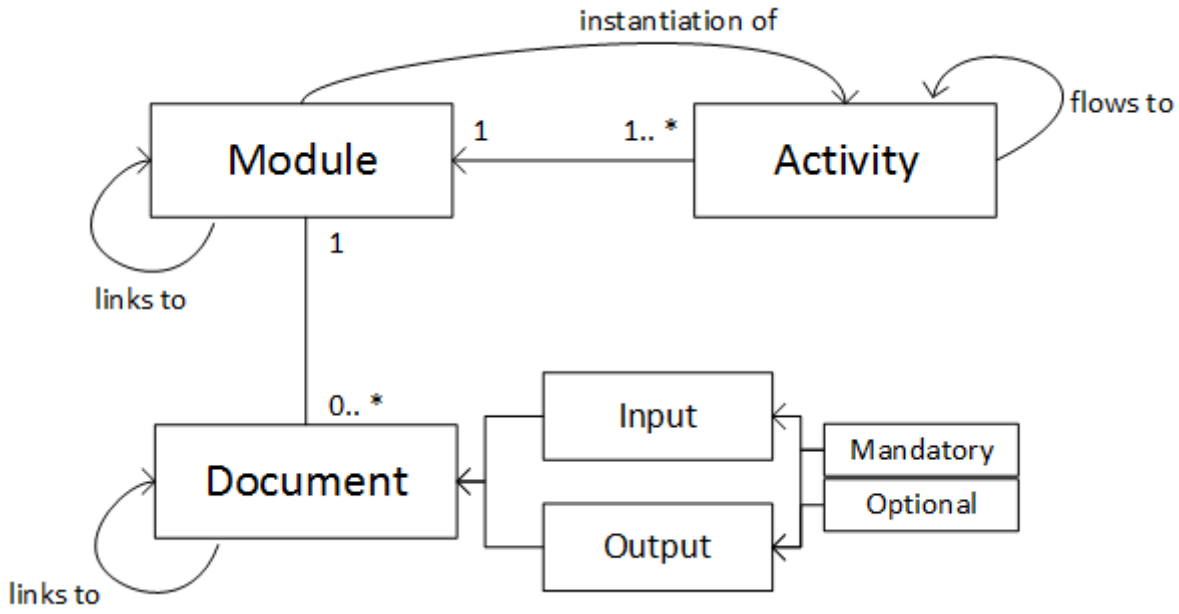


Figure 21. Meta-model of the technique with dependencies and multiplicities

In order to describe the semantics of the elements as presented in this section, we will explain the flow of a model using the elements of the language. The model depicted in Figure 22 shows the syntax of the DMCM technique and the flow descriptions in the declarative technique Declare (Pesic & van der Aalst, 2006; Reijers et al., 2013). The intended flow of the activities is either *ABEFCGHDIJ or ACGHBEFBIJ,* where the only difference is the order of execution at module B and C. The two modules can also be executed in parallel, resulting in any order of activities E, F, G and H. The use of the input documents that are associated with the main module are optional for the activities and other modules. Each of the activities can only be executed once, unless a flow connector would feed back into a previous activity.

Since the documents are connected to the main module, they have an implicit relation with the modules B, C and D, allowing activities E through J to use these documents where needed. The documents restricted to one module only can be depicted by drawing the association line between the specific modules. The final product that is delivered is an output document for module D, and only by fulfilling the activities I and J in module D can this document be generated.

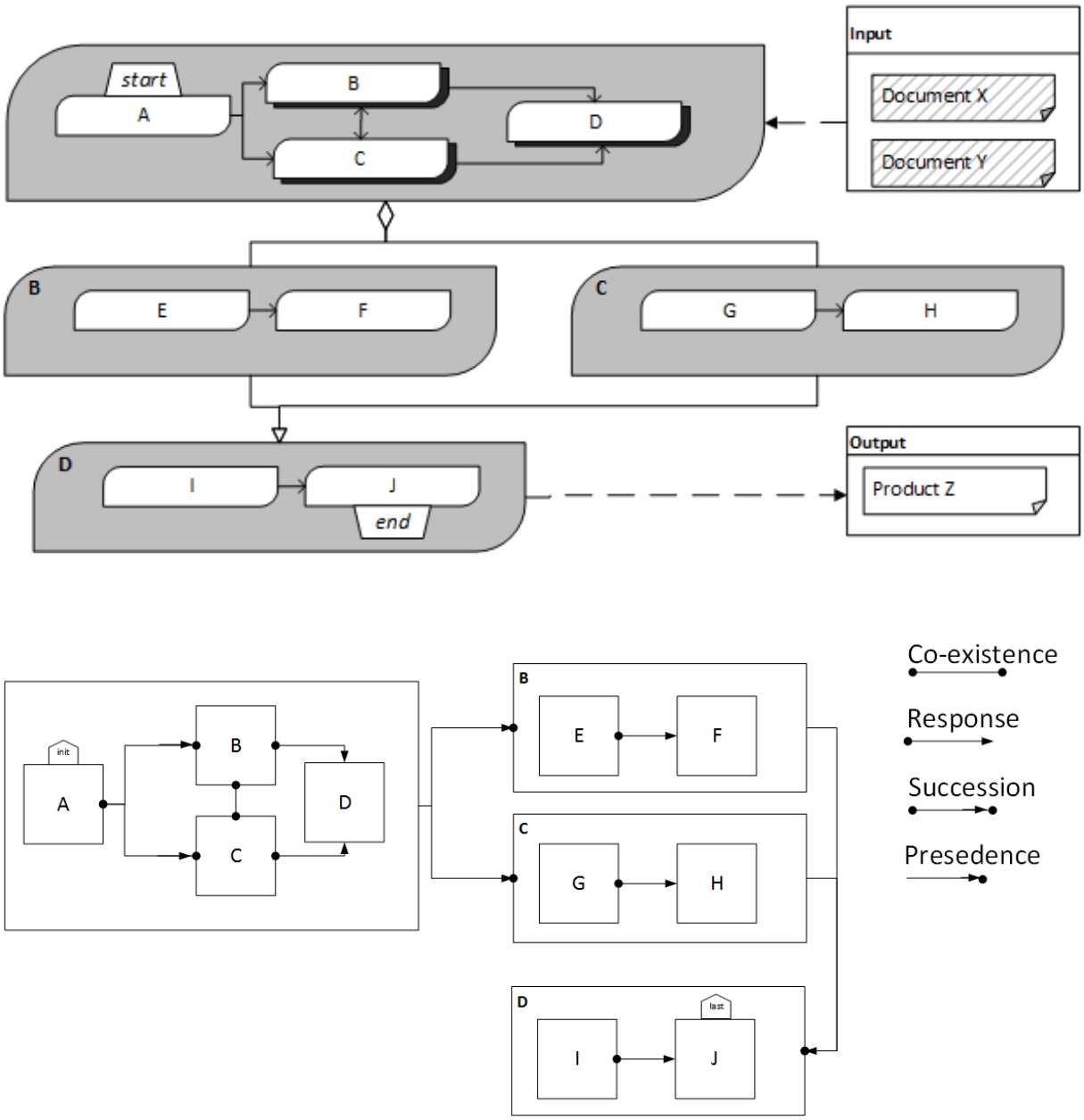


Figure 22. Model with dependencies in the Declare language (Reijers et al., 2013)

6. Case study

This chapter uses the literature and gathered data the previous chapters, and applies it to a case study at the *Netherlands Food and Consumer Product Safety Authority.*

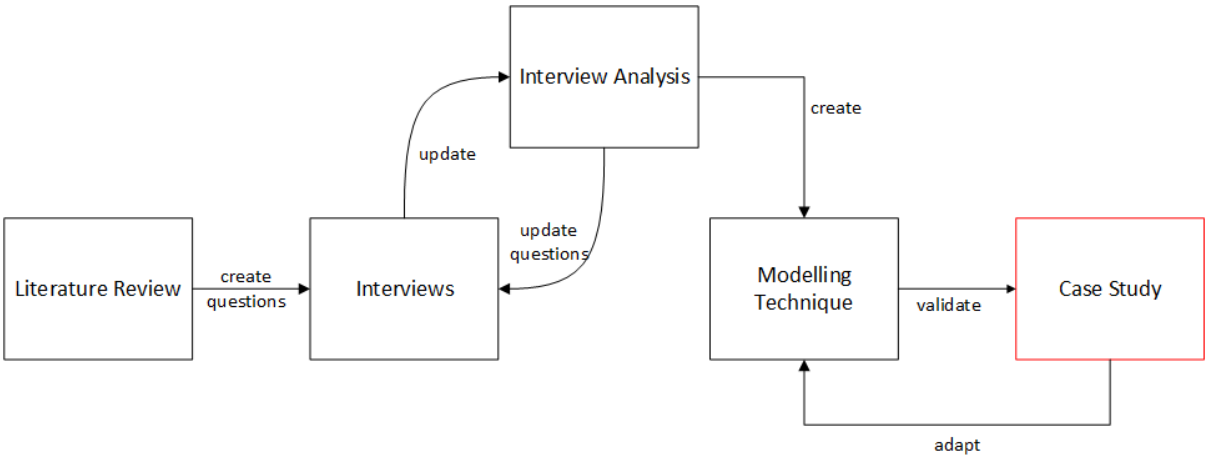


Figure 23. Research process – Case study

6.1. About the case company

The *Nederlandse Voedsel- en Warenautoriteit (NVWA)*, is one of the governmental organizations that is in the process of making the switch to a Case-handling system. The NVWA has started a program to tackle the current information management problems that resulted from the merger of the three organizations that now make up the NVWA (Kloese, Neuteboom, & Rutges, 2014). With this program, the NVWA wants to get a better overview of the organizational processes by creating uniform processes and using modern and fitting IT-systems to support these processes. The better overview refers to opening the information of all divisions to the right staff when they need it, in which a case-handling environment can be a supportive factor.

The information management of the NVWA is not on the level the organization wants it to be, which is confirmed in multiple reports over the past years (Kloese et al., 2014). Organizational change is needed to complete the merger of the three original organizations, who still use their own legacy systems separately from each other, resulting in almost no collaboration between divisions.

The main goals of the program *Blik op NVWA 2017* (Kloese et al., 2014) are to create uniform and renewed processes that fit with modern IT-systems. The cost of making changes in current legacy systems are high and take a significant amount of time due to the complexity of these systems. The systems are stand-alone and the communication between them is almost non-existent. The technology on which the systems are based has become obsolete and hard to work with. The support for mobile devices is not up to par with current technologies, there are almost no possibilities for improvement and there is not enough information exchange between internal and external parties.

The organization needs to make drastic changes to adhere to the standards set by the government (Dijksma, 2013). The structuring of the information management of the NVWA should create opportunities for better analysis and control. To do this, the processes need to be reformed, modern technology has to be applied, and staff should be educated in the use of these new systems.

The primary focus of the NVWA in this stage of the organizational change is to analyze the best fitting IT structures and to reorder and renew the processes that the new environment should support. The program is split up in three levels of execution, '*Determine', 'Realize'* and '*Support'.* The levels are

executed in parallel, after determining the right support for a certain process, it can be realized and supported. For an organization renewing their whole IT landscape it is important to have guidelines based on the lessons learned from other organizations. Fernandez and Rainey (2006) describe eight factors that can influence the change within a public organization in different points of the process.

The factors described by Fernandez and Rainey (2006) are:

1. *Ensuring the need* – The management of the organization must support and communicate the need for the change in order to persuade the rest of the organization.
2. *Providing a plan* – Creating a plan of action, describing the goals of the implementation and setting deadlines for the project.
3. *Building internal support for change and overcoming resistance* – By informing employees and providing them with the necessity of the change, including them in the process and elaborating on the benefits of the new system.
4. *Ensure top-management support and commitment* – By creating a core group that promotes the change, the internal support can grow. The project team can fill the role of this group.
5. *Building external support* – External support is an important aspect for an organization in the public sector, as the government has to agree on the approach and necessity of the plan.
6. *Providing resources* – Providing employees with training sessions and material to study the new system.
7. *Institutionalizing changes* – The changes need to become part of the daily routine of an employee, which automatically becomes a necessity when the old system is replaced by the new one.
8. *Pursuing comprehensive changes* – The change must be done in a way that is comprehensive for all involved actors, allowing the implantation to proceed smoothly.

The NVWA has already fulfilled the first five factors depicted above. Factor 1 and 3 are provided by both internal and external support, example cases were distributed amongst the employees to create an overview of the to-be situation. The need to build external support was not necessary, since one of the triggers for these changes came from the government herself. Lastly the top-management of the organization was actively involved in the writing of the project plan, which fulfilled factor 4 and 2.

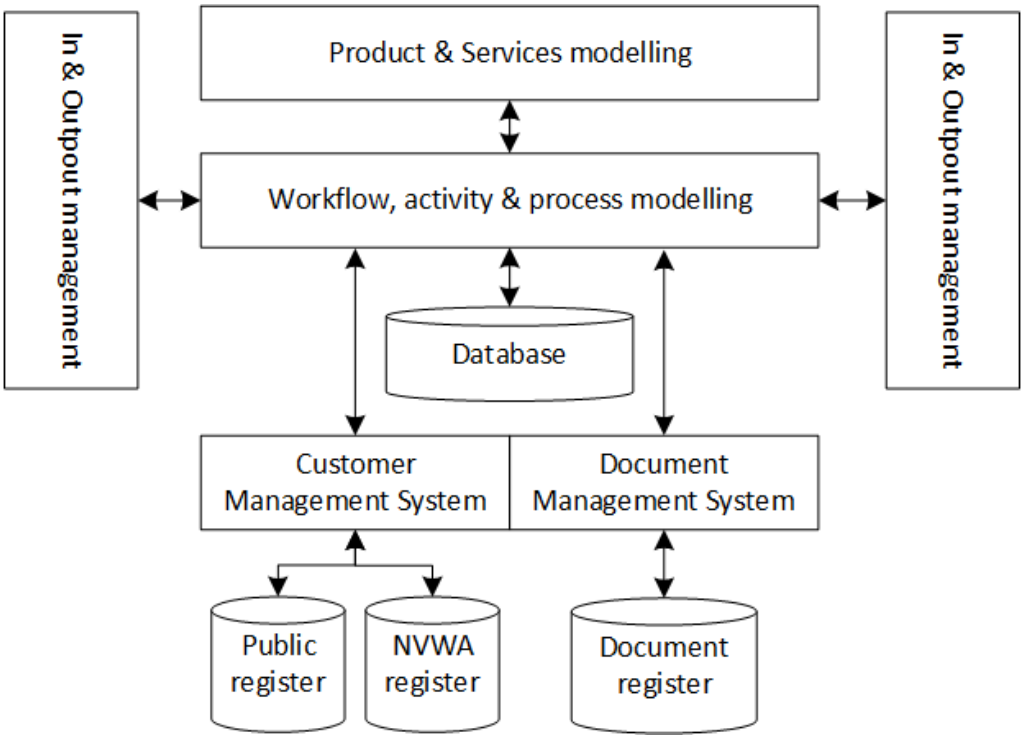


Figure 24. Case handling concept for NVWA

This results in the last three factors as the main points of interest in this stadium of the organizational reform. How can resources be changed or added to support the new process, how can the changes become embedded in the organization and how does the organization ensure the transformation of the organization by iteratively updating subsystems to the desired state.

The concept as depicted in Figure 24 has a similar design as the proof of concept of Hee et al. (2008) including a document manager, a workflow engine and a database that includes all the processes that can be used in a case.

The definitions given by the NVWA show an overlap with those found in the literature. The situation at the NVWA presents an opportunity to validate a solution to the problem described in the previous chapter. As the NVWA is in the early stages of transforming their organization, the organization can benefit from obtaining lessons learned from other organizations that have implemented case handling and take the results from the validation with them during the implementation process.

6.2. Current status at the NVWA

The NVWA is currently focusing on the redefinitions of their processes using a cluster based approach. The first process which was remodeled is the *reporting process.* This process is the front-end of the system where all notifications, such as phone calls, emails, web forms and postal pieces are routed to the correct process, the matter of completeness of the request is validated, the assignment of the right cluster and the closing/archiving of the notification.

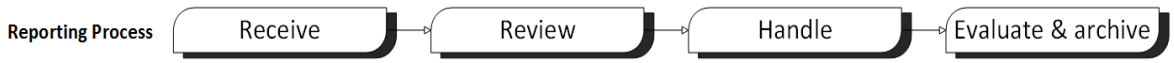


Figure 25. Simplified version of reporting process

A full version of the reporting process can be found in Appendix C. According to Hählen (2015), the project team has chosen a process based approach on the remodelling of their processes. They apply an end-to-end method to work through the processes with the process teams, resulting in a specifics oriented process model. This approach differs from the generalized modular approach as described previously, since this focusses more on generalizing the data that is needed to create the end product.

The project team is currently analyzing user stories per process and working through these to optimize the process step by step. They look at the overlap in the user stories, scrapping or moving steps in the current model to optimize the process as good and specific as possible. At the start of the project, a general 'project start architecture' has been created for the specific processes. This architecture describes the process on a high abstraction level and is meant to create a clear scope of the process redefinition. The document has some aspects of the intended modelling approach, abstracting the information on a process-specific level, but has been executed at the start of the remodelling of that specific process. The organization has not inventoried other processes that might have some overlap with this process on this level of abstraction.

6.3. Re-modelling the process

Taking the modular modelling approach as presented in the previous chapter into account, several changes can be made to the reporting-process model. Appendix C shows the full process as modeled after the current version available at the NVWA. The complete model contains 48 process steps and 32 choice nodes. An observation that can be made when looking at the complete process is the multiple use of steps such as 'Contact customer', 'Request for information', 'Feedback to reporter' and 'Close notification', as the result of a choice node creating a short branch resulting in the closing of the process.

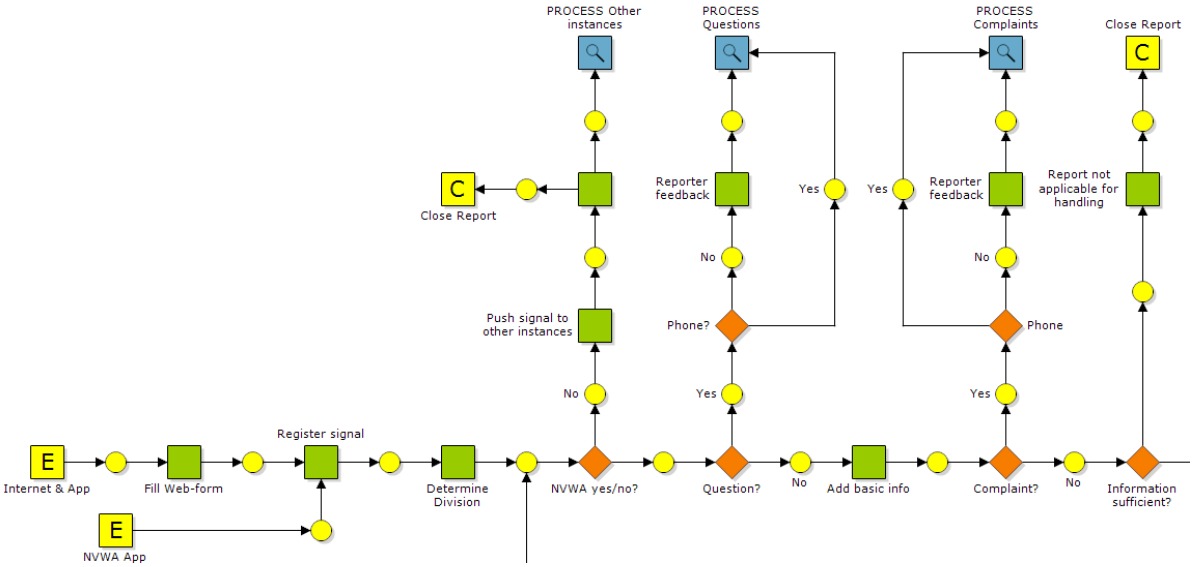


Figure 26. Example of branching within the process model

The branching of the process flow results in many duplicate nodes and end points of the model, ultimately generating a large specified process model as shown in Appendix C. The process model is the result of the process-based approach the project team took as described in section 6.2.

The simplified version of the reporting process shown in Figure 25, can be applied to most of the processes in the organization on a general level. Similar process steps and process modules can be found in the reporting, programmed-handling, questions and complaints processes. The overlap found does not necessarily cover the entirety of the processes, but rather modules and other fragments that could be re-used in the implementation process.

The organization has not focused on the mapping of all processes on a general level, which is in contrast with the lessons learned from the interviewed organizations. The mapping of a general overview can supply the organization with benefits like transparency on processes, agreements on process flows and reusability of certain fragments in different processes. According to Hählen (2015), the four processes currently in the redesign phase show overlaps in structure on a high level. Every process receives certain signals or cases, which go through a review phase to check the case on completeness and whether this case has to be handled in the process. After the reviewing process the case is handled based on the available information and according to the criteria set in that specific process. After the handling the case is evaluated and archived in the system, bringing the process to a close.

The overlaps between the four processes are not one on one, but have a similar flow on the generic level. The questions and complaints processes for example might differ in the reviewing module, but show an overlap in the evaluation and archiving module. Figure 27 illustrates the similarities of the four processes.

To extract reusable moadules from the from the reporting process model, we need to create an overlapping generic model of this process. As described in the figure above, the process should be able to be split into the four basic modules. The overlays in Figure 28 show the different positions of the modules in the model. The light red overlay indicates the receiving and registering of the signal, blue the reviewing of the information, yellow the handling of the signal and grey the evaluation and archiving.

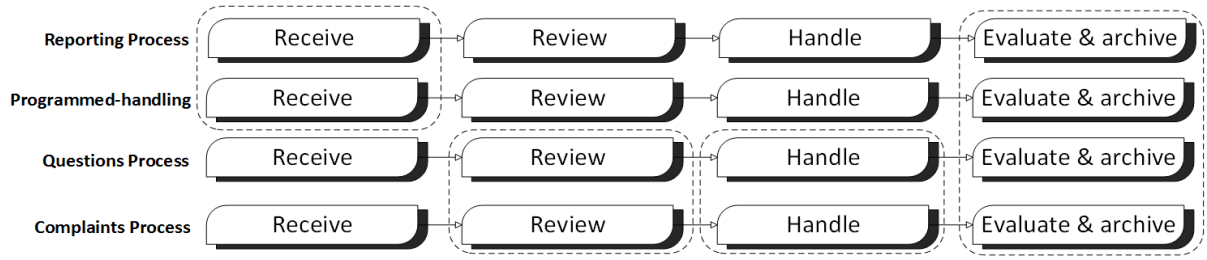


Figure 27. Overlapping modules across different processes

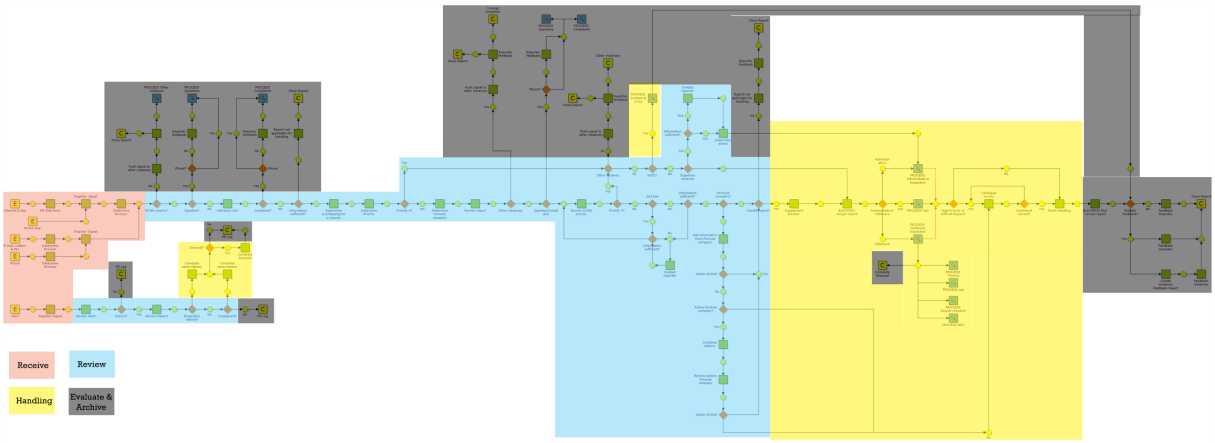


Figure 28. Reporting process with colored overlay.

What becomes apparent with using these colored overlays on process, is the grouping of the related processes. At the front of the model we see the receiving and the registering of the signal, which is a logical first step in the reporting process. The reviewing processes consist of a chain of choice nodes, confirming if the report is relevant for the NVWA, information supplied is sufficient for handling and when it is not, extra information is gathered from either the client or the organization responsible for this specific case. When the report information is found to be sufficient, the handling process is started. The correct process is selected and the case follows the criteria defined in the process. The outer bounds of the process model are the evaluation and archiving process steps, and are scattered throughout the model. The obvious reason for this are the many choice nodes in the reviewing part of the model. When exceptions move the report towards a different organization or the questions/complaints processes, the case does not have to move towards the handling phase. Using the modelling techniques described in this chapter, the total process can be generalized as depicted in Figure 29.

This high level view of the processes shows the general flow of the process model, where the review module goes through a choice node to determine if the report is applicable to handle and otherwise move straight through to the evaluation and archiving of the report.

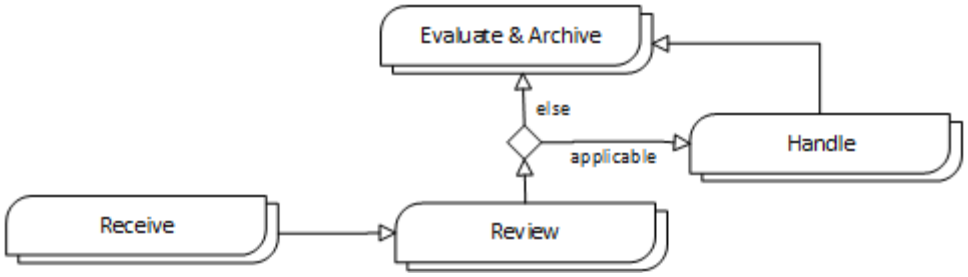


Figure 29. The reporting process as generic modules

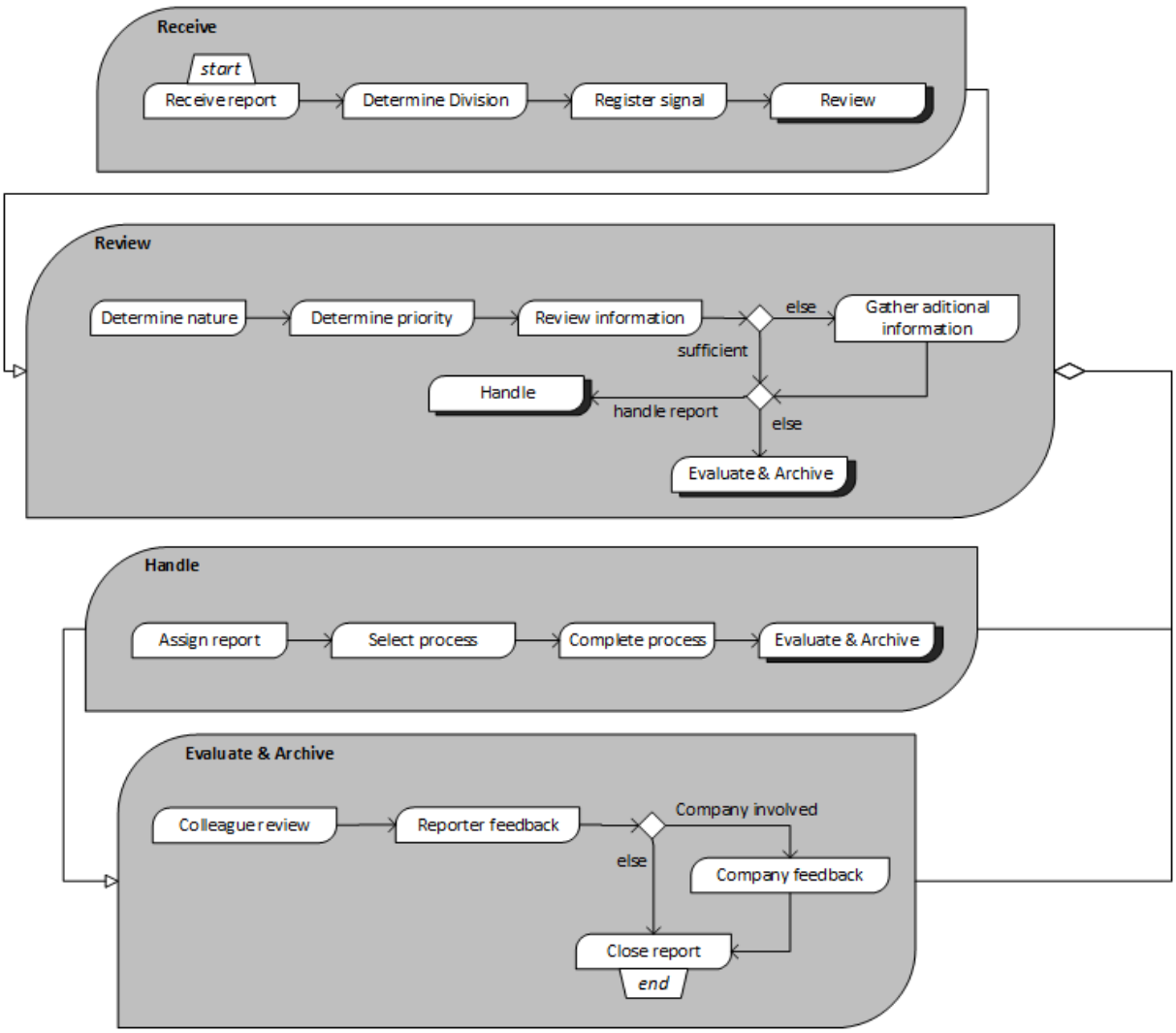


Figure 30. Generic modules of the reporting process with process steps

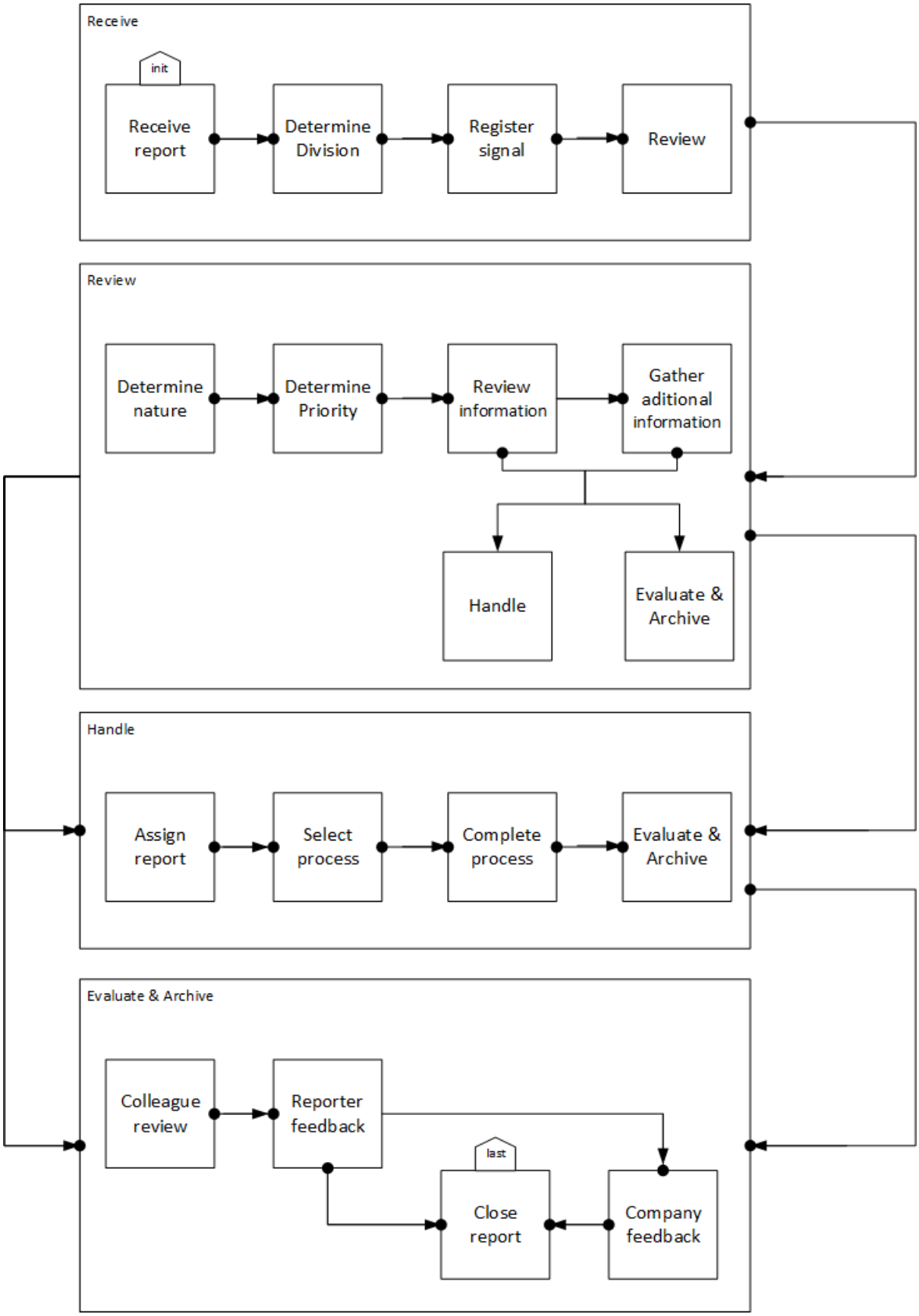


Figure 31. Declare model of the reporting process

The modules are further specified in Figure 30 and Figure 31, in the review module, the basic steps are the receiving of the report, determining the correct division for reviewing and registering the report signal for that division. The end node represents the next module the process will enter, the reviewing module. The review part of the process has many specific choice nodes that route the process into the right direction. In order to generalize this, the *review information* process has been used. After the priority of the report has been determined, the review results determine if additional information is need from either a reporter or related company. Afterwards another choice node determines if the report should be handled by either the NVWA of another organization. If the latter is the case, the process can move on towards the Evaluate & Archive module.

The Handling module then starts with the assigning of the report to a certain team or employee, which results in the selection of the correct process. The process is then handled according to the process steps and criteria that are linked to that specific process. After the process has been completed, the report can exit the handling phase and enter the evaluation phase.

The Evaluation and Archiving module start with a *colleague review* in which the whole process of reviewing and handling the report is checked for mistakes in procedure and results. After completion of this evaluation, the result from the report, either a complete result of the handling cycle or a notification to the reporter that the report is not being handled by the NVWA and that the report is pushed to the relevant organization. If a company was involved in the additional information gathering in the review module, that organization is also notified of the results of the report. If all steps are completed the report is closed and archived into the case dossier of the reporter and related companies.

As shown in Figure 30, using modules to generalize complex processes results in a less cluttered and complex module, still capturing the basic elements of the process as a whole. The related documentation was left out of this case study, as the related documentation will be connected during the actual implementation (Hählen, 2015).

6.4. Evaluation, validation & next steps

The modules described in Figure 29 and Figure 30 match the remodeled reporting process as depicted in Appendix C. As a next step for the NVWA, we suggest to take the gathered best practices described in section 4.4.1 into account and focus on creating a generic overview using the modelling technique presented in chapter 5. We believe the organization will benefit from focusing on creating a generic overview before modelling the specifics any further. The reuse of certain modules can benefit the project team in the redefinition and implementation of the case types. The coherent understanding of the landscape will put every involved actor on the same level and agreement on the way the processes are executed and modelled.

The basic steps in using the DMCM language are to create an overview of the generic overlaps between different processes. After the overlaps have been found, the process owners and project team determine the generic process flow and split these up into modules. The benefit of the modules with this approach is as described above, the coherent understanding and agreements of the execution. After the generic modelling step is completed, the process owners and their team members can take these generic modules and specify them to fit their needs. The modelling language that is used for this specific step can be filled in by the organization itself, such as BPMN.

To apply this technique, the project team should be instructed in the use of the language and associated concepts, after which they can bring the process owners up to speed with this new

approach. The language is designed to be familiar to those who have experience with modeling techniques and easy to read and understand for those who do not.

The first reactions of the language at the NVWA were positive, but might not fit the style of the organization, according to Hählen (2015). The organization sees the benefit of starting with a generic view, but their culture is focused on getting straight to the specifics, due to their expert knowledge of the processes. Further validation of the language can be achieved by adopting a multi-case study, which we will expand upon in the next chapters.

7. Discussion

During this research, we focused on gaining a wider understanding of the case handling paradigm and matching the experiences of organizations with the scientific body of literature that is available on the subject. Like any research, there are some limitations to be found in the research approach and results. The exploratory nature of a design science approach creates research results that would not have been expected when this research started. This chapter elaborates on the limitations and validity of the thesis project.

7.1. Limitations

The literature review that has been performed was based on a narrative and snowballing method. This approach is common in exploratory research, but has the downside of lacking explicit search patterns in order to support the repeatability of the research. Paré et al. (2014) describes this approach as opportunistic, avoiding the generalization of the complete body of literature.

The semi-structured nature of the interviews had up- and downsides, where some interviewees could passionately speak about the implementation process and mistakes that have been made, others had to be led more actively through the prepared question list. This resulted in different volumes of information gathered from the different interviewees. The second limitation to the interview part of the research is the amount of interviewees. The conducted interviews at the current organizations resulted in interesting data, forming the basis for answering several sub-research questions. It would be beneficiary to the validity of the research to interview more organizations to gain an even wider understanding of problems encountered and lessons learned.

The modelling technique was only tested on a single case at one case company. In order to fully test the applicability and usability of the technique, the technique should be used for the duration of the whole implementation, monitoring the results and adapting the technique where needed. The benefits of a multi-case setup can be found mainly in the application of the same generic modules in different case types. This will allow the technique to be fully tested, focusing on the reusability of the modules. This, however, was not possible due to the time restraints of the thesis project and the available reworked processes.

7.2. Research validity

The validity of the research will be described according to *construct validity, internal validity, external validity* and *reliability.* Construct validity focusses on the correct adoption of operational measures to study the concepts of the research. When using a case study, a way to adhere to this is to use multiple sources of evidence and have experts verify the results (Yin, 2009). This research has used multiple sources, the interviewees serve as experts on the organizational domain and the literature review serves as the scientific source. Documentation gathered from the case company is also embedded into the research results.

The internal validity focusses on the structure of the research approach and techniques used. This research dedicated chapter 2 to the explanation of the approach and the techniques used in the research. This supports the integrity of the results and supports the repeatability of the research.

External validity focusses on the generalization of the results and its applicability in other areas outside of the research scope. The modelling technique is mainly focused on the implementation of a Case-handling System, since the reusability of modules has its main benefits at the creation of comparable case types. However, the technique can be used to create a generic overview of the organizational landscape outside of this context, supporting transparency and agreement on the structure of the processes within the organization.

As mentioned in the previous section, the repeatability of the research shows some limitations. There is no method for literature review presented and the conducted interviews were of a semi-structured nature. However, the application of the modelling technique can be repeated in organizations at the start of their implementation phase. The technique can support the generalization of processes and result in similar models as shown in chapter 6.

8. Conclusion

This chapter will summarizes the answers on the research questions as described in chapter 2. The goal of the research was to bridge the gap between the scientific literature and current state of the art applied within organizations and to create support for the design phase of the implementation of a Case-handling System. We delivered a definitive definition of case handling, gathered best practices from organizations and proposed a modelling technique to support the modelling of processes and case types.

The main research question of the research, as described in chapter 2, is as follows: "H*ow can the design phase of implementing a Case-handling System be supported using a modelling technique?"*

We will first answer the five sub-research questions and conclude with the answer on the main research question. After this we will describe the possibilities for future work in context to the deliverables of this thesis.

SRQ1 *What is the definition of case handling?*

By taking the concepts found in literature and data gathered from interviews, we constructed the following definition of case handling: *"Case handling is a standardized, transparent and product-driven approach to Business Process Management that provides all case-related data to involved actors, focusing on the case as a whole by removing the separation between processes and data."*

This definition covers the load of the concepts used within case handling a combines the aspects of organizations with the established scientific basis.

SRQ2 *How does case handling compare to conventional Workflow Management and Business Process Management?*

Case handling is, as described above, a standardized, transparent and product-driven approach to Business Process Management. The main difference between a Workflow Management System and a Case-handling System is the lack of so-called "context tunneling". In a CHS, the focus is not on the control flows of the process, but on the data needed to execute the activities in the process. Data is available to all actors involved with the case instead of every employee only getting the data needed to complete their assigned task. This allows for employees to understand the context of a specific case, allowing them to use their knowledge more effectively.

SRQ3 *How can a Case-handling System be implemented in a comprehensive and unambiguous way?*

The data needed to answer this and the next sub-research question was gathered from the conducted interviews. The main consensus of the results was that the processes within the organization should be generalized as much as possible to allow the implementation to speed up. According to the interviewees, focusing on every specific aspect of the processes and modelling every exception to the process take up an enormous amount of time until everything is implemented across the width of the organization. By gathering process owners and discussing the similarities between processes and activities, a generic overview can be created on

which all involved parties agree. This allows for faster modelling of the specifics, and supports consistency across the system.

SRQ4 *What are the most occurring problems during the design phase of the implementation process and what would the interviewed organizations do differently if they would start the implementation now?*

The best practices that have been defined in chapter 4 can be seen as the solutions for the encountered problems within the organizations. As described in the previous question, one of the main problems was that employees with in-depth knowledge of the process tend to focus on specifics and exceptions. The best practice defined for this problem is: *Create a generic overview of the current state of processes in the organization to support the generation of traction amongst management and employees and to create a coherent view of the processes on which all process owners agree.* This practice also supports the use of the modelling technique as described in chapters 5 and 6.

The other reoccurring problem was to prepare the employees for the change to a new system. After all, the employees use the system as their daily driver and they cannot lag behind when the system is implemented, as this would be a big problem for the organization. A solution for this is to focus on intensive training and to create experts that can help other employees with the problems they have with the system. The best practice defined for this problem is: *The training and aftercare of an implementation is best handled by a point of contact within a team. By training internal experts, the project team can focus on implementation of the rest of the organization, while avoiding answering questions that can easily be answered by an expert.*

These practices cover the changes that the organizations would focus on if they would start the implementation with the knowledge they have gathered over the years.

SRQ5 *How can the processes of a Case-handling System be modelled, while including all relevant data but avoiding unnecessary complexity of the model?*

The Declarative Modular Case Modelling technique described in chapters 5 and 6 is the result of the data gathered from the interviews and scientific literature on modelling languages. In order to make the DMCM technique comprehensive and not to complex, the focus was put on creating a syntax that showed similarities with comparable high level modelling techniques. The technique focusses on creating a generic high level overview that can be used throughout the organization, on which the specifics implementations can be based.

The use of modules that contain fragments of a certain process result in the reusability of previously modelled processes. These modules are contained in a case repository from which the specific case types can be formed by combining the modules that are present in the process, after which the process steps can be further specified. With the reusing of modules, the focus on standardization within case handling is supported, allowing for the faster implementation of the system.

With the answers to these sub-research questions, the main research question on how to support the design phase with a modelling technique can be answered. As gathered from the best practice described in SRQ4, it is important to apply a generic approach before jumping to the specifics. The DMCM technique focusses on this generalization as described in SRQ5, and can support organizations in the design phase of the implementation.

We believe that this research contributes to the understanding of the case handling paradigm and bridges a gap between the scientific and organizational parts of case handling. The best practices and the modelling technique can assist organizations in the early stages of the implementation of a Case-handling System and support the complex phase of designing the processes within the organization.

8.1. Future research

There are several future research possibilities of the concepts described in this thesis. As described in the previous chapter, there are opportunities for extending the scope of the research to include more organizations as interview data source of as supplier of use cases for the modelling technique. The DMCM technique would benefit from an embedded multiple-case study, spread across different organizations and across different use cases.

With the involvement of more organizations, there is a possibility of creating a maturity model for Case-handling Systems. Research can be done in order to determine all factors that are used within the paradigm and how organizations can focus on creating a more mature system. For a reliable maturity model to come to light, a structured approach involving all Dutch municipalities is highly suggested. The popularity of the paradigm is mainly present at these organizations, making them the best sources for this kind of information.

Another research opportunity is to embed the concepts of the DMCM technique into a tool that can be linked to the actual Case-handling System. Currently, the syntax of the technique are confined to a Microsoft Visio stencil, meaning there is no way to actually analyze the model directly. A tool that is linked to the system or even a part of the system would allow organizations to directly model and adapt their case repository and case types on the go. Changes made within the repository could then automatically feed back into the case types that employ a form of that module, allowing for editing case types on the fly. Allowing the tool to convert the DMCM language to a more specific imperative language such as BPMN can also benefit organizations in the use of the DMCM technique and will add value to the language as an useful asset in the design phase of the implementation.

References

Dekker, C. (2009). Zaakgericht werken - Betere Informatievoorziening en Betere Dienstverlening bij de Overheid. *IenPM*, 81.

Dijksma, S. A. M. (2013). Kamerbrief over toezicht slachterijen NVWA. *Ministerie van Economische Zaken*, p. 7.

Fernandez, S., & Rainey, H. G. (2006). Managing Successful Organizational Change in the Public Sector. *Public Adminisration Review*, (April), 168–176. http://doi.org/10.1111/j.1540-6210.2006.00570.x

Günther, C. W., & Aalst, W. M. P. van der. (2005). Modeling the Case Handling Principles with Colored Petri Nets. *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools CPN 2005*, 211–230. Retrieved from http://is.tm.tue.nl/staff/wvdaalst/publications/p291.pdf

Günther, C. W., & Reichert, M. (2008). Supporting flexible processes with adaptive worklow and case handling. *Workshop on Enabling …*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4806924

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. http://doi.org/10.2307/25148625

Hove, S. E., & Anda, B. (2005). Experiences from conducting semi-structured interviews in empirical software engineering research. In *Proceedings - International Software Metrics Symposium* (Vol. 2005, pp. 203–212). http://doi.org/10.1109/METRICS.2005.24

Kloese, A., Neuteboom, J., & Rutges, L. (2014). *Verbeterlijn Procesvernieuwing, Informatie & ICT - Programmaplan Blik op NVWA 2017*.

Kueng, P. (2000). The Effects of Workflow Systems on Organizations: A Qualitative Study, (1998), 301–316. http://doi.org/10.1007/3-540-45594-9_19

Lindert, F., & Deiters, W. (1999). Modelling Inter-Organizational Processes With Process Model Fragments. *Workshow Processings Enterprise-Wide and Cross-Enterprise Workflow Management: Concepts, Systems, Applications (Informatik '99)*, 1–9.

Mutschler, B., Reichert, M., & Bumiller, J. (2008). Unleashing the Effectiveness of Process-Oriented Information Systems: Problem Analysis, Critical Success Factors, and Implications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 280–291. http://doi.org/10.1109/TSMCC.2008.919197

Mutschler, B., Weber, B., & Reichert, M. (2008). Workfow Management versus Case Handling: Results from a Controlled Software Experiment. In *Proceedings of the 2008 ACM symposium on Applied computing*. http://doi.org/10.1145/1363686.1363711

Paré, G., Trudel, M., Jaana, M., & Kitsiou, S. (2014). Information & Management Synthesizing information systems knowledge : A typology of literature reviews. http://doi.org/10.1016/j.im.2014.08.008

Pesic, M., & van der Aalst, W. M. P. (2006). A Declarative Approach for Flexible Business Processes Management. *Business Process Management Workshops*, 169 – 180. http://doi.org/10.1007/11837862_18

Polya, G. (1945). How to Solve It: A New Aspect of Mathematical Method. *American Mathematical Monthly*. http://doi.org/10.2307/2306109

Reijers, H., Slaats, T., & Stahl, C. (2013). Declarative modeling - An academic dream or the future for BPM? *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8094 LNCS*, 307–322. http://doi.org/10.1007/978-3-642-40176-3_26

Reijers, H., & van der Aalst, W. M. P. (2005). The effectiveness of workflow management systems: Predictions and lessons learned. *International Journal of Information Management*, *25*(5), 458–472. http://doi.org/10.1016/j.ijinfomgt.2005.06.008

Van de Weerd, I., & Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. In *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (pp. 35–54). http://doi.org/10.4018/978-1-59904-887-1

Van der Aalst, W. M. P. (1998). the Application of Petri Nets To Workflow Management. *Journal of Circuits, Systems and Computers*, *08*(01), 21–66. http://doi.org/10.1142/S0218126698000043

Van der Aalst, W. M. P., & Berens, P. J. S. (2001). Beyond Workflow Management: Product-Driven Case Handling. In *Proceedings of the 2001 International ACM SIGGROUP*. http://doi.org/10.1145/500286.500296

Van der Aalst, W. M. P., ter Hofstede, A. H. M., & Weske, M. (2003). Business Process Management: A Survey. *Lecture Notes in Computer Science, BPM 2003*.

Van der Aalst, W. M. P., Weske, M., & Grünbauer, D. (2005). Case handling: A new paradigm for business process support. *Data and Knowledge Engineering*, *53*, 129–162. http://doi.org/10.1016/j.datak.2004.07.003

Van der Werf, J. M. E. M. (2011). Compositional design and verification of component-based information systems. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Compositional+Design+and+Verification+of+Component-Based+Information+Systems#0

Van Hee, K., Keiren, J., & Post, R. (2008). Designing case handling systems. *Transactions on Petri Nets and Other Models of Concurrency*, 119–133. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-89287-8_8

Van Hee, K., Oanea, O., Post, R., Somers, L., & van der Werf, J. M. E. M. (2006). Yasper: A tool for workflow modeling and analysis. In *Proceedings - International Conference on Application of Concurrency to System Design, ACSD* (pp. 279–281). http://doi.org/10.1109/ACSD.2006.37

Van Klink, P. J., & Hartelman, G. (2013). Concept & Casus Zaakgericht Werken. *NVWA*, 1–35.

White, S. A. (2004). Introduction to BPMN. *BPTrends*, 1–11. http://doi.org/10.3727/000000006783982421

Yin, R. K. (2009). *Case Study Research: Design and Methods*. *Essential guide to qualitative methods in organizational research* (Vol. 5). http://doi.org/10.1097/FCH.0b013e31822dda9e

Appendix A

Case-handling Systems in municipalities

Case-handling System	Company	# of municipalities
Decos D5	*Decos*	65
Verseon E-Suite	*Circle Software*	44
Atos e-Suite	*Atos*	29
Green Valley Zaaksysteem	*Green Valley*	23
CiVision Zaakafhandeling	*PinkRoccade*	20
Exxellence ZaakSysteem	*Exxellence*	20
Centric Zaaksysteem	*Centric*	16
InProces	*Brein*	15
Mozard Suite	*Mozard*	15
Zaaksysteem.nl	*Mintlab*	12
Liber Zaaksysteem	*BCT*	11
Corsa	*BCT*	10
WebNext	*PinkRoccade*	9
PerfectView KCC-Zaaksysteem	*PerfectView*	8
Mozaiek	*Drechtsteden*	6
IZIS	*Roxit*	3
KIM	*Kodision*	2
Nloket Zaak	*COSA*	2
SinglePoint-for-all	*Splitvision*	2
Unknown		*18*
None		*65*
Case-handling System		*312*
No Case-handling System		*83*

Table 3. Distribution of Case-handling Systems across municipalities (Zaaksystemeninbeeld.nl)

Appendix B

Interview data

#	Respondent name	Organization	Role	Date	Duration of interview
1	Peter Klink	NVWA	Blik op 2017 team	16-03-2015	90 minutes
2	Sven Blom	KBenP	Case-handling Consultant	01-04-2015	70 minutes
3	Frank de Keijzer	Municipality of Bunnik	Information Coordinator	28-04-2015	50 minutes
4	Marco Klerks	Municipality of 's Hertogenbosch	Information Architect	01-05-2015	50 minutes
5	Sebastiaan Gerats	Muncipality of Molenwaard	Team Leader Answers & Communications	08-05-2015	70 minutes
6	Harry Post	Province of Noord-Brabant	Advisor Information Management	20-05-2015	90 minutes
7	Marc Hählen	NVWA	Blik op 2017 team	02-06-2015	60 minutes

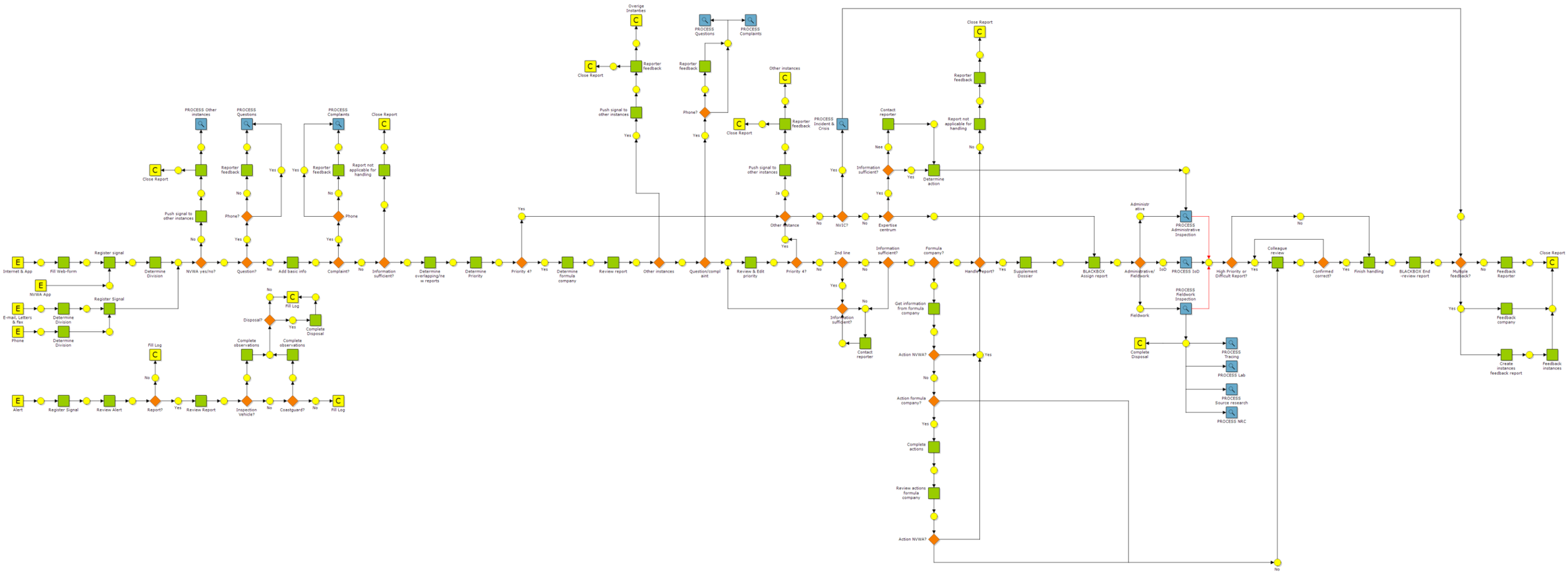
Table 4. Overview of interviewees' organizations, roles and duration of interview

Semi-structured interview questions

1. When did the organization start working with a Case-handling System?
2. How long did the implementation of said system take?
3. How many employees work with the system?
4. What was the trigger to change to a Case-handling System?
5. What were the steps that the organization took during the implementation of the system?
6. What went well, what went wrong during the implementation? What would the organization change if it would start the implementation with the knowledge from the previous implantation?
7. How were the processes modeled and transformed to case types?
8. What were specific KPI's to model the case types?
9. How many case types have been defined?
10. How many cases are handled each year?
11. How do sub-cases influence the complexity of a case?
12. How are the underlying rules of a process defined? How are these rules implemented in the system?
13. How does the organization react to changes in legislation and/or processes?
14. How was the switch to the new system communicated to the employees? How did the employees react to the switch?
15. Have there been improvements in services after the implementation of the Case-handling System?

Appendix C

Reporting process at NVWA – Remodeled in Petri Net.



Reporting process at NVWA – Colored module overlay