Decision Mining:

Discovery of decisions using decision event logs

Matthijs Berkhout

Master thesis Business Informatics

Decision Mining:
Discovery of decisions using decision event logs

Matthijs Berkhout
Department of Information and Computer Sciences
Utrecht University
2020

Supervisors:
Hajo Reijers
Jan Martijn van der Werf
Martijn Zoet

Acknowledgements

I would like to thank a number of people who helped me during the project. I would like to thank my second supervisor Jan Martijn van der Werf for the useful feedback on this project and for better scoping it. I would also like to thank my first supervisor Hajo Reijers for providing the guidance during the different phases and asking the right questions during our meetings. I also want to thank Martijn Zoet as my daily supervisor for his support and time, and keeping me motivated when there was a setback. The provided resources were indispensabile for testing the algorithm. I would like to thank the HU University of Applied Science, and especially the research group Digital Smart Services for creating the right environment for writing this thesis. I want to thank Robin Kuijt for the support with the programming questions I had and helping adapting the algorithm so it can read and analyze the decision event logs. I would like to thank Sam Leewis for the many discussions we had on decision mining, which helps defining this new field further. I would also like to thank Koen Smit. His support through the years helped me getting where I am now. I would also like to thank my colleagues at the HU University of Applied Sciences Utrecht. Finally, I would like to thank Astrid van Barneveld for proofreading this thesis, correcting typing mistakes and improving the readability of this thesis.

Table of Contents

Figures

Tables

1. Introduction

Information Technology (IT) has been in motion since its conception. The evolution of IT architectures from the 1960s until the 1990s shows that the separation of different functions has become a best practice in IT. This separation is the underlying principle of IT-architectures (van der Aalst, 1998; Zoet, 2014). It began with the separation of data from the application using a database management system (DBMS). Next, in the 1980s, User Interface Management Systems (UIMS) were separated. In the 1990s, the evolution continued with the separation of functions into Workflow Management Systems (WFMS) as seen in figure 1 (van der Aalst, 1998). The core functionality of a WFMS can be described as the ability to support an operational business process, e.g., automating the flow of decisions (Netjes, Reijers, & van der Aalst, 2006).

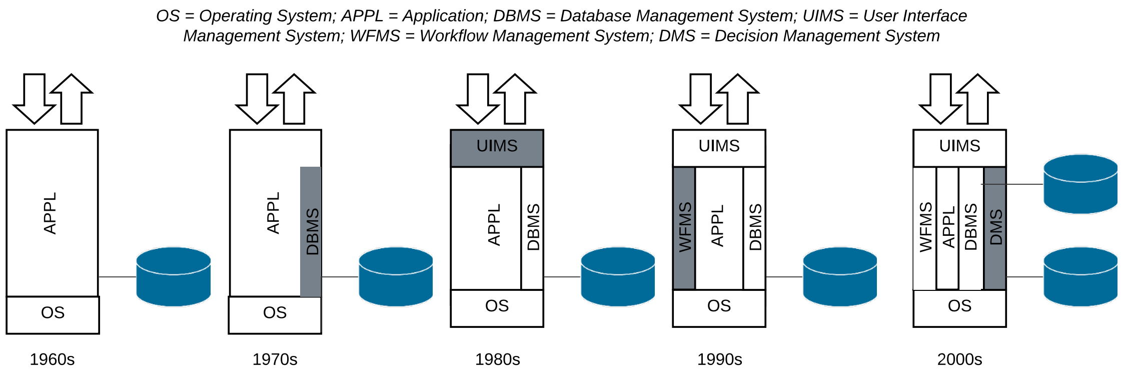

Figure 1: Separation of functions within IT architectures through the years (adapted from van der Aalst, 1998; Zoet, 2014)

Nowadays, WFMSs have evolved into Business Process Management Systems (BPMSs) (Netjes et al., 2006). A BPMS provides a broad range of facilities for designing, enacting, controlling and analyzing business processes. This is called Business Process Management (BPM). In 2003, van der Aalst et al. (2003) defined BPM as *"supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information"* (van der Aalst, ter Hofstede, Weijters, & Weske, 2003). To support the different phases of the BPM lifecycle shown in figure 2, a BPMS is used. The BPM lifecycle consists of four phases: process design, system configuration, process enactment, and diagnosis. While the lifecycle suggests a closed loop, as visible in figure 2, in reality, the execution of processes supported by BPMSs is disconnected from the design of the process (van der Aalst, Netjes, & Reijers, 2007).

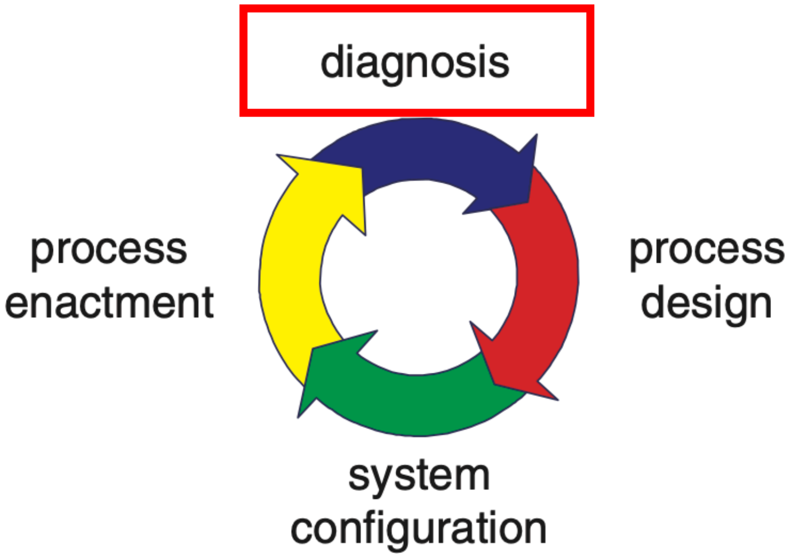

Figure 2: BPM lifecycle adapted from van der Aalst et al. (2003)

To fully close the BPM lifecycle, it is necessary to automatically interpret information stored in event logs exported from the BPMSs and use this to discover, check, and enhance models that describe what actually happens in a business process, as opposed to a theoretical model. In 2004, Van der Aalst (2004) proposed a solution to close the BPM lifecycle using process mining techniques (Van Der Aalst, Reijers, & Song, 2005; Van der Aalst & Weijters, 2004; Van Der Aalst, Weijters, & Maruster, 2004). Process mining is defined as: *"the discovery, monitoring and improvement of real processes by extracting knowledge from event logs readily available in today's information systems''* (van der Aalst, 2011a).

With process mining, organizations can identify trends, patterns and details that are recorded in event logs by a software system such as a BPMS or an Enterprise Resource Planning System. It also aims to improve the efficiency and understanding of processes (van der Aalst & Weijters, 2005).

Along with the rise of process mining, the evolution of IT architectures continued with the separation of functions into Decision Management Systems (DMSs), shown in figure 1. These systems support the management of decisions. A decision is *"a conclusion that a business arrives at through business logic and which the business is interested in managing"* (Object Management Group, 2016). In this definition, business logic is defined as *"a collection of business rules, business decision tables, or executable analytic models to make individual business decisions"* (Object Management Group, 2016).

Decisions are made within different contexts and within different systems. Examples of decisions that are executed in a DMS are "Determine Risk-Free Years" and "Determine Mortgage Eligibility". The first decision determines the number of risk-free years a car owner has accumulated in order to determine the premium of his/her car insurance. The second decision determines if a customer is eligible for acquiring a mortgage from a bank. These kinds of decisions are "black and white" decisions:

decisions where the input and output are known, as well as the theoretical steps taken to reach this result. The opposite is true with opaque decisions. These are black box decisions: the input and output is known, but the decision making process itself is not retrievable (Holzinger, Langs, Denk, Zatloukal, & Müller, 2019). An example of an opaque decision is a decision made during a game of Go by AlphaGo (Silver et al., 2017). While many studies are being conducted on opaque decisions made by e.g. neural networks, deep learning, and reinforcement learning, and how to make these self-explanatory (De Fauw et al., 2018; Silver et al., 2017), no studies are being conducted, at the time of writing of this thesis, on discovering "black & white" decisions in DMSs.

Only "black and white" decisions are managed using a DMS that aims to support the decision management lifecycle (DM lifecycle) (Smit & Zoet, 2018). An example of this is shown in figure 3. The different phases of this lifecycle will be elaborated upon in chapter 3.2.

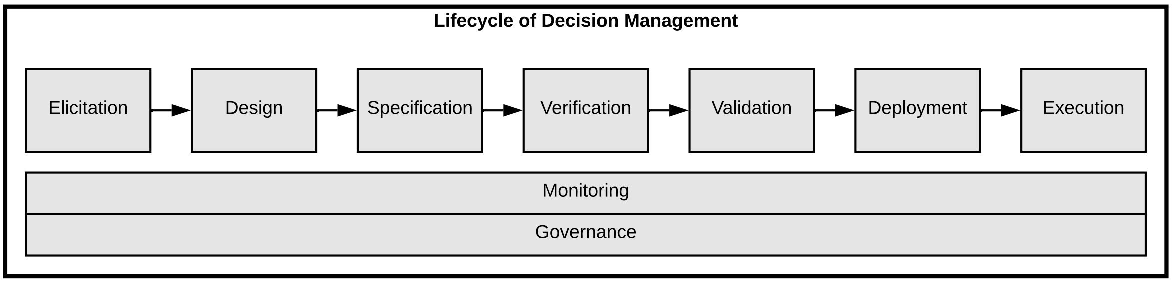

Figure 3: Decision management lifecycle. Adapted from (Smit & Zoet, 2018)

Similar to the BPM lifecycle, the DM lifecycle is not fully closed yet. This means that, as with process management, it is necessary to automatically interpret information stored in decision event logs (see below) that are exported from the DMSs. These decision event logs can be used to discover, check and improve models describing what actually happens in decision models and the underlying business logic and so fully close the DM lifecycle.

A problem that arises when using process mining for the discovery of decisions is that the output of a BPMS is different than that of a DMS. The output of a BPMS is a process event log, which consists of a minimum of a CaseID, an activity, and a timestamp (shown in figure 4).

CaseID	Activity (course name)	Timestamp (exam date)	Extra data (grade)
1	Software Architecture	16-1-2019	8
2	Software Architecture	16-1-2019	7
3	Software Architecture	16-1-2019	6
4	Software Architecture	16-1-2019	9
2	BPM	17-1-2019	7
3	BPM	17-1-2019	8
2	Process Mining	20-1-2019	6
3	Process Mining	20-1-2019	9
4	Process Mining	20-1-2019	7

Figure 4: Example of a process event log

The output of a DMS is a decision event log and consists of a minimum of a CaseID, one or more conditions and a conclusion, as shown in figure 5. Decision event logs can come from a wide variety of sources such as database systems, business suites (i.e. Oracle, SAP), Decision Support Systems, DMSs, and an API which provides data from websites.

CaseID	Condition (age)	Condition (health issue)	Conclusion (eligibility)
1	18	Yes	Not Eligible
2	25	No	Eligible
3	40	Yes	Not Eligible
4	16	Yes	Not Eligible
5	12	No	Not Eligible

Figure 5: Example of a decision event log

As traditional process mining focuses on sequential patterns and a decision event log is not sequential, traditional process mining techniques cannot be directly used on decision event logs. The main difference is that a timestamp is obligatory to find the sequences, while this is not obligatory for decision event logs. This creates a need for other algorithms to enable decision discovery in this type of event log. This difference was recognized by De Smedt et al. (2017), who proposed a new type of mining, called decision mining, to address this problem. They said: “First of all, it is a challenge for researchers to further develop a decision mining approach that is driven by the construction of a decision model, rather than by the control flow containing decision points.” (De Smedt et al., 2017, p. 203). This type of mining can be used to extract information from decision event logs and create a decision model from this. Therefore, in this thesis, decision mining is defined as *“a method of extracting and analyzing decision event logs with the aim to extract information from such decision event logs for the creation of decisions, to check compliance to decisions and regulations, and to present performance information for improvement”.*

The goal for this thesis is to identify or design an algorithm with the potential to discover decision models and the underlying business logic, using a decision event log, to close the DM lifecycle. What process mining does for the discovery of processes (van der Aalst, 2011), this study aims to do for the discovery of decisions with decision mining using an algorithm.

2. Research question

As stated in the introduction, the goal of this research is to find an algorithm for the discovery of decisions within decision event logs. Based on the gaps in research, the following research question is formulated:

How can a decision discovery algorithm be designed so that a decision requirement diagram can be extracted?

The research objectives pursued in order to answer the research question are:

RO1: *Give an overview of the current body of knowledge for decision mining.*

The goal of this research objective is to show the current body of knowledge for decision mining. It focuses on decision mining, but will also show the different types of decision mining.

RO2: *Give an overview of relevant algorithms (that are used in process mining) for decision mining.*

The goal of this research objective is to find algorithms in the context of process mining that could be useful for decision mining.

RO3: *Adapt and test algorithm for decision discovery.*

The goal of this research objective is to design or adapt the identified algorithm for decision mining and test it using different datasets. The algorithm will be explained in order to achieve a clear understanding of how the algorithm works.

RO4: *Validate the created algorithm.*

The goal of this research objective is to validate the algorithm. This is done by using datasets with already known answers for comparison.

3. Theoretical background

In this section, related literature to process mining and decision management are explained to come to a better understanding of the topic.

3.1. Process model

To understand how a decision works, it is important to first establish the basics of where its foundations lie and what a process consists of. A process is defined as "*a collection of inter-related events, activities and decision points that involve a number of actors and objects, and that collectively lead to an outcome that is of value to at least one customer"* (Dumas, La Rosa, Mendling, & Reijers, 2018).

To document, specify and analyze processes, organizations use process models (Van Der Aalst & Weijters, 2005). Process models specify the theoretical order of activities (see figure 6) and are commonly visualized according to the Business Process Model and Notation standard (BPMN; a notation also applied throughout this thesis).

In many cases, the execution of a process requires decisions to be made between multiple alternatives (Mannhardt, de Leoni, Reijers, & van der Aalst, 2016b). In BPMN, these decisions are modelled as decision points or gateways. Figure 6 is an example of a simple process model depicting the steps taken when someone orders medicine online. The customer *places an order*, the *order is processed* by the organization and the *order is shipped* to the customer.

Figure 6: Example of a simple process model

If the government were to decide that minors should not be able to buy medicine online, the company would be forced to *check the age* of a patient before they *accept the order*. To comply with the new legislature, our fictional company decides to create new tasks and a simple gateway is built in (figure 7). The gateway has two different paths. If the customer's *age is 18 years or above*, the process will continue, while the *order will be declined* if the customer's *age is below 18 years*.

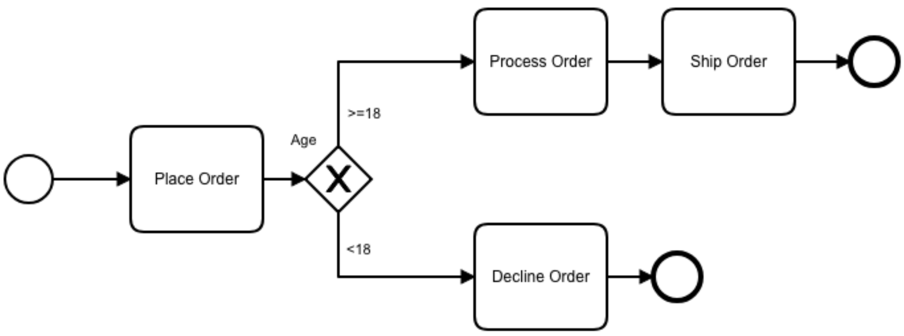

Figure 7: Example of a process model with simple gateway

A few months later, the government decides that an *age check* alone is not enough and the customer must comply with another condition before the online order is processed: the customer may not suffer a serious health issue. If the patient does not suffer a serious health issue, the order can be processed, but if there is a serious health issue, the order must be declined. This leads to the creation of cascaded gateways, as modelled in figure 8.

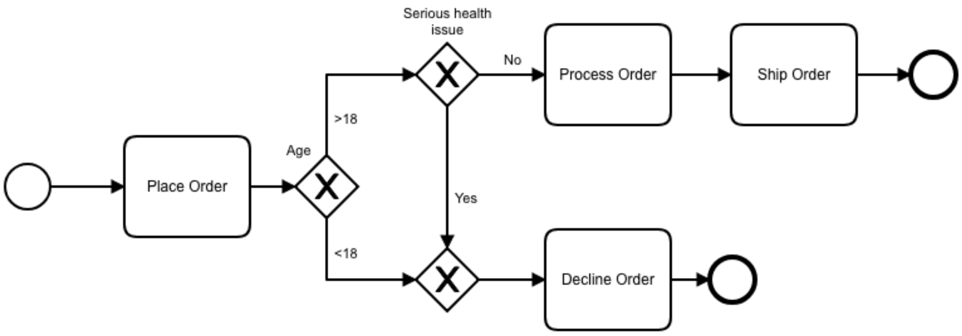

Figure 8: Example of a process model with cascaded gateways

While this is a relatively easy example of a cascaded gateway, it can already be used to demonstrate some of the issues that arise when decisions are modelled within process models in this manner (Zoet, 2014). For example, one question that arises is: what counts as a serious health issue? The government is likely to have a list of these issues, or which conditions apply, but modelling that into this process model would be impractical, if not impossible. This problem can be solved by modelling decisions separately from the processes that contain them. A common way to do this is by using the Decision Modelling & Notation Standard, or DMN. (Object Management Group, 2016). DMN is described in more detail in section 3.3.

3.2. Decision management

The argument to manage decisions explicitly and separately from the processes or information systems that contain them is in line with the separation of concerns as described in (Boyer & Mili, 2011; Nelson, Rariden, & Sen, 2008; Zoet, 2014). This principle advocates the creation of separate sections or modules that enable the management or editing of one concern (in this case, a decision) without altering the other (in this case, the various process steps).

For the business process shown in figure 8, the concerns that can be separated from the process itself are the business decisions and the underlying business logic. A business decision is defined as *"a conclusion that a business arrives at through business logic and which the business is interested in managing"* (Object Management Group, 2016). Business logic is defined as *"a collection of business rules, business decision tables, or executable analytic models to make individual business decisions"* (Object Management Group, 2016).

Decision making is often closely related to business processes, but decisions within a process may change faster than the process itself. For example, the business decisions for receiving a student loan may change twice a year while the business process stays the same. Managing processes and decisions separately results in compliant and adaptable organizations (Zoet, Versendaal, Ravesteyn, & Welke, 2011). To manage these decisions the decision management lifecycle is designed (Smit & Zoet, 2018). This lifecycle consists of 9 phases (see figure 9).

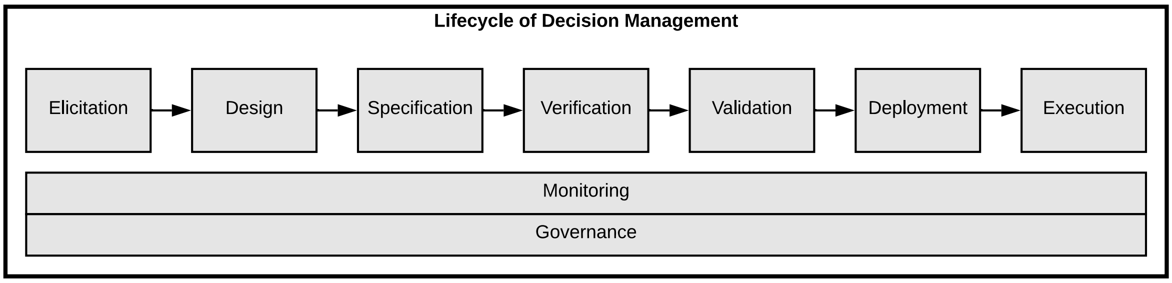

Figure 9 DM lifecycle (previously shown in figure 3)

The first step in the lifecycle is elicitation, where decisions are extracted from their sources, e.g. laws or internal documents. The phase is to design a decision model, for making the relations between the decisions explicit. An example of designing a decision model is creating a decision requirement diagram. The next phase is specification. This is where the business rules for the decisions are made. After the design and specification of the decisions and the underlying business logic the verification and validation of the decisions and business rules takes place. With verification the business logic is compared to predefined criteria such as language guidelines or syntax errors. Validation checks if the decisions and business logic have the intended outcome. After the verification and validation, the decisions and business logic are deployed into a DMS so that they can be executed. Monitoring and governance are done during the whole lifecycle, where monitoring uses KPI's to validate if goals are matched. Governance consists of the traceability and version management within the lifecycle (Smit & Zoet, 2018; Zoet, 2014).

3.3. Decision Model and Notation

One of the latest tools for decision management, the Decision Model and Notation standard, was introduced in 2015 by the Object Management Group (OMG). DMN provides both a method and a corresponding notation to model decisions separately from processes (see table 1). An updated version of the DMN specification was published in June 2016 (Object Management Group, 2016).

The DMN standard acknowledges two levels of abstraction for decisions. The first is the decision requirements level, which is captured in a decision requirements diagram (DRD). A DRD identifies relations and derivations between decisions. It can be used to identify decisions, input data, business knowledge needed to make the decision, and the knowledge source that denotes the authority for the business logic. Business logic is the second level acknowledged by the standard. On this level, the business rules applied to execute a decision are specified. It covers decision tables, S-FEEL, FEEL, JAVA, Python and other programming languages (Object Management Group, 2016; Smit, Zoet, & Berkhout, 2016).

Adopting DMN modelling enables organizations to cover these two abstraction levels, but it also creates possibilities to improve communication between users, validation of decisions, and the automation of decisions (Object Management Group, 2016).

A summary of modeling elements used within DMN is presented in table 1.

Table 1: DMN elements (taken from Smit et al., 2016 with permission)

Element	**Notation**	**Description**
Decision	Decision	A decision denotes the act of determining an output from a number of inputs, using decision logic which may reference one or more business knowledge models.
Input data	Input data	An input data element denotes information used as an input by one or more decisions. When enclosed within a knowledge model, it denotes the parameters to the knowledge model.
Knowledge source	Knowledge source	A knowledge source denotes an authority for a business knowledge model or decision.
Business knowledge	Business knowledge	A business knowledge model denotes a function encapsulating business knowledge, e.g., as business rules, a decision table, or an analytic model.
Knowledge requirement		A knowledge requirement denotes the invocation of a business knowledge model.
Authority requirement		An authority requirement denotes the dependence of a DRD element on another DRD element that acts as a source of guidance or knowledge.
Information requirement		An information requirement denotes input data, or a decision output being used as one of the inputs of a decision.

To gain a better understanding of the separate modelling of decisions using DMN, the example of online medicine sales described in the previous section will be modelled again as a process, this time with decisions modelled separately using DMN. Figure 10 shows the process with the new, legally required *determine eligibility* process step.

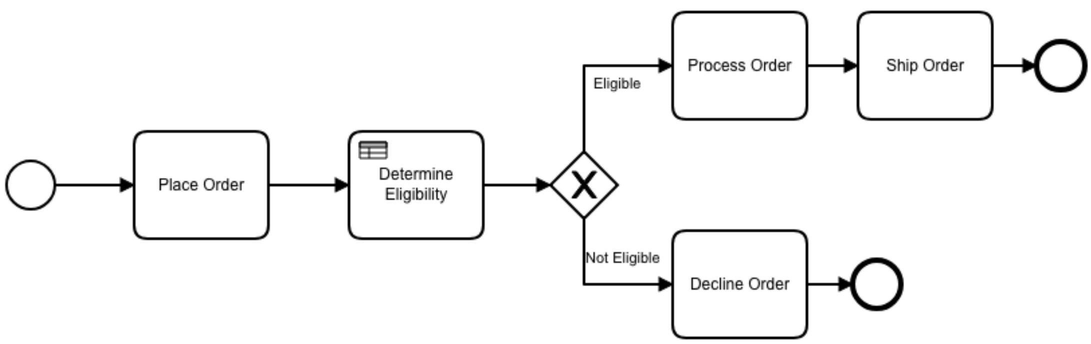

Figure 10: Process model with DMN process step

In figure 11, the *determine eligibility* activity is modelled as a decision requirement diagram using DMN, with a *determine eligibility* decision. In line with the stipulations of the government, this decision has two input values: age and health issue.

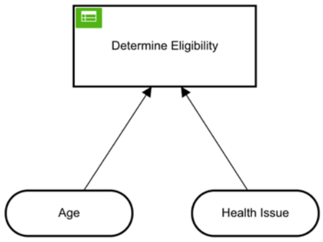

Figure 11: Decision Requirement Diagram

The decision *determine eligibility* is made using a decision table (figure 12). The table has two input variables with different conditions for each input. This decision table visualizes how a customer can only order a product if they are 18 years or older and without serious health issues. Otherwise, the order will be declined.

Determine Eligibility

Decision_Eligbility_v1

View DRD

U	Input +		Output +	
	Age	Health Issue	-	
	string	string	string	Annotation
1	<18	-	Not Eligble	If Age is below 18, order is declined
2	>18	True	Not Eligble	If Age is above 18, but health issue is true, order is declined
3	>18	False	Eligble	Ife Age is above 18 and no health issue, order is approved
+	-	-	-	-

Figure 12: Decision table

DMN was created for several reasons (Object Management Group, 2016). First, the OMG felt that operational decisions should be carefully modeled because they are a subject of attention during, for example, compliance checks. Also, providing a common visual language, like BPMN for business processes, would allow a better understanding of decision making and a common understanding between decision stakeholders. Lastly, a visual decision notation would serve as a tool for the identification of issues, thereby providing an opportunity for decision discovery and improvement (Smit et al., 2016).

3.4. Process mining

An increasing number of companies now create process models, often still on paper (Dumas et al., 2018). But are theoretical models an accurate representation of the actual processes that are executed in practice, or do unnoticed discrepancies exist? To detect these potential discrepancies, or even 'hidden' process steps, a technique called process mining is used (Van Der Aalst, 2011).

Process mining is defined as *"the discovery, monitoring and improvement of real processes by extracting knowledge from event logs readily available in today's information systems"* (van der Aalst, 2011a). This technique is frequently used for business process management (BPM) (Dumas et al., 2018; Weske, 2012). Process mining is both a technique for the discovery of processes as well as an analysis technique for mining data and getting useful analytics. Therefore, it can be subdivided into three capabilities (van der Aalst, 2011): 1) process discovery, 2) process conformance, and 3) process enhancement. These capabilities are shown in figure 13.

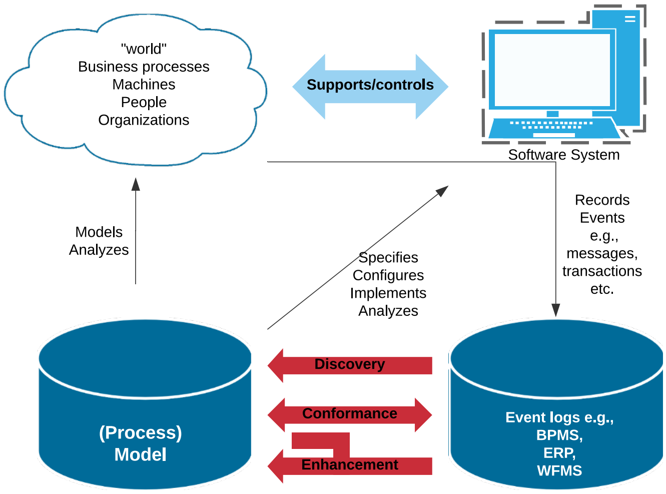

Figure 13: Positioning of the three types of process mining (van der Aalst, 2011)

The first capability, process discovery, constructs a representation of an organization's business main process and variations. Event logs are used as input to set up the process models (van der Aalst, 2011) and serve as the starting point of process mining (van der Aalst et al., 2012). Every information system used by an organization stores detailed information in event logs about the sequence of activities performed during the execution of a process (Dumas et al., 2018). The events in the event logs each symbolize one activity within a business process. In event logs, detailed information on all events is stored, such as the (total) duration of an activity.

The second capability is process conformance. During conformance checking, the discovered and created process model is analyzed and checked on any difference between the event logs and the (theoretical) process model (Rozinat & van der Aalst, 2008; van der Aalst, 2011a). The main purpose of conformance checking is to identify any problem areas that can be improved upon by using this knowledge.

The modification of process models to comply with the event logs can be done during the third capability: process improvement. This capability aims to extend or improve the process model using information from the event logs about the actual process (van der Aalst, 2011a). Two improvement types exist. Firstly, repair: the modification of the process model to better reflect the actual process (Rovani, Maggi, de Leoni, & van der Aalst, 2015). The second type, extension, means adding a new perspective after cross-referencing the model with the event log (van der Aalst, 2011a).

Process mining is based on four characteristics that are related to each other. The four perspectives are: case number (or ID), control-flow, organization, and time (van der Aalst, 2011a). The control-flow perspective focuses on the ordering of activities. The case perspective focuses on case properties, for example the characterization of a case by its process path. The organizational perspective focuses on resource information hidden in the event logs. For example, involved actors and their relations.

The time perspective focuses on the time and frequency of events e.g. showing total duration using timestamps.

3.5. Process discovery algorithms

As mentioned in the process mining section, process mining has three main algorithms for discovering processes from event data. While over 150 process mining algorithms exist in total, three lie at the root of all these algorithms and are used the most (Garcia et al., 2019). These are the heuristic miner, the multi-phase miner and the fuzzy miner (van der Aalst et al., 2012).

The heuristic miner was developed to address problems that occurred with the original alpha algorithm. One example of such a problem is duplicate tasks. The alpha algorithm cannot distinguish different tasks with the same label, while the heuristic miner can distinguish duplicate tasks (de Medeiros, van der Aalst, & Weijters, 2003). This makes the heuristic mining algorithm more suitable in practice. It derives XOR and AND connectors from dependent relations. Because it leaves out the edges in the data, it can abstract exceptional behavior and noise. This mining algorithm is suitable for real-life logs with a limited amount of different events. One example of possible output is a Petri net which can be used for further processing of the data, as seen in figure 14 (de Medeiros et al., 2003).

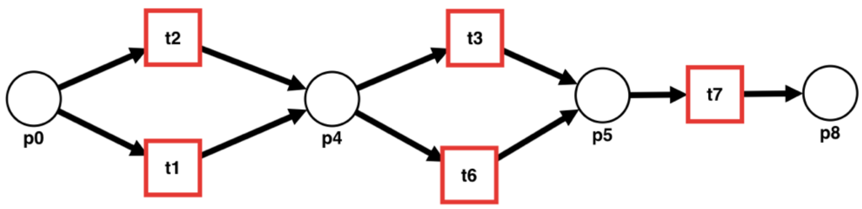

Figure 14: Example of a petri-net model

The multi-phase mining algorithm can express complex behavior within a relatively well-structured model. It folds “AND” and “OR” connectors and displays the resulting models as an Event-driven Process Chain (EPC), as seen in figure 15. The EPC can be used for further processing, such as when there is simple and structured log data and the mining results must be exported to an analysis tool. One of the main advantages of this type of mining is that it constructs a model that always shows the complete event log. On the other hand, this advantage makes multi-phase mining unsuitable for use in complex processes as the model becomes unreadable due to information overload (Van Dongen & van der Aalst, 2004).

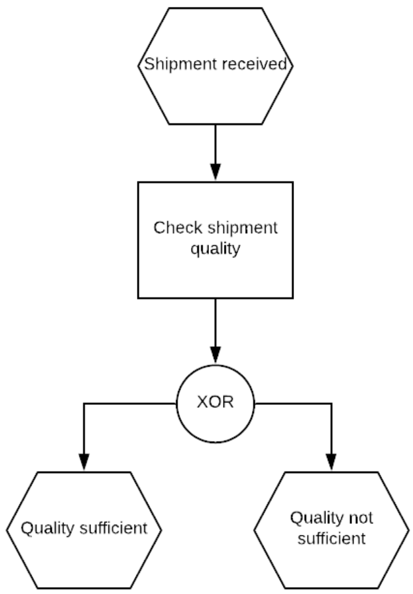

Figure 15: Example of an EPC

The fuzzy mining algorithm was created in 2007 by Günther & van der Aalst (2007). It was the first algorithm that could handle both a large number of activities as well as highly unstructured behavior. By using significance and correlation metrics it can simplify the process model to a higher granularity so that the output is readable. One of the advantages of the fuzzy mining algorithm over the heuristic mining algorithm is that it can hide less important activities in clusters or even completely leave them out of the model if necessary, e.g. when there are hundreds of activities. The fuzzy mining algorithm cannot be converted to other types of modelling languages, but the output of the algorithm can be projected on top of the created model, seen in figure 16. There, it can use the event log data to display dynamic process behavior to reveal bottlenecks (Günther & van der Aalst, 2007).

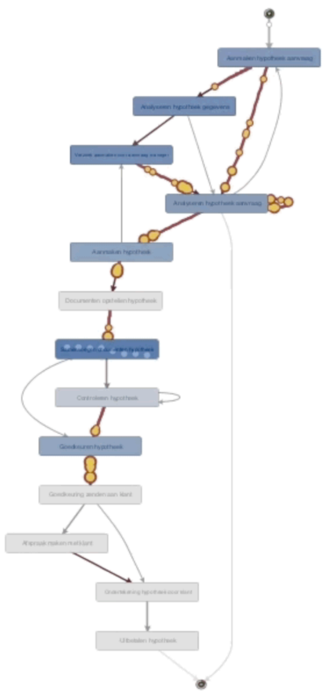

Figure 16: Example of a projection on top of a process model using fuzzy mining

4. Research method

In Information Science, Design Science is widely used for creating an artefact and validating such an artefact (March & Storey, 2008). There are different approaches to design science research, but in this case the design cycle from Wieringa for designing a new artefact is used (Wieringa, 2014), as seen in figure 17.

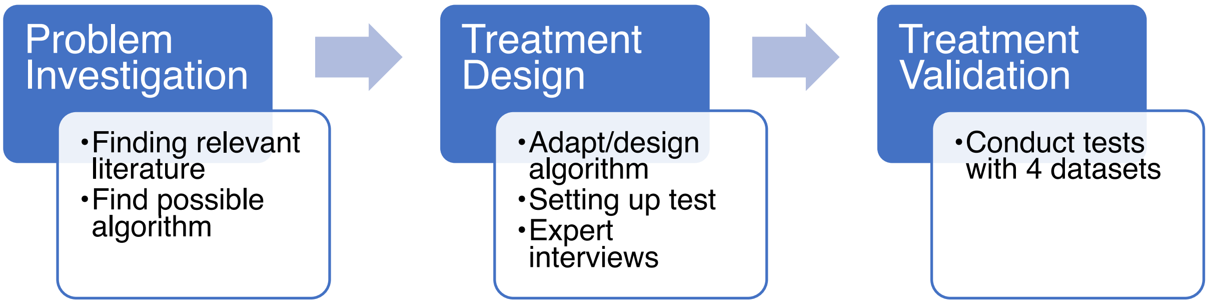

Figure 17: Adaption of the design cycle proposed by Wieringa (2014)

The first step of the engineering cycle is the problem investigation. As the research field on discovery of decisions is nascent, a literature review including a look at neighboring research fields was performed to define the topic, as exploration of neighboring fields may yield useful requirements for creating the artefact (Wieringa, 2014). The literature review is incorporated into the related works as it also consists related works to further define the area of study. One such relevant, and much more mature field, was process mining (Van Der Aalst & Weijters, 2005).

An existing algorithm found in Bazhenova & Weske (2016) was the input for creating a decision discovery algorithm. This existing algorithm was altered for decision discovery. The algorithm had the same purpose of finding decisions within data as this study, but it was focused on extracting decisions from event logs rather than discovering decisions in decision event logs. As the foundation of the algorithm is the same for both algorithms, the existing algorithm could be altered to also find decisions in decision event logs. Parallel to the altering of the algorithm, expert interviews were conducted to gain knowledge about potential algorithms and validate the found algorithm. Finally, the altered algorithm was tested against both synthetic and real datasets within the treatment validation phase.

The results of this study will motivate a new cycle, as is common in design science research (Wieringa, 2014). In this research, only one algorithm is created, tested and validated due to time restrictions within a master thesis.

4.1. Literature search protocol

The first step was to give an overview of the already available state-of-the-art literature surrounding decision mining. For that purpose, a search protocol was developed (see figure 16). The first step of this protocol was identifying the right databases. In this case Google Scholar was used as the main search database since it has a higher coverage compared to other available search engines, as well as compared to executing queries in individual databases (Amara & Landry, 2012; Franceschet, 2010; Harzing & Alakangas, 2016; Wildgaard, 2015). Within the databases, the search was limited to the Information Science field.

Information Science is an interdisciplinary field based on research from other disciplines (Webster & Watson, 2002). To identify all concepts of decision mining and its relations, a broad search range is necessary. For the literature review, the following search queries were executed, where a difference was made between the primary search query ("decision mining") and search queries providing context from other research fields. The (combined) executed queries are shown in table 2. The search query was executed in Google Scholar by using the following syntax:

"decision mining" AND ("process mining" OR "decision management" OR "Business Process Management" OR "Decision Support System")

Table 2: Search Queries

Search Query:	**Primary/Context:**
Decision mining	Primary
Process mining	Context
Decision management	Context
Business process management	Context
Decision support system	Context

The search query resulted in a list of 640 papers, but exclusion criteria were established, seen in figure 18, to narrow down the relevant literature for this state-of-the-art overview.

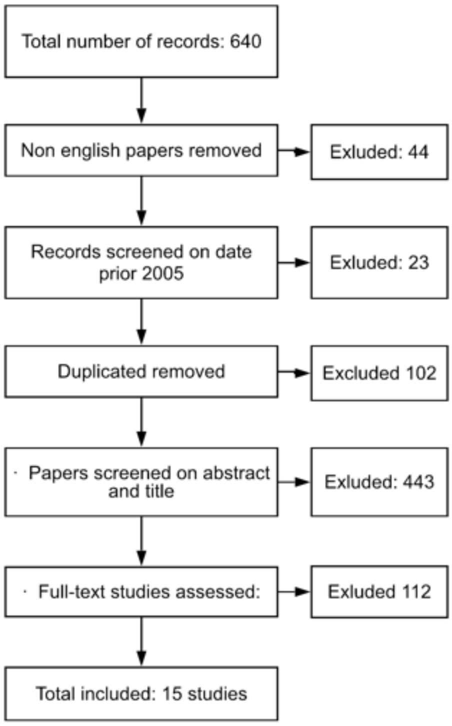

Figure 18: Selection procedure

First all non-English papers were excluded. Additionally, all papers prior to 2005 were excluded since decision mining was non-existent before that date. Next, all duplicates were removed. Papers were screened on relevance by assessment of title and abstract. Finally, remaining papers were assessed in full. This resulted in 15 papers eligible for the literature overview on decision mining. An overview of the selection process is shown in figure 18.

4.2. Expert interview

After the literature study, expert interviews were conducted. The goal for these interviews is twofold. On the one hand the interviews are conducted to gather knowledge from experts about the fields process mining and decision mining. On the other hand the interviews are conducted to gather knowledge on designing or adapting an algorithm. An expert interview is a qualitative research method designed to explore expert knowledge and is conducted with someone who has special knowledge on a certain topic ascribed by the researcher (Dexter, 1970; Meuser & Nagel, 2009).
Special knowledge is knowledge that pertains to the experts' specific professional field.
This study focused on experts in the field of decision mining and the adjacent field of process mining. The experts were chosen by the researcher in collaboration with his supervisors. This resulted in a list of key persons within the decision mining field. To include an expert for interviewing, the following criteria had to be met:

- The expert speaks English or Dutch;
- The expert must have published on the decision mining topic;

- The expert has knowledge on decision mining and/or process mining algorithms.

Interviews were semi-structured and performed according to an interview protocol with pre-determined questions as defined within Appendix A. Interviews were transcribed and analyzed using the qualitative analysis tool NVivo 12. Analysis of the interviews was done using thematic coding. This involves identifying and coding of text that is linked by a common theme. This ensures that the text is divided into categories and in the end a framework is established (Gibbs, 2007). In total, two rounds of coding were performed. In the first round, categories were defined. The second round of coding consisted of validating the found data in the created categories and validating if all information was categorized. All coding was done by the researcher. In total, 8 persons met the inclusion criteria. Five persons met the inclusion criteria to be eligible for this interview but were not available for an interview. This resulted in three interviews that were conducted. The interviews were conducted at the beginning of the designing of the algorithm.

4.3. Algorithm design

An algorithm will be adapted using already existing algorithms. The design consisted of three steps. The first step is designing requirements for the algorithm. The second step is the design of the algorithm. The last step is the actual creation of the algorithm. This will be further elaborated upon in section 6.

4.4. Algorithm validation

After the design of the algorithm, it has to be validated. An appropriate research method to evaluate the usefulness and applicability of a product, algorithm, method, framework or categorization, is an experiment based on 1) synthetic and 2) real-life datasets (van der Aalst et al., 2012). This is because experiments based on synthetic data allow the researchers to control 1) the model, 2) the input(s), 3) the experiment setup and the 4) actual simulation. Subsequently, when real-life data sets are applied, researchers are still able to control the model and experiment setup, while less control on the input and simulation can be asserted. Reproducibility and traceability are fundamental requirements for both synthetic and real-life-based experiments (van der Aalst et al., 2012).

To meet the reproducibility and traceability requirements, researchers have to report on different aspects per type of experiment. With respect to the experiment model, researchers have to report the aim of the experiment, the purpose of the model and the model outputs (Rahmandad & Sterman, 2012). The data sources, the input parameters, the pre-processing of the dataset and underlying assumptions have to be reported with respect to the inputs (Dalle, 2012).

For the experimentation setup the following elements have to be reported: 1) the base model overview, 2) model logic, 3) the scenario logic, 4) the algorithm, and 5) the components applied (Taylor et al., 2017). The reporting on the components mainly consist of the instruments (e.g. software or programming language), the system specification, and the sampling. Lastly, the following elements of the experiment execution have to be reported: 1) the initialization, 2) the run length, and 3) the estimation approaches (Dalle, 2012). Each of the previously described aspects will be specified for this specific study in the next paragraphs.

4.4.1. Test goals

As specified, the aim of the study was to evaluate the usefulness and applicability of the algorithms identified. The purpose of the model was therefore to determine the proper classification of each decision based on a predefined business rule model.

4.4.2. Datasets

Two synthetic datasets, a minimum viable synthetic dataset and a big synthetic dataset were created for this study. One synthetic business rule model to test both data sets and in addition, two real-life datasets with the accompanied business rules models were collected. Both synthetic datasets had the same purpose: assess the usefulness, completeness, and applicability of the decision discovery algorithm. The minimum viable synthetic dataset only tests whether the classification works correctly and was therefore created based on one theoretical criterion: the dataset had to contain one record per identified variation to assess each possible outcome (Rahmandad & Sterman, 2012). This resulted in a dataset with all possible variants possible. In addition, the expanded synthetic dataset extrapolates the minimum viable synthetic dataset and contains at least 5000 randomly created records. This is used to test if the algorithm does not break with large datasets. No input parameters were set for either synthetic dataset, nor had pre-processing occurred.

The real-life datasets were selected based on three theoretical criteria and one practical criterion. Regarding the theoretical criteria, the case had to provide: 1) one or multiple decisions, 2) underlying business logic e.g., decision table, 3) and the execution data exported as decision log. With regards to the practical criterion, the organization naturally had to be willing to provide context of the decision and the decision log needed to perform the experiment and test the algorithm. Based on these criteria two cases were selected. With regards to pre-processing of the dataset, the dataset was checked for incomplete decisions and null values by the daily supervisor together with the researcher. No incomplete decisions or null values were found in the data.

5. Decision Mining

While the need to model decisions separately from processes is clear (Object Management Group, 2016; Von Halle & Goldberg, 2009; Zoet, 2014), many organizations still model their decisions within processes as gateways. An initial attempt to mine decisions is done by Rozinat and van der Aalst (2006b). They developed a process mining algorithm to extract decisions from single gateways in process models and defined decision mining as follows: *"[an approach that] aims at the detection of data dependencies that affect the routing of a case"* (Rozinat & van der Aalst, 2006b).

This definition is focused on decision point analysis in process event logs (Batoulis, Meyer, Bazhenova, Decker, & Weske, 2015; Bazhenova, 2018; Janssens, Bazhenova, De Smedt, Vanthienen, & Denecker, 2016; Rozinat & van der Aalst, 2006a, 2006b). The approach, proposed by Rozinat et al (2006b) is comparable to the process mining method where event logs are mined for process discovery, conformance and process enhancement (van der Aalst, 2011b). The algorithm determines the specific points where a choice is made and which branches are followed. After the identification of a decision point, the authors try to determine if certain cases with certain properties follow each a specific route and visualize this into a petri-net. This approach has some limitations. For example, this approach cannot deal with more complex control-flow constructs and with event logs containing deviating behavior (De Leoni & van der Aalst, 2013). Other variations on this decision point analysis decision mining have been published through the years, for example Mannhardt et al.(2016a).

In 2016, Bazhenova created another algorithm which can draw a decision requirements diagram and a decision table from process event logs (Bazhenova, 2018; Bazhenova, Buelow, & Weske, 2016). This algorithm also focusses on decision point analysis, but extracting data directly from extra data included in process event logs. A limitation of this algorithm is that this type of decision logic is often implicitly contained in event logs and thus has to be processed before it can be used (Bazhenova & Weske, 2016). This algorithm uses fuzzy logic to create a decision table. A limitation of this approach is the dependency on experts for defining the boundaries for the fuzzy membership functions (Bazhenova, 2018). The decision requirement diagram is drawn using a decision tree algorithm for showing the dependencies between a decision and the used input data.

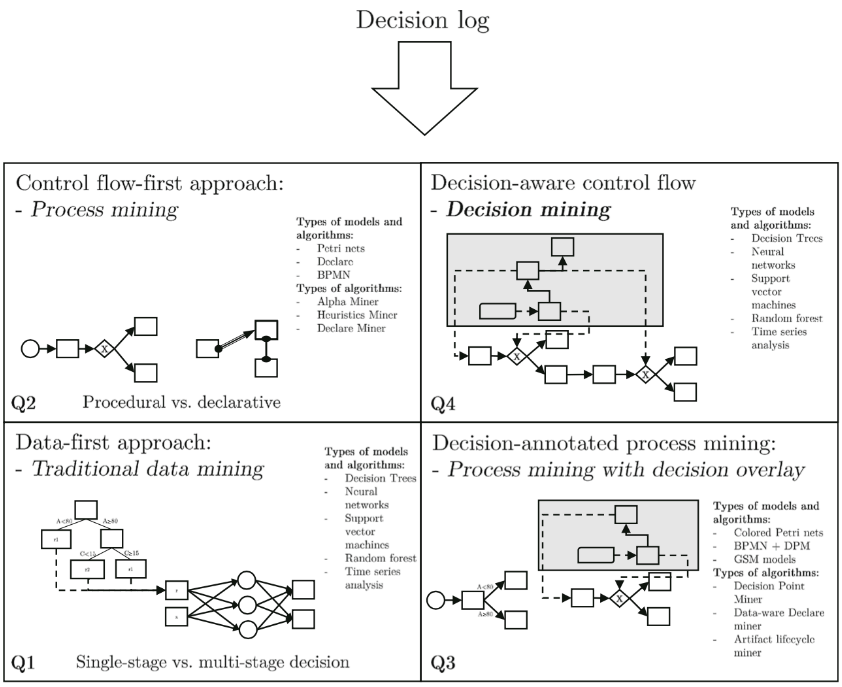

Figure 19: Decision mining quadrant (De Smedt et al., 2017)

De Smedt et al. (2017) recognised an upcoming field of managing decisions in a separate manner and the use of decision modelling for describing the relationships between decisions as decision management (De Smedt et al., 2017). This is seen in figure 19. Q1 is tradional data mining using a data first approach. Q2 is process mining which uses a control flow-first approach. Q3 is decision point analysis or decision annotated process mining. This is mining decisions from a single gateway and creating a decision model of it. Q4 is decision mining using a decision-aware control flow which is the type of decision mining used for this thesis.

At the time of writing the amount of literature on decision mining with decision event logs is scarce. Table 3 shows an overview of decision elements described in the existing literature. The decision elements are decision point analysis, decision requirement diagrams, business rules and decision tables. The decision elements are mapped by the method of extracting them: process event log-based according to the definition of Rozinat et al. (2006a), versus decision event log based according to decision-aware control flow from de Smedt at al. (2017). At the time of writing, no literature on decision event log based decision mining yet existed.

Table 3: Literature mapped on decision elements

Decision Element	**Process event log**	**Decision event log**
BPMN: Decision point analysis	(Rozinat & van der Aalst, 2006b)	*X*
Decision Requirement level: *Decision requirements diagram*	(Bazhenova, 2018)	*X*
Business Logic: *Business Rules*	(Bazhenova & Weske, 2016)	*X*
Business Logic: *Decision Tables*	(Bazhenova & Weske, 2016)	*X*

6. Interviews

After the literature exploration, interviews were conducted to validate the requirements of this type of algorithm. The interviews were also used to find pitfalls and tips for two things: 1) designing such an algorithm and 2) if there are other possible algorithms that could be used for decision mining. The interviews covered several topics.

6.1. Definition of decision mining

The first topic was the definition for decision mining. Interviewee 3 gave us the following definition of decision mining:

"There is little consistency in that terminology, but I think the decision mining method is understood as trying to extract the decision models or the model from data or event logs or decision logs or on some form of data."

Interviewee 1 on the other hand was following the same definition for decision mining as the definition of Rozinat et al.(2006b):

"the decision mining in the context of processes is that you mine not only the business rules but also of their different points in the process where the rule can appear so in particular in the form of decision models and then you have a dependency between decisions."

6.2. Algorithm choice: transparency

For the choice of algorithm, the experts did not agree with each other. Algorithms such as support vector machines or neural networks are very useful and could have accurate results, but as suggested by two interviewees, making something transparent using an algorithm that usually is a black box is a problem, as interviewees 2 and 3 stated:

(interviewee 2) "The main problems of many domain problems, that those algorithms have is that they are usually black boxes, they take input and they produce output. But it's not really clear how this output is produced from the inputs, because they are noise between the rules, If you take a neural network, and you cannot really understand which are the rules that this made the production of the output."
(interviewee 3) "Fuzzy mining or neural nets is just a form to mine the processes but if that makes a lot of difference in the decisions. I don't really think so."

However, interviewee 1 stated the following:

"So, I think maybe you can have a look in the direction of where he applies neural Networks, so he did a bunch of experiments. "
"For example, support vector machine. So the SVM and decision tree classification. I think that's for starters. "

The contradiction between interviewee 2 & 3 and 1 about transparency of the algorithm was taken into account during the search of useful algorithms. This was a requirement that a potential used algorithm must met. Making something transparent with an algorithm that is not transparent itself is not right. Algorithms based on neural nets or support vector machines were excluded. A disadvantage of not using neural nets or support vector machines for discovering rules is that other potentially useful algorithms are slower. As interviewee 2 stated, that other algorithms could be very slow and use a lot of RAM memory of a computer to extract and process the data:

> "The main problem there is that these algorithms can be quite slow, it can take a lot of time to discover these rules. And also, keeping in memory." {ed.: processing rules in computer RAM}

6.3. Difference between process mining and decision mining

The next topic was the difference between declarative and procedural mining. Process mining algorithms can be divided into two types: declarative mining algorithms and procedural mining algorithms. As decisions are declarative, and thus relational in nature, procedural mining algorithms are not adequate to use for the discovery of decision mining as interviewee 3 stated:

> "What may be important is that process mining is about process models and you undoubtedly know there are two types of declarative and procedural and the mining of declarative models could actually go hand in hand with the mining of decisions because they are both declarative things."

6.4. Datasets for validation

The last part was the use of real-life date for testing the algorithm. Interviewee 1 stated the following:

“And it's not always working well, because the data that you have available are not enough to fully determine what the rules are.”
“you're going to get another success factor of developing algorithms that you have test data which you can apply.”

As finding good real-life data is one of the success factors for developing useful algorithms this was a point of focus.

7. Algorithm Design

There are algorithms for discovering DRDs and business logic such as business rules and decision tables using process event logs, but no algorithms exist, to the knowledge of the researcher, for discovering these decision elements using a decision event log extracted from a DMS.

7.1. Algorithm requirements

The goal is to design or create an algorithm capable of the following requirements which are acquired from literature and the interviews:

Rq1: *The algorithm has to extract one or more decisions from decision event logs;*
Rq2: *The algorithm has to extract the business logic within the decision event log;*
Rq3: *The algorithm has to find relations between found decisions if they exist (e.g. a derivation structure);*
Rq4: *The algorithm has to create a visualization of a decision requirement diagram and the business logic used for the decision using DMN;*
Rq5: *The algorithm has to be transparent.*

First, the algorithms mentioned in the sections before were evaluated against these five requirements to show which algorithm has to most potential to adapt or that a new algorithm has to be created. Four possible useful algorithms were found in literature during the problem investigation. These algorithms are analyzed using the available literature and mapped against the requirements. Table 4 shows the mapping of the different algorithms against these requirements. X shows that the requirement is fully fulfilled and / shows that is it partly fulfilled.

Table 4: Mapping of algorithms against requirements

	Rq1	Rq2	Rq3	Rq4	Rq5
Rozinat et al.(2006a)	(X*)	(X*)			X
Mannhardt et al.(2016a)	(X*)	(X*)	X		X
Bazhenova et al.(2016)	(X*)	(X*)	X	X	/
C4.5 algorithm (2014)	X	X			X

*= Can extract decisions from a process event log

The table shows that the algorithm created by Bazhenova et al. (2016) already meets one of five requirements partly, two fully and two with an asterisk as it is used for process event logs instead of decision event logs. As this algorithm already fulfills the most requirements, it will be adapted to also fulfill the other requirements so that it also can read decision event logs and is transparent. In the next sections the mechanism of the algorithm will be described.

7.2. Algorithm design explanation

The algorithm created by Bazhenova et al. (2016) is adapted to read a decision event log. This is done by removing the part that extracts the decisions from process event logs and adapting that part so that the input of the algorithm is a decision event log. Also the discovering of decision rules is adapted so that both numbers and strings are read the same way. The Nefclassifier algorithm is removed and replaced by a descriptive decision tree, so that it only discovers data that is present and does not try to reproduce a complete decision table. The last part is that the two algorithms for finding dependencies from Bazhenova et al. (2016) are combined together so that is can identify dependencies and relations between decisions immediately. Below the adapted algorithm will be elaborated upon with an example.

The input for the algorithm is a decision event log. For this example, a decision event log is created to show how this algorithm works. An insurance company has special rules to handle fraud cases. In this case a decision called Alert is created for the employees. It has two input values, Fraud and Amount. Fraud is a Boolean and Amount is a numeric value. Both input values are used to determine if an alert is necessary or not. The decision event log output of this decision is shown in table 5.

Table 5: Decision event log sample

CaseID	Fraud	Amount	Alert
1001	True	500	True
1002	False	100000	False
1003	True	15000	False
1004	False	500	False
1005	True	500	True
1006	False	500	False
1007	False	500	False

The decision event log consists of four attributes: a CaseID, two conditions called Fraud and Amount, and a conclusion called Alert.

7.2.1. Business rule creation

The first step 1) for the algorithm is to read what type of data each attribute is. This is necessary as the original algorithm defines a difference between nominal and numeric values. The numeric values are put through a separate learner in the original algorithm. This has a limitation as expert knowledge is necessary to define threshold values for the numeric values. Therefore, the original algorithm is changed so that both numeric as nominal values are treated the same. 2) This is done by reading all values of the decision event log and if the value is numeric, it will explicitly be changed to a string, so the algorithm handles it as a nominal value.

The next step, 3) is the discovery of the rules within the decision event log. It starts by reading the header of the decision event log to identify the different attributes. 4) After that rules are created of every case. An example of what a rule looks like is the following:

IF Fraud = True AND Amount = 500 THEN Alert = True

The output is a total list of all business rules executed. These business rules are used for the creation of the decision table.

7.2.2. Decision table creation

Bazhenova et al. (2016) uses both a generic and a Nefclassifier algorithm which consist of a combination of neural nets and fuzzy mining to discover the decision tables. As a neural net is opaque and thus not transparent this could not be used. This algorithm also optimizes the data by predicting unique business rules instead of only discovering them. To create a decision table a transparent algorithm has to be used. The Nefclassifier is removed from the algorithm and replaced by a descriptive decision tree algorithm. This algorithm is transparent in contrast to the Nefclassifier. This part of the algorithm has the following steps: 1) It starts with the output of the rule, in this case Alert = True. 2) After that the input data is added as a node. In this case a node Fraud = True and a node Amount = 500. 3) This is repeated for every rule. If a new node is discovered this will be added to the conclusion. 4) When all rules are analyzed, a decision tree is created 5) which in turn is converted into a decision table. The decision tree created for this example is shown in figure 20.

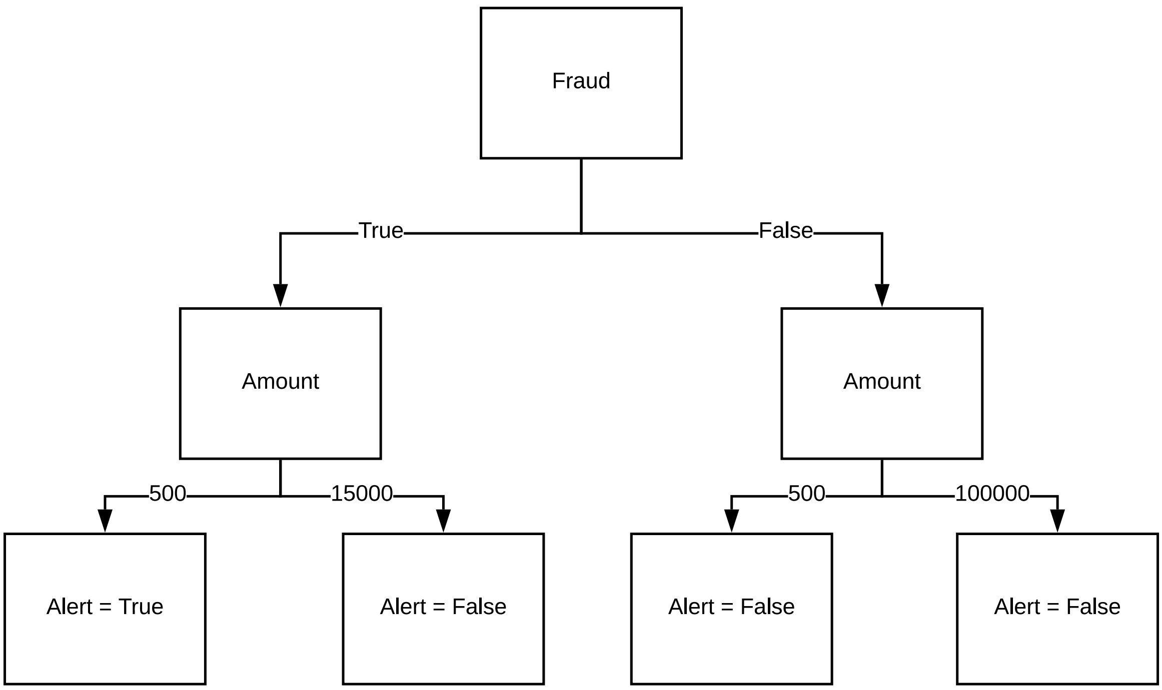

Figure 20: Decision tree of example dataset

A decision tree can be converted to a decision table, seen in table 6. According to the DMN standard, a decision table describes the relation between a set of input values and a set of output values (Object Management Group, 2016). Each decision table input has a domain over which logical expressions can specify conditions. During execution, each input is assigned a value, and if the association of logical expressions of all inputs checks out to true, the corresponding output values are provided. The possibilities of relating different inputs to outputs are represented in table 6 which concurs to each unique business rule.

Table 6: Decision table of example dataset

Fraud	Amount	Alert
True	500	True
True	15000	False
False	500	False
False	10000	False

A formal way to describe a decision table is the following (adapted from Bazhenova (2018)):

A *decision table dt* of the set of decision tables *DT* is a tuple $dt = (I, O, R, \chi)$, $dt \in DT$, where:

- $I = \{I_1,...,I_v\}$, $v \in \mathrm{N}^+$ is a finite non-empty set of input variables *(inputs)*;
- O is an output variable *(output)*;
- $R = \{R_1, ..., R_n\}$, $n \in \mathrm{N}^+$ is a finite non-empty set of mappings (*decision rules as seen in section 7.2.1*), which relate a subset of inputs to an output:

$$\forall i \in [1;n]\ R_i : \bigwedge_{j=1}^{w} (I_j\ op_j\ q_j) \longrightarrow Dom(O), 1 \leq w \leq v$$

- where $op_1, ..., op_w$ are comparison predicates, $q_1 \in Dom(I_1), ..., q_w \in Dom(I_w)$ are constants representing values from the domains of the inputs, and $j, w \in \mathrm{N}^+$;

- χ: $DT \longrightarrow \{$ *first, collect, unique* $\}$ is a function that assigns each decision table $dt \in DT$ a hit policy. This can differ from a unique hit policy to a first or collect policy.

A decision table is made up of columns where where each column represents one condition or one conclusion.
A condition is a predicate test that must evaluate to true for the associated action to be executed. If a decision table rule uses multiple conditions, all conditions for a row must evaluate to true for the action to execute depending on the function (χ). Condition tests might be looking to match the exact value of an incoming field's value, but can also compare an incoming value for whether it is less than, greater than, less than or equal to or greater than or equal to a specified value.

In each rule's row (*R*), the specific conditions (*I*) and actions (*O*) that comprise that rule are defined. For example, if a condition column is Amount (comparing to an incoming tuple field of that name), then each row can define a different amount or amount range. The action for each row would define what action to take if an incoming tuple contains an Amount value within the specified range.
If a second condition column called Fraud is added, then before the action is taken, the decision table operator (*op*) tests both conditions to see if both the incoming Amount and Fraud field values are within the ranges (*Dom*) specified in the rules. There is also the case of overlapping rules in a decision table. One or more conditions could be the same, but with a different output. To overcome this, the DMN standard has a function (χ) which assigns a hit policy to a decision table (*DT*). The hit policy can differ from a unique hit policy, where the decision table consists of only unique rows, to a first hit, where the first row that matches with the input data will show.

7.2.3. Decision requirement diagram

With the business logic discovered, the last step of the algorithm starts. Finding decision dependencies. In this case one decision, Alert, was analyzed, but most of the time multiple decisions are made. On top of that these decisions can be related, such that a decision can be a derivation of another decision. For example, the decision Alert is used for another decision, Finished. That means that the conclusion of Alert is used as input data for the decision Finished. While this could be solved by using another decision tree algorithm, a problem arises that a simple decision tree algorithm cannot solve. When a decision derivates from another decision, but also has separate input, a normal decision tree algorithm will not work. Our example does not only have the input data Alert (conclusion of this decision), but also new input data, Fine. Figure 21 shows the relation between the decisions and the corresponding input data. The red circle shows the problem which has to be solved. While the decision finished uses the output of the decision Alert, it also has its own input data. Bazhenova et al. (2016) has found a way to find these dependencies, but also found an improved way to find these dependencies in 2018 (Bazhenova, 2018). These algorithms are adapted by combining these two algorithms into one. That way it can find multiple dependencies between decisions and create a decision requirement diagram.

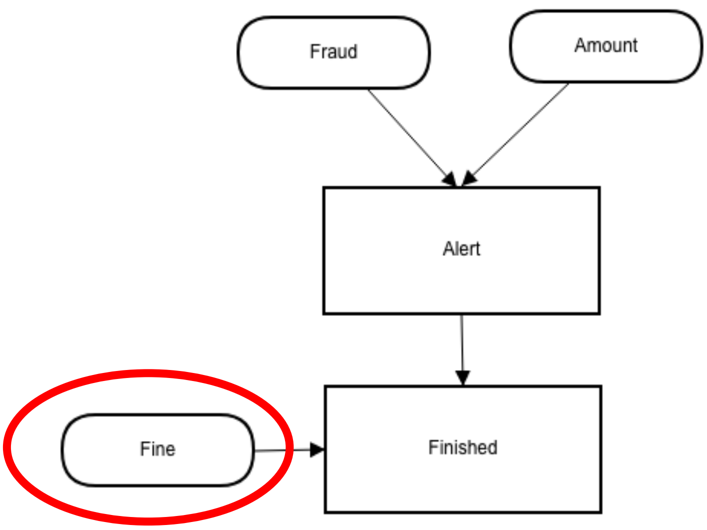

Figure 21: Decision dependencies with separate input data

1) To find these relations between decisions the algorithm will start again with all attributes (note: at the moment every decision is uploaded separately to the algorithm). In case of the example it will analyze six attributes again: Fraud, Amount, and Alert for the first decision and Fine, Alert, and Finished for the second decision.
2) It starts with the first decision Alert. If an attribute is used as condition for another attribute it will be labeled as input data. In this case Fraud and Amount are labeled as input data for Alert. 3) Then it recognizes the relation between Alert and Finished, as Alert is the conclusion for the first decision, but input data for the second decision. 4) When this happens, the algorithm will check if this is the only input data, or that there are other attributes used for the conclusion. In this case the attribute Fine is also a condition for Finished, so the algorithm will use that as input data. 5) To visualize the different relations the attributes are labeled as a decision or input data.

This part of the algorithm is also written in pseudo code, as seen in figure 22. It has three variables: a decision model named *m*, a decision eventLog *L* and decisions *D*. First, it checks for all decisions *D* the attributes *A* that are influencing the decision (row 2). If the decision only contains attributes as input data and not a decision it returns a trivial decision dependency which means a decision with one or more input data attached to it (row 2-4). After that it analyzes if there are decisions that are influencing other decisions, for example within our example that has two related decisions (row 9-11). The algorithm uses a decision tree dt, with the decision as root node. It also creates a new decision tree dt_{new}. For all influencing attributes the algorithm creates a leaf to the new attribute (row 12-14). When all attributes are analyzed the algorithm checks if the leaves on the new decision tree are greater than the original decision tree. If so, the decision has not only input data, but also another decision as dependency (row 16-19). This way the relations between the decisions is found, so that the right DRD (Decision model *m*) can be drawn.

Algorithm FindDependency(Decision model m, Decision eventLog L, decisions D)
1: **for all** $d \in D$ **do**
2: A_{inf} {attributes that influences d}
3: **for all** $a_{inf} \in A_{inf}$ **do**
4: **if** D contains dd for a_{inf} **then**
5: **return** decision dependency dd = d {trivial dependency}
6: **end if**
7: **end for**
8: D_{inf} {possible influencing decision for d}
9: **for all** $d_{inf} \in D_{inf}$ **do**
10: dt_{inf} {decision tree where the root node splits according to d_{inf}}
11: dt_{new} {is dt_{inf}}
12: **for all** $leaf \in dt_{inf}$ **do**
13: dt_{leaf} {decision tree for d classifying the instances from leaf}
14: dt_{new} {add dt_{leaf} to the leaf of dt_{new}}
15: **end for**
16: $levels_{inf}$ {number of levels of dt_{new}}
17: $levels_{original}$ {number of levels of orginal decision tree for d}
18: **if** $levels_{inf} <= levels_{original}$ **then**
19: **return** decision dependency $d_{inf} \to d$ {non-trivial dependency}
20: **end if**
21: **end for**
22: **end for**

Figure 22: Decision dependency pseudocode

8. Data preparation

To test the algorithm a proof of concept was designed in Java using Jetbeans IntelliJ Idea version 2018.2. The WEKA data mining packages were used for the decision tree algorithm and the algorithm created by Bazhenova et al.(2016) was used as base. The source code is available at GitHub[1]. The algorithm was tested on a computer system with specifications shown in table 8.

Table 7: Computer system specifications

Item	**Details**
Laptop	Macbook Pro, 2018 (MacBookPro15,2)
Processor	Intel Core I5 (2,3GHz Quad-Core)
Memory	8gb 2133MHZ DDR3
Graphics	Intel Iris Plus 655 1536 MB
OS	OSX 10.14.6 (Mojave)

8.1. Algorithm test data

Four datasets were either created or gathered for testing the algorithm. The test data was gathered in August 2019. The four datasets consist of two synthetic datasets and two real-life datasets which are further described in detail in section 8.2.

Table 8: Overview of synthetic dataset

	Number of decisions in datasets	Number of executed decisions in log
Synthetic dataset 1	1	10
Synthetic dataset 2	2	5000

The synthetic datasets were created manually with the help of mock data generators[2] to fill in the data. Dataset 1 consists of one decision and dataset 2 consists of two decisions, as seen in table 9. Both datasets were checked on the aforementioned criteria for synthetic datasets by an additional researcher with a Ph.D. in decision management & Business Rules Management and ten years of research experience on the topic. The real-life datasets both consists of one decision, seen in table 10.

Table 9: Overview of real-life dataset

	Number of decisions in dataset	Number of executed decisions in log
Industrial dataset	1	50
Hospital dataset	1	23000

[1] Github.com/matthijsberk/DMiner
[2] www.mockaroo.com

8.2. Test Setup

The setup consisted of four different datasets: two synthetic datasets and two real-life datasets. The two synthetic datasets, as presented in figure 23, had the following structure:

Fraud	Amount	Alert

Alert	Fine	Finished

Figure 23: Representation of the synthetic datasets

The synthetic datasets consisted of two decisions. A decision Alert and a decision Finished. Alert consists of two conditions Fraud and Amount. Finished consists of two conditions, namely the conclusion of the first decision Alert and another condition Fine. The first decision was used to check the basics of the algorithm, while the second decision was used to see if derived decisions were taken into account and also to test if the algorithm could handle large datasets.

The real-life datasets were made available by two organizations: one organization in the industrial sector, the other was a hospital. Both organizations are located in the Netherlands.
The data from the first dataset was extracted from the company's SAP database and consisted of one decision. The decision model was known by the organisations and is handed over, so it could be used for the validation. The second real-life dataset was based on a decision made in a hospital within the treatment of Parkinson's disease. Both datasets were made anonymous by the organisations using synonyms or by replacing the attributes with isomorphic data. Unnecessary columns were deleted before the execution of the algorithm.

The first real-life dataset, as can be seen in figure 24, consisted of the necessary attributes as stated in section 4.3.2. The attributes are a Case ID, conditions and a conclusion.

CaseID	EBELN	BUKRS	VEND_NAME	KOSTL	WRBTR

Figure 24: Representation of the columns of the industrial dataset

The conclusion was the value of the order as 'calculated' by four conditions: EBELN, BUKRS, VEND_NAME and KOSTL (shown in table 11).

Table 10: Attribute description

Attribute	Function	Description
CaseID	Case ID	Unique number
EBELN	Condition	Purchase Document Number
BUKRS	Condition	Company Code (Domain)
VEND_NAME	Condition	Vendor Name
KOSTL	Condition	Cost Center Information

WRBTR	Conclusion	Amount on invoice

The second real-life dataset, seen in figure 25, had more columns than the first dataset. Again a case ID was present, in the form of an anonymous patient number. However, this dataset consisted of multiple timestamps. The timestamps in this case are also a condition for the decision. To check if the decision was made, the last column, Datum_midden (date-middle), is included. This date is a validation check by the system if the treatment has started but is excluded for the algorithm as this is not a condition.

DBC_ Profiel	Anoniem _patient	Patient_ Geslacht	Externe_ verrichting	Begindatum _ verrichting	DBC_Begin- datum	COACH_ patient	Datum_midden

Figure 25: Representation of the columns of the hospital dataset

After the pre-processing, the attributes used for the decision COACH_patient can be found in table 12. All real-life datasets were stored in a separate CSV file as input for the algorithm. The datasets were all exectuted three times, to check if the output was the same.

Table 11: Attributes for decision COACH

Attribute	**Function**	**Description**
DBC_Profiel	Condition	treatment DBC ID (used for insurance)
Anoniem_Patientnummer	CaseID	ID of Patient
Patient_Geboortejaar	Condition	Birth year of patient
Patient_Geslacht	Condition	Gender of patient
Begindatum_verrichtingen	Condition	Start date of treatment in clinic
DBC_Begindatum	Condition	Start date of treatment for Insurance
COACH_Patient	Conclusion	Eligible for COACH program
Datum_midden	Extra data (excluded in pre-processing)	Date to ensure decision is executed

For all datasets an excerpt of the used data can be found in Appendix A.

9. Algorithm validation

In this section, the results of the designed algorithm are presented by testing the algorithm using four different decision event logs. Two synthetic datasets and two reallife datasets. First, a short introduction about the dataset is given. The next step is the output of the DRD and the differences between the output of the algorithm compared to the known model. The last step is the comparisons between the underlying decision logic, visualized as decision tables. The treatment validation took place between August 2019 and October 2019.

9.1. Synthetic datasets

The dataset consisted of two decisions, where the decision Determine *Finished* was derived from the decision *Alert*. The input data for the decision Alert were Fraud and *Amount*, while the input data for Determine *Finished* was the conclusion of *Alert* and *Fine,* see figure 27 and 28. For the synthetic datasets the only validation is if the decisions and the underlying business logic are the same as created beforehand by the researcher for validation of the algorithm output.

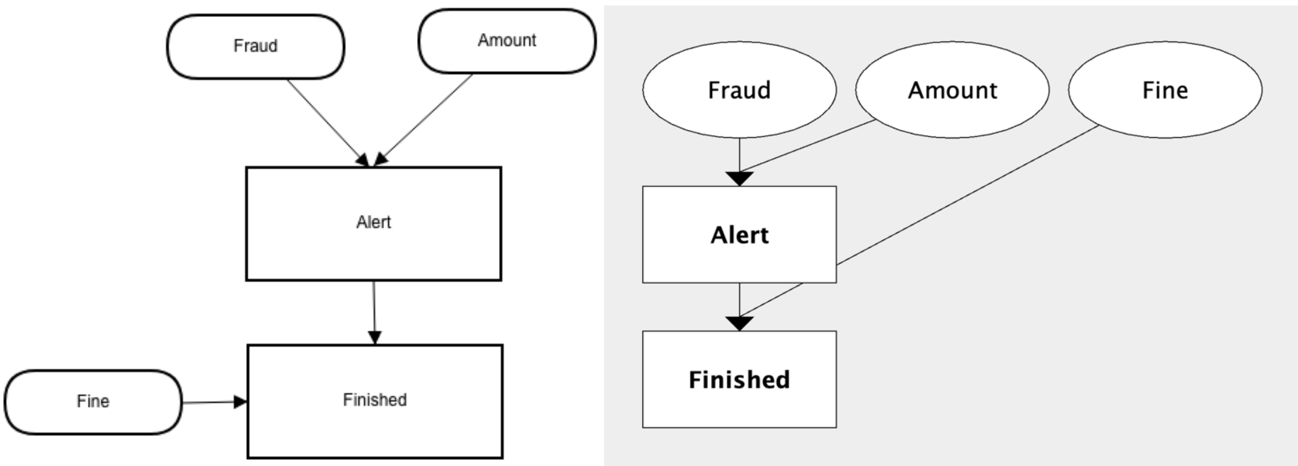

Figure 26: Decision Requirement Diagram known model *(l)* and algorithm *(r)* output

First the decision *Alert* was executed using the algorithm. The decision requirement diagram, seen in figure 26, was the same as the known model. Figure 27 shows the business logic created beforehand by the researcher as validation, while figure 28 shows the output of the business logic of that decision. The decision consisted of only four different rules. The algorithm gave the same output as the theoretical model created beforehand.

Rule Number	Fraud	Amount	Alert
1	Ja	500	Ja
2	Nee	100000	Nee
3	Ja	15000	Nee
4	Nee	500	Nee

Figure 27: Decision table from known model

Rule Number	Fraud	Amount	Alert
1	Ja	500	Ja
2	Nee	100000	Nee
3	Nee	500	Nee
4	Ja	15000	Nee

Figure 28: Decision table output from algorithm

The business logic of the decision *Finished* is bigger due to the fact that the data for this decision consisted of 5000 rows. Each row stands for an individual executed decision. The output of the algorithm consisted of 13 different rules, which was the same as the theoretical model. The decision table can be seen in figure 29 and 30. The algorithm was executed three times in a row to confirm and validate that the decision tables were the same every time the algorithm was used, which was indeed the case. The total running time of the algorithm for both decisions was 22 seconds.

Rule Number	Alert	Fine	Finished
1	Nee	150000	Akkoord
2	Nee	500	Niet Akkoord
3	Ja	150000	Niet Akkoord
4	Ja	300000	Akkoord
5	Ja	1000	Akkoord
6	Nee	10000	Akkoord
7	Nee	5000	Niet Akkoord
8	Ja	10000	Niet Akkoord
9	Ja	15000	Akkoord
10	Nee	1000	Niet Akkoord
11	Nee	15000	Niet Akkoord
12	Nee	300000	Niet Akkoord
13	Ja	500	Niet Akkoord

Figure 29: Decision table algorithm output

Rule Number	Alert	Fine	Finished
1	Nee	150000	Akkoord
2	Nee	500	Niet Akkoord
3	Nee	5000	Niet Akkoord
4	Ja	10000	Niet Akkoord
5	Ja	150000	Niet Akkoord
6	Ja	300000	Akkoord
7	Nee	1000	Niet Akkoord
8	Nee	15000	Niet Akkoord
9	Nee	300000	Niet Akkoord
10	Ja	1000	Akkoord
11	Nee	10000	Akkoord
12	Ja	15000	Akkoord
13	Ja	500	Niet Akkoord

Figure 30: Decision table created beforehand

9.2. Industrial dataset

The first real-life dataset was made available by a company in the industrial sector. It is an extraction from their SAP database where the invoices are stored. In this section two different questions for the same dataset are elaborated upon.

9.2.1. Industrial dataset with vendor

The initial question the company had was:

Which invoices were coded wrong?

To use the given dataset as validation for the algorithm the decision and the underlying business logic is given by the organization. This way the output of the algorithm can be compared to the output found by the organization.
This dataset was small in size with only 50 executed decision events, but complex for the algorithm due to the high number of input data and different cases which in theory could mean many different business rules. The DRD output of the known model is the same as the output of the algorithm and the result is shown in figure 31.

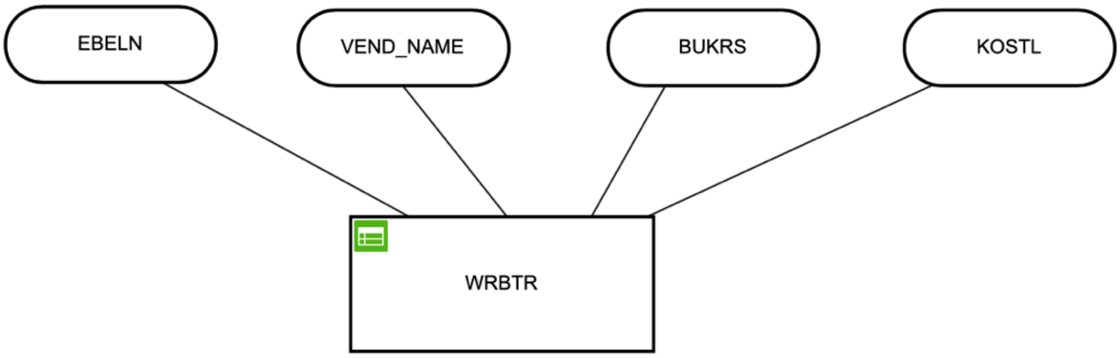

Figure 31: Decision Requirement Diagram Industrial sector known model

The underlying business logic of the decision was also created by the algorithm. A part of the decision table is shown in figure 32. The complete decision table can be found in Appendix B together with a screenshot of the algorithm output of the DRD. The total running time was four seconds.

Rule Number	EBELN	BUKRS	VEND_NAME	KOSTL	WRBTR
1	Geen EBELN	3210	Nilton Hotels	Geen waarde ingevuld	500.0
2	Geen EBELN	400	MeteoGroup	Geen waarde ingevuld	5000.0
3	Geen EBELN	3210	De Berg Vervoer	Geen waarde ingevuld	50.0
4	Geen EBELN	3210	Computer Company	Geen waarde ingevuld	500.0
5	Geen EBELN	3210	Best South	Geen waarde ingevuld	500.0
6	Geen EBELN	3210	Carwash	Geen waarde ingevuld	50.0

Figure 32: Excerpt of decision table output with algorithm

9.2.2. Industrial dataset without vendor

In the previous decision, the vendor is taken into account. To show that another question can be asked with the same dataset and to allow the organization to design rules independent from the vendor, the algorithm was also used to find the rules without the vendor as input data. The question is still the same, but the decision table will be different as the vendor is not taken into account. While this does not change the inherent mechanism of the algorithm, it shows that it can support the organization with different questions for the discovery of decisions using the dataset.

Without the vendor, the total decision table became much smaller, as shown in figure 33. It provides an answer to the question what conditions were necessary to get the right amount in column *WRBTR*. The DRD in figure 34 shows the used input data for the conclusion WRBTR using a DRD. The total running time for the algorithm was three seconds.

EBELN	BUKRS	KOSTL	WRBTR
Geen EBELN	3210	Geen waarde ingevuld	500
Geen EBELN	3555	Geen waarde ingevuld	50
Geen EBELN	3210	62210H3210	50
Geen EBELN	3210	62220H3210	50
Geen EBELN	3210	62230H3210	50
Geen EBELN	3556	Geen waarde ingevuld	500
Geen EBELN	3036	Geen waarde ingevuld	50
Geen EBELN	3210	80030H3210	5000
Geen EBELN	3557	Geen waarde ingevuld	5000
Geen EBELN	3554	Geen waarde ingevuld	5000
Geen EBELN	3210	62100H3210	5000
Geen EBELN	400	Geen KOSTL	50
Geen EBELN	3210	80020H3210	500

Figure 33: Business logic without vendor

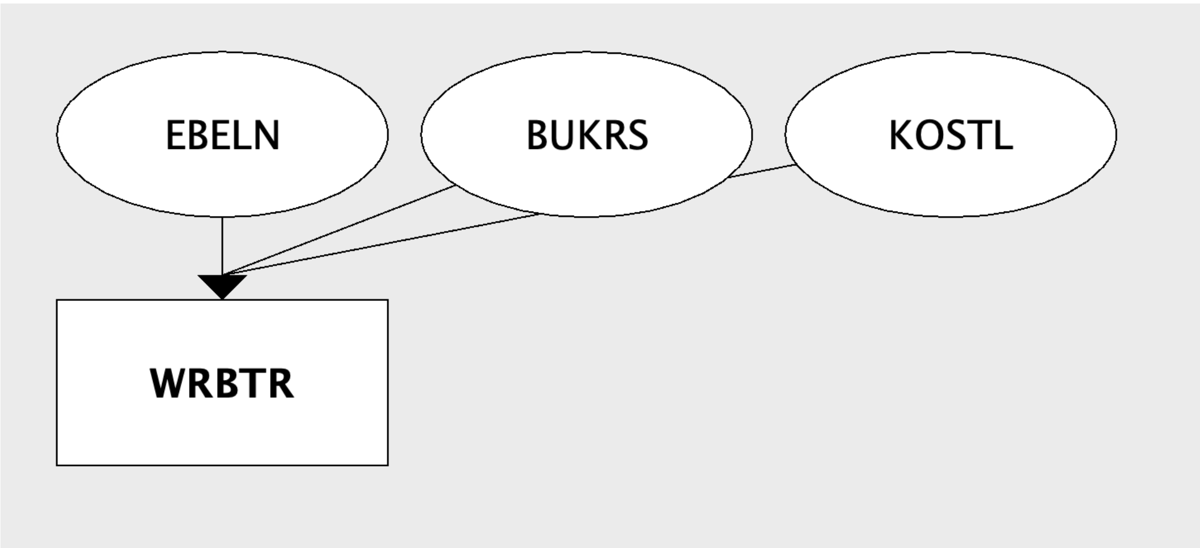

Figure 34: Decision Requirement Diagram without vendor (output algorithm)

9.3. Hospital dataset

The second real-life dataset is an extraction from a Dutch hospital. The question the hospital posed was the following:

Did we include all patients that were eligible into the COACH program?

To find an answer to this question, a researcher from the hospital analyzed all rows to find the different conditions and variants for this decision. This took multiple weeks of work to find all variants of the decision. This question posed a new challenge for the testing of the algorithm due to the number of rows: a total of 23000 rows had to be analyzed.

Figure 35 shows the output of the algorithm. The input data was the same in both the output as the model created by the researcher. The underlying business logic created by the algorithm is shown in part in figure 36. The complete decision table is available in appendix B together with the rules made by the hospital. Time to output of this decision table including DRD was 1 minute and 9 seconds.

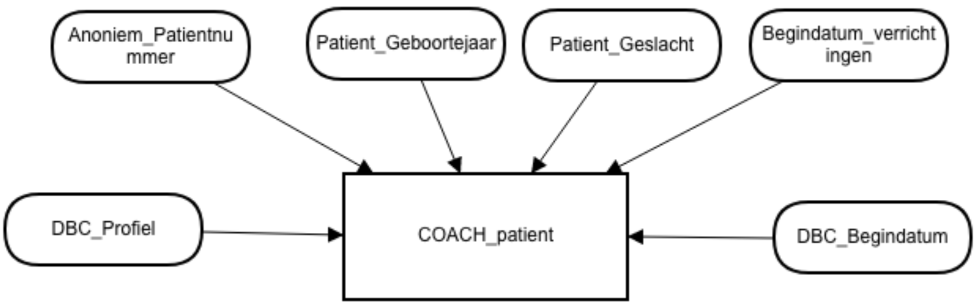

Figure 35: Decision Requirement Diagram COACH decision

DBC_Profiel	Geboortedatum	Geslacht	Begindatum	DBC_begindatum	Coach
30.11..0501	1957	Man	03/02/2015	03/02/2015	NULL
30.21..0501..111	1938	Vrouw	21/03/2012	21/05/2011	NULL
30.11.100.0501	1939	Man	07/01/2016	07/01/2016	NULL
30.21..0501	1943	Vrouw	02/10/2015	15/06/2015	NULL
30.21..0501	1944	Man	07/07/2014	18/11/2013	NULL
30.21.100.0501	1943	Man	28/12/2016	16/11/2016	x
30.21..0501	1950	Vrouw	04/06/2013	27/08/2012	x
30.11..0501	1957	Vrouw	05/12/2013	05/12/2013	x
30.11.100.0501	1960	Vrouw	19/01/2016	19/01/2016	x
30.11..0501	1943	Man	15/05/2013	18/03/2013	x

Figure 36: (part of) Decision table for COACH program

Table 12 shows the basic statistics for the different datasets. In this table the running time for each dataset is shown together with the number of cases and number of decisions.

Table 12: Summary of execution data

	Number of rows in decision event log	Number of different cases	Number of data elements	Number of rows in decision table	Number of decisions	Total Duration
Synthetic1	10	4	50	5	1	3 seconds
Synthetic2	5000	5000	30000	13	2	22 seconds
Industrial with vendor	50	50	250	42	1	4 seconds
Industrial without vendor	50	50	200	13	1	3 seconds
Hospital	23000	1670	10020	1600	1	1 minute 9 seconds

10. Conclusion

In this section, the conclusion of this research is presented. The research objectives will be adressed one by one, after which the conclusion of the research will be given.

RO1: *Give an overview of the current body of knowledge for decision mining.*

This research has found that current body of knowledge on decision mining is nascent. By conducting a literature review the context for decision mining using decision event logs is created. The literature review also showed a potential algorithm in the form of a fuzzy based mining technique used for the discovery of decisions in event logs.

RO2: *Give an overview of relevant algorithms (that are used in process mining) for decision mining.*

By discovering this specific algorithm, a potential algorithm was found. This research also showed that potential algorithms based on neural networks or Support Vector Machine could be interesting, but are not advisable due to the opaqueness of these kind of algorithms. The adapted and used algorithm is based on a miner used for the discovery of decisions from process event logs.

RO3: *Design and test algorithm for decision discovery.*

The algorithm is then adapted for the mining of decisions using a decision event log. It produces a decision table for each decision that is discovered. It also includes a DRD where the relations between the decisions and the input data was drawn. It can find multiple dependencies between decisions. This algorithm was tested using four different datasets to confirm it finds all business rules and can create a DRD. Before and during the design of the algorithm interviews were conducted and the interviewees pointed out that this discovered algorithm was potentially useful.

RO4: *Validate the created algorithm*

After the first test with synthetic data, the algorithm was tested using four different datasets. All datasets were executed within the algorithm and had a succesfull result. The output were decision tables using a first hit policy, together with the corresponding decision and input data. Especially the real-life datasets were useful for an extensive test as the variables are unclear.

How can a decision discovery algorithm be designed so that a decision requirement diagram can be extracted?

It is possible to design a decision discovery algorithm to extract decision requirement diagrams together with the underlying business logic from decision event logs. While research was done on mining decisions using process event logs, this was not present for decision event logs. The adapted algorithm now can discover decisions and create

not only a DRD, but also the underlying business logic. As it is tested with four different datasets, it shows potential.
The algorithm that was used as foundation for this study was limited to a dataset with set input data from event logs. This adapted algorithm can be used for all decision log data. However, this is only the first step as the results of this study also discovered new questions and problems, which are described in the future work section.

11. Discussion & future work

As every study, this study also has limitations. While the algorithm is tested with four different datasets, it is still possible that with enormous datasets it is not usable on normal computers. While we tried to provide different datasets to the algorithm from different sectors, it is not possible to tackle all different kinds of datasets. This should be tested more. It is also only possible to upload one csv file at a time. This is a limitation, as every decision must be uploaded separately. For convenience it would be better to upload one csv with all the decisions, or an option to bulk upload the different csv files. The algorithm does not have a problem with the order in which CSVs are uploaded to find relations, as it checks every time a CSV is uploaded if there is a relation with an already uploaded decision log.

The output of the decision table is based on a first hit policy. While this is a common practice in organizations, it would be better to create a unique hit policy decision table. At the moment this is only possible using neural networks and thus not useful as these kind of algorithms are opaque.

Also the number of people interviewed is small, this is due to the fact that the number of experts is very low on this subject. This is also shown in the literature review. However, even after three interviews data saturation was reached as seen in the similarities within the given answers.

The approach for this study only mapped process mining algorithms and one data mining algorithm, the c4.5 decision tree algorithm. It would be useful to map other data mining algorithms against the requirements found in this thesis. That way improvements to this existing algorithm can be found, but also opportunities for new algorithms to extract a decision requirement diagram and the underlying business logic.

Conceivably, there are other algorithms that might be adapted for the same purpose. Ultimately, every decision mining algorithm should be transparant, and a discovery algorithm should not optimize the data. For every found algorithm, the parts that use this optimization or are opaque must be changed or removed. Also, the creation of a decision requirement diagram must be build into every algorithm for showing the dependencies between decisions. Any alternative strategy adapting these algorithms would therefore require similar alterations and yield similar results as this study.

This study is a first step that shows that decision mining using a decision event log is possible. However, it shows more research questions. First, the efficiency of the algorithm. One of the limitations of algorithm is the speed. With more rows in the dataset, the algorithm becomes very slow. Future work might consist of other faster algorithms or an ensemble of algorithms to find these decisions and underlying decision logic. For example faster decision tree algorithms used in data mining. By using multiple algorithms the accuracy and speed could be improved.

Another suggestion is to add business knowledge to the DRD. Is it possible to find the business knowledge from decision logs and show this in the output? For finding business knowledge, different sources must be coupled as this is not found directly in the decision logs at the moment.

More research should be conducted on the other types of decision mining. This is only a first step in the discovery of decision mining, but also conformance and improvement of decisions could use algorithms so that organizations can evaluate and improve decisions found within their systems.

12. References

Aggarwal, C. C. (2014). *Data classification: Algorithms and applications*. *Data Classification: Algorithms and Applications*. https://doi.org/10.1201/b17320

Amara, N., & Landry, R. (2012). Counting citations in the field of business and management: Why use Google Scholar rather than the Web of Science. *Scientometrics*, *93*(3), 553–581. https://doi.org/10.1007/s11192-012-0729-2

Batoulis, K., Meyer, A., Bazhenova, E., Decker, G., & Weske, M. (2015). Extracting decision logic from process models. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9097*, 349–366. https://doi.org/10.1007/978-3-319-19069-3_22

Bazhenova, E. (2018). Discovery of Decision Models Complementary to Process Models, (May), 234.

Bazhenova, E., Buelow, S., & Weske, M. (2016). Business Information Systems, *255*, 237–251. https://doi.org/10.1007/978-3-319-39426-8

Bazhenova, E., & Weske, M. (2016). Deriving Decision Models from Process Models by Enhanced Decision Mining. In *Lecture Notes in Business Information Processing* (Vol. 202, pp. 444–457). https://doi.org/10.1007/978-3-319-42887-1_36

Boyer, J., & Mili, H. (2011). *Agile business rule development: Process, Architecture and JRules Examples.* Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19041-4

Dalle, O. (2012). On reproducibility and traceability of simulations. In *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)* (pp. 1–12). IEEE. https://doi.org/10.1109/WSC.2012.6465284

De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., … Ronneberger, O. (2018). Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature Medicine*. https://doi.org/10.1038/s41591-018-0107-6

De Leoni, M., & van der Aalst, W. M. P. (2013). Data-aware process mining: Discovering decisions in processes using alignments. In *Proceedings of the ACM Symposium on Applied Computing*. https://doi.org/10.1145/2480362.2480633

de Medeiros, A. K. A., van der Aalst, W. M. P., & Weijters, A. J. M. M. (2003). Workflow Mining: Current Status and Future Directions, (February), 389–406. https://doi.org/10.1007/978-3-540-39964-3_25

De Smedt, J., vanden Broucke, S. K. L. M., Obregon, J., Kim, A., Jung, J.-Y., & Vanthienen, J. (2017). Decision Mining in a Broader Context: An Overview of the Current Landscape and Future Directions. In M. La Rosa & P. Soffer (Eds.) (Vol. 132, pp. 197–207). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-319-58457-7_15

Dexter, L. (1970). Elite Interviewing. In *The SAGE Dictionary of Social Research Methods*. 1 Oliver's Yard, 55 City Road, London England EC1Y 1SP United Kingdom: SAGE Publications, Ltd. https://doi.org/10.4135/9780857020116.n60

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of Business process management*. Springer Berlin Heidelberg.

Franceschet, M. (2010). A comparison of bibliometric indicators for computer science scholars and journals on Web of Science and Google Scholar. *Scientometrics*, *83*(1), 243–258. https://doi.org/10.1007/s11192-009-0021-2

Garcia, C. dos S., Meincheim, A., Faria Junior, E. R., Dallagassa, M. R., Sato, D. M. V., Carvalho, D. R., … Scalabrin, E. E. (2019). Process mining techniques and applications – A systematic mapping study. *Expert Systems with Applications*, *133*(1), 260–295. https://doi.org/10.1016/j.eswa.2019.05.003

Gibbs, G. (2007). *Thematic Coding and Categorizing*. *Analyzing Qualitative Data*. 1 Oliver's Yard, 55 City Road, London England EC1Y 1SP United Kingdom: SAGE Publications, Ltd. https://doi.org/10.4135/9781849208574.n4

Günther, C. W., & van der Aalst, W. M. P. (2007). Fuzzy mining - Adaptive process simplification based on multi-perspective metrics. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4714 LNCS*, 328–343. https://doi.org/10.1007/978-3-540-75183-0_24

Harzing, A.-W., & Alakangas, S. (2016). Google Scholar, Scopus and the Web of Science: A longitudinal and cross-disciplinary comparison. *Scientometrics*, *106*(2), 787–804. https://doi.org/https://doi.org/10.1007/s11192-015-1798-9

Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainabilty of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(4), e1312. https://doi.org/10.1002/widm.1312

Janssens, L., Bazhenova, E., De Smedt, J., Vanthienen, J., & Denecker, M. (2016). Consistent integration of decision (DMN) and process (BPMN) models. *CEUR Workshop Proceedings*, *1612*, 121–128.

Mannhardt, F., de Leoni, M., Reijers, H. A., & van der Aalst, W. M. P. (2016a). Decision mining revisited-discovering overlapping rules. *International Conference on Advanced Information Systems Engineering*, 377–392.

Mannhardt, F., de Leoni, M., Reijers, H. A., & van der Aalst, W. M. P. (2016b). Decision Mining Revisited - Discovering Overlapping Rules (Vol. 16, pp. 377–392). https://doi.org/10.1007/978-3-319-39696-5_23

March, S. T., & Storey, V. C. (2008). Design Science in the Information Systems Discipline: Introduction to the special issue on information systems for an aging society. *MIS Quarterly*, *32*(4), 725–730. https://doi.org/10.1145/1183456.1183457

Meuser, M., & Nagel, U. (2009). The Expert Interview and Changes in Knowledge Production. *Interviewing Experts*, (1993), 17–42. https://doi.org/10.1057/9780230244276_2

Nelson, M. L., Rariden, R. L., & Sen, R. (2008). A lifecycle approach towards business rules management. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences* (pp. 113–123). IEEE.

Netjes, M., Reijers, H. A., & van der Aalst, W. M. P. (2006). Supporting the BPM life-cycle with FileNet. *CEUR Workshop Proceedings*, *364*, 135–146.

Object Management Group. (2016). Decision Model and Notation (DMN), Version 1.1. Retrieved February 15, 2018, from http://www.omg.org/spec/DMN/1.1

Rahmandad, H., & Sterman, J. D. (2012). Reporting guidelines for simulation-based research in social sciences. *System Dynamics Review*, *28*(4), 396–411. https://doi.org/10.1002/sdr.1481

Rovani, M., Maggi, F. M., de Leoni, M., & van der Aalst, W. M. P. (2015). Declarative process mining in healthcare. *Expert Systems with Applications*, *42*(23), 9236–9251. https://doi.org/10.1016/j.eswa.2015.07.040

Rozinat, A., & van der Aalst, W. M. P. (2006a). Decision mining in business processes. *BPM Center Report BPM-06-10, {…}*, (September), 16. https://doi.org/10.1039/b000000x

Rozinat, A., & van der Aalst, W. M. P. (2006b). Decision Mining in ProM. In S. Dustdar, J. L. Fiadeiro, & A. P. Sheth (Eds.), *Business Process Management: 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006. Proceedings* (pp. 420–425). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11841760_33

Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, *33*(1), 64–95. https://doi.org/10.1016/j.is.2007.07.001

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., … Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*, 354.

Smit, K., & Zoet, M. (2018). An organizational capability and resource-based perspective on business rules management. *International Conference on Information Systems 2018, ICIS 2018*, (2002), 1–17.

Smit, K., Zoet, M., & Berkhout, M. (2016). *Assess Malnutrition Risk Case Study - Technical Report*.

Taylor, S. J. E., Anagnostou, A., Fabiyi, A., Currie, C., Monks, T., Barbera, R., & Becker, B. (2017). Open science: Approaches and benefits for modeling & simulation. In *2017 Winter Simulation Conference (WSC)* (pp. 535–549). IEEE. https://doi.org/10.1109/WSC.2017.8247813

van der Aalst, W. M. P. (1998). Three Good Reasons for Using a Petri-Net-Based Workflow Management System. *Information and Process Integration in Enterprises*. https://doi.org/10.1.1.147.3781

van der Aalst, W. M. P. (2011a). *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media.

van der Aalst, W. M. P. (2011b). *Process Mining*. *Process Mining* (Vol. 5). https://doi.org/10.1007/978-3-642-19345-3

van der Aalst, W. M. P., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., … Wynn, M. (2012). Process mining manifesto. In *Lecture Notes in Business Information Processing* (Vol. 99 LNBIP, pp. 169–194). https://doi.org/10.1007/978-3-642-28108-2_19

van der Aalst, W. M. P., Netjes, M., & Reijers, H. A. (2007). Supporting the full BPM life-cycle using process mining and intelligent redesign. *Contemporary Issues in Database Design and Information Systems Development*, 100–132. https://doi.org/10.4018/978-1-59904-289-3.ch004

van der Aalst, W. M. P., Reijers, H., & Song, M. (2005). Discovering social networks from event logs. *Computer Supported Cooperative Work*. https://doi.org/10.1007/s10606-005-9005-9

van der Aalst, W. M. P., ter Hofstede, A., Weijters, A., & Weske, M. (2003). Business process management: A survey. In *Business Process Management* (pp. 1–12). https://doi.org/10.1007/3-540-44895-0_1

van der Aalst, W. M. P., & Weijters, A. (2004). Process mining: A research agenda. *Computers in Industry*. https://doi.org/10.1016/j.compind.2003.10.001

van der Aalst, W. M. P., & Weijters, A. (2005). Process Mining. In *Process-Aware Information Systems: Bridging People and Software through Process*

Technology. https://doi.org/10.1002/0471741442.ch10
van der Aalst, W. M. P., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2004.47
Van Dongen, B. F., & van der Aalst, W. M. P. (2004). Multi-phase process mining: Building instance graphs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3288*, 362–376. https://doi.org/10.1007/978-3-540-30464-7_29
Von Halle, B., & Goldberg, L. (2009). *The Decision Model: A Business Logic Framework Linking Business and Technology*. CRC Press.
Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(2), xiii–xxiii. https://doi.org/10.1.1.104.6570
Weske, M. (2012). *Business Process Management*. *Business Process Management* (2nd ed.). Springer. https://doi.org/10.1007/978-3-642-40176-3
Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. *Springer Berlin Heidelberg*. https://doi.org/10.1145/1810295.1810446
Wildgaard, L. (2015). A comparison of 17 author-level bibliometric indicators for researchers in Astronomy, Environmental Science, Philosophy and Public Health in Web of Science and Google Scholar. *Scientometrics*, *104*(3), 873–906. https://doi.org/10.1007/s11192-015-1608-4
Zoet, M. (2014). *Methods and Concepts for Business Rules Management* (1st ed.). Utrecht: Hogeschool Utrecht.
Zoet, M., Smit, K., & de Haan, E. (2014). Business Model for Business Rules.
Zoet, M., Versendaal, J., Ravesteyn, P., & Welke, R. (2011). Alignment of business process management and business rules. In *19th European Conference on Information Systems, ECIS 2011*.

Appendix A

Dataset for the decision Alert

Fraud	Amount	Alert
Nee	15000	Ja
Ja	500	Ja
Nee	100000	Nee
Nee	500	Nee
Ja	15000	Nee

Dataset for the decision Finished

Alert	Fine	Finished
Ja	15000	Akkoord
Nee	150000	Akkoord
Ja	10000	Akkoord
Ja	500	Akkoord
Nee	500	Niet Akkoord
Ja	150000	Niet Akkoord
Nee	300000	Akkoord
Ja	150000	Niet Akkoord
Ja	300000	Akkoord
Ja	5000	Niet Akkoord
Nee	150000	Niet Akkoord
Nee	5000	Akkoord
Ja	1000	Akkoord
Ja	1000	Akkoord
Nee	10000	Akkoord
Nee	5000	Niet Akkoord

Nee	5000	Niet Akkoord
Ja	10000	Niet Akkoord
Nee	300000	Akkoord
Nee	5000	Niet Akkoord
Nee	150000	Niet Akkoord
Ja	15000	Akkoord
Ja	300000	Niet Akkoord
Ja	5000	Niet Akkoord
Ja	10000	Niet Akkoord
Nee	10000	Akkoord
Ja	300000	Akkoord
Ja	5000	Akkoord
Ja	500	Akkoord
Ja	150000	Niet Akkoord
Ja	150000	Akkoord
Ja	150000	Niet Akkoord
Nee	1000	Niet Akkoord
Ja	15000	Niet Akkoord
Nee	5000	Niet Akkoord
Ja	10000	Niet Akkoord
Ja	15000	Akkoord
Ja	1000	Akkoord
Ja	300000	Niet Akkoord

Dataset for the decision WRBTR (reallife decision 1)

EBELN	BUKRS	VEND_NAME	KOSTL	WRBTR
EBELN	3210	Geelwegen Transport	Geen waarde ingevuld	500.0
Geen EBELN	3210	Nilton Hotels	Geen waarde ingevuld	500.0
Geen EBELN	3210	Personal Safety BV	Geen waarde ingevuld	5000.0
Geen EBELN	400	MeteoGroup	Geen waarde ingevuld	5000.0
Geen EBELN	400	MeteoGroup	Geen waarde ingevuld	5000.0
Geen EBELN	3210	Nilton Hotels	Geen waarde ingevuld	500.0
Geen EBELN	3210	Geelwegen Transport	Geen waarde ingevuld	500.0
Geen EBELN	3210	Nilton Hotels	Geen waarde ingevuld	500.0
Geen EBELN	3210	De Berg Vervoer	Geen waarde ingevuld	50.0
Geen EBELN	3210	Computer Company	Geen waarde ingevuld	500.0
Geen EBELN	3210	Best South	Geen waarde ingevuld	500.0
Geen EBELN	3210	Geelwegen Transport	Geen waarde ingevuld	50.0

Geen EBELN	3210	Best South	Geen waarde ingevuld	500.0
Geen EBELN	3210	Car Lease Nederland BV	80020H3210	100.0
Geen EBELN	3210	Carwash	Geen waarde ingevuld	50.0
Geen EBELN	3210	Car Lease Nederland BV	80020H3210	50.0
Geen EBELN	3210	Marinabay Ship Co	Geen waarde ingevuld	5000.0
Geen EBELN	3555	Car Lease Belgie BV	Geen waarde ingevuld	50.0
Geen EBELN	3210	Sea Robotics LTD	62210H3210	50.0
Geen EBELN	3556	AMT Neco	Geen waarde ingevuld	500.0
Geen EBELN	3210	Sea Robotics LTD	62480H3210	5000.0
Geen EBELN	3210	Sea Robotics LTD	62210H3210	5000.0
Geen EBELN	3210	Sea Robotics LTD	62480H3210	500000.0
Geen EBELN	3210	Sea Robotics LTD	62480H3210	5000.0
Geen EBELN	3210	Sea Robotics LTD	62480H3210	5000.0
Geen EBELN	3210	Sea Robotics LTD	62210H3210	5000.0

Dataset for the reallife decision COACH_patient (reallife decision 2)

DBC_Profiel	Patient_Geboortejaar	Patient_Geslacht	Begindatum_verr	DBC_Begindatum	COACH_patient
30.21..0501	1949	Man	13/11/2014	28/06/2014	NULL
30.11..0501	1957	Man	03/02/2015	03/02/2015	NULL
30.21..0501..111	1938	Vrouw	21/03/2012	21/05/2011	NULL
30.11.100.0501	1939	Man	07/01/2016	07/01/2016	NULL
30.21..0501	1943	Vrouw	02/10/2015	15/06/2015	NULL
30.21..0501	1944	Man	07/07/2014	18/11/2013	NULL
30.21..0501	1944	Man	03/12/2014	16/04/2014	NULL
30.11..0501	1953	Vrouw	18/12/2015	16/12/2015	NULL
30.21..0501	1931	Man	21/11/2012	26/01/2012	NULL
30.21..0501	1937	Man	04/11/2013	27/04/2013	NULL
30.21..0501	1937	Man	11/03/2014	15/03/2013	NULL
30.11..0501	1944	Man	01/05/2014	08/04/2014	NULL
30.21..0501	1937	Man	04/12/2013	08/09/2013	NULL
30.21..0501	1933	Man	23/12/2013	19/12/2013	NULL
30.21..0501	1939	Man	11/03/2014	04/11/2013	NULL
30.21..0501	1936	Man	01/07/2014	14/05/2014	NULL
30.21..0501	1938	Man	23/11/2012	09/09/2012	NULL

30.21..0501	1940	Vrouw	17/11/2014	24/09/2014	NULL

Appendix B

Industrial decision table with vendor name:

EBELN	BUKRS	VEND_NAME	KOSTL	WRBTR
Geen EBELN	3210	Nilton Hotels	Geen waarde ingevuld	500.0
Geen EBELN	400	MeteoGroup	Geen waarde ingevuld	5000.0
Geen EBELN	3210	De Berg Vervoer	Geen waarde ingevuld	50.0
Geen EBELN	3210	Computer Company	Geen waarde ingevuld	500.0
Geen EBELN	3210	Best South	Geen waarde ingevuld	500.0
Geen EBELN	3210	Carwash	Geen waarde ingevuld	50.0
Geen EBELN	3210	Marinabay Ship Co	Geen waarde ingevuld	5000.0
Geen EBELN	3555	Car Lease Belgie BV	Geen waarde ingevuld	50.0
Geen EBELN	3210	Sea Robotics LTD	62210H3210	50.0
Geen EBELN	3556	AMT Neco	Geen waarde ingevuld	500.0
Geen EBELN	3557	Postbezorging	Geen waarde ingevuld	100.0
Geen EBELN	3555	Telenet	Geen waarde ingevuld	500.0
Geen EBELN	3556	Russian Offshore	Geen waarde ingevuld	500.0
Geen EBELN	3036	BioRio BV	Geen waarde ingevuld	50.0
Geen EBELN	3210	Sea Equipment	80030H3210	5000.0
Geen EBELN	3210	Annebel Limited	Geen waarde ingevuld	500.0
Geen EBELN	3556	Atlas Services Group	Geen waarde ingevuld	5000.0
Geen EBELN	3210	Accountant 2	Geen waarde ingevuld	5000.0
Geen EBELN	3210	BlueCool	Geen waarde ingevuld	5000.0
Geen EBELN	3210	Port Health	Geen waarde ingevuld	500.0
Geen EBELN	3557	Atlas Services Group	Geen waarde ingevuld	5000.0
Geen EBELN	3210	Banketbakker Jannie Bakker	Geen waarde ingevuld	100.0
Geen EBELN	3210	Uitjes.nl	Geen waarde ingevuld	500.0
Geen EBELN	400	Holland Consulting Group	Geen waarde ingevuld	10000.0
Geen EBELN	3554	De accountant	Geen waarde ingevuld	5000.0
Geen EBELN	3333	Babycompany	Geen waarde ingevuld	50.0
Geen EBELN	400	Schaatsbaan Heereveen	Geen waarde ingevuld	500.0
Geen EBELN	3333	Baby's KADO	Geen waarde ingevuld	100.0
Geen EBELN	400	Language Service	Geen waarde ingevuld	500.0
Geen EBELN	400	Drop Offshore BV	Geen waarde ingevuld	5000.0
Geen EBELN	400	ConsultCRM Ltd	Geen waarde ingevuld	1000.0
Geen EBELN	3555	Telecom Group	Geen waarde ingevuld	50.0
Geen EBELN	3210	Network Innovations BV	Geen waarde ingevuld	500.0
Geen EBELN	3210	Damstad BV	62100H3210	5000.0
Geen EBELN	3556	Maritieme industriegroep	Geen waarde ingevuld	500.0
Geen EBELN	400	Anglo Ship Management	Geen KOSTL	50.0
Geen EBELN	3556	AMT	Geen waarde ingevuld	500.0
Geen EBELN	3556	Offshore Brazil	Geen waarde ingevuld	500.0
Geen EBELN	3210	Car Lease Nederland BV	80020H3210	500.0

Hospital decision table:

30.11..0501	1940	Man	17/06/2015	24/03/2015	NULL
30.21..0501..111	1951	Man	21/02/2012	01/09/2011	NULL
30.21..0501	1947	Man	12/02/2014	20/05/2013	NULL
30.11.100.0501	1936	Vrouw	04/03/2016	04/03/2016	NULL
30.21..0501	1939	Man	13/08/2014	28/10/2013	NULL
30.21..0501	1933	Man	28/08/2013	06/04/2013	NULL
30.21..0501	1945	Man	24/02/2015	16/06/2014	NULL
30.11..0501	1956	Man	12/05/2015	06/05/2015	NULL
30.11..0501	1937	Man	13/06/2014	01/04/2014	NULL
30.21..0501	1941	Vrouw	24/08/2015	20/07/2015	NULL
30.21..0501	1939	Man	27/01/2014	16/09/2013	NULL
30.21..0501	1942	Vrouw	29/05/2013	10/04/2013	NULL
30.11.100.0501	1958	Man	21/10/2016	21/10/2016	NULL
30.11.100.0501	1938	Man	23/09/2016	08/09/2016	NULL
30.11.100.0501	1945	Man	10/03/2016	26/02/2016	NULL
30.21..0501	1949	Man	08/10/2013	17/03/2013	NULL
30.11..0501	1931	Man	12/02/2014	20/01/2014	NULL
30.11.100.0501	1951	Vrouw	17/11/2016	20/10/2016	NULL
30.11..0501	1951	Man	27/08/2015	25/08/2015	NULL
30.11.100.0501	1930	Man	22/03/2017	22/03/2017	NULL
30.11..0501	1969	Vrouw	06/03/2015	06/03/2015	NULL
30.21..0501	1927	Man	30/01/2013	04/07/2012	NULL

30.21..0501	1930	Vrouw	07/11/2012	20/05/2012	NULL
30.21..0501	1934	Man	03/04/2013	30/04/2012	NULL
30.21.100.0501	1950	Man	22/03/2016	09/01/2016	NULL
30.11.100.0501	1958	Man	01/02/2016	01/02/2016	NULL
30.21..0501	1937	Man	30/10/2012	05/07/2012	NULL
30.21..0501	1943	Man	27/03/2014	05/07/2013	NULL
30.21..0501	1952	Vrouw	17/02/2014	28/08/2013	NULL
30.21..0501	1941	Man	13/08/2013	23/06/2013	NULL
30.11..0501	1946	Man	02/12/2014	28/10/2014	NULL
30.21..0501	1935	Man	29/01/2013	01/10/2012	NULL
30.21..0501	1945	Vrouw	18/02/2014	16/03/2013	NULL
30.21..0501	1933	Man	07/04/2015	21/07/2014	NULL
30.21..0501	1943	Man	01/07/2015	10/07/2014	NULL
30.21..0501	1944	Vrouw	04/12/2013	25/06/2013	NULL
30.11.100.0501	1940	Vrouw	14/04/2016	18/01/2016	NULL
30.21..0501	1957	Man	20/11/2014	24/02/2014	NULL
30.21..0501	1947	Man	01/12/2014	19/11/2014	NULL
30.11.100.0501	1951	Vrouw	24/03/2016	14/03/2016	NULL
30.11..0501	1936	Man	30/01/2014	09/01/2014	NULL
30.21..0501	1941	Man	17/06/2013	06/01/2013	NULL
30.21..0501	1952	Man	08/09/2014	25/06/2014	NULL
30.11..0501	1953	Vrouw	02/05/2013	10/04/2013	NULL
30.21..0501	1933	Man	12/08/2014	11/07/2014	NULL

30.21..0501	1946	Vrouw	19/11/2014	15/04/2014	NULL
30.21..0501	1940	Vrouw	21/01/2013	01/11/2012	NULL
30.21..0501	1942	Man	18/02/2014	06/09/2013	NULL
30.21..0501	1942	Vrouw	07/02/2014	14/01/2014	NULL
30.21..0501	1931	Vrouw	06/11/2013	08/08/2013	NULL
30.21..0501	1935	Man	28/11/2014	21/12/2013	NULL
30.21..0501	1938	Vrouw	20/05/2014	19/10/2013	NULL
30.11.100.0501	1948	Vrouw	19/02/2016	01/02/2016	NULL
30.11..0501	1936	Vrouw	24/11/2015	19/11/2015	NULL
30.21..0501	1929	Man	05/02/2013	22/11/2012	NULL
30.21..0501	1946	Man	07/04/2014	30/08/2013	NULL
30.21..0501	1951	Man	20/01/2014	25/07/2013	NULL
30.21..0501	1933	Man	03/03/2015	06/08/2014	NULL
30.21..0501	1940	Man	30/03/2015	19/10/2014	NULL
30.11.100.0501	1944	Vrouw	18/10/2016	28/09/2016	NULL
30.11..0501	1938	Vrouw	21/05/2015	22/04/2015	NULL
30.11..0501	1933	Man	01/10/2015	17/09/2015	NULL
30.11.100.0501	1935	Man	15/07/2016	17/06/2016	NULL
30.11.100.0501	1940	Vrouw	28/06/2016	20/06/2016	NULL
30.21..0501	1931	Man	10/06/2014	07/11/2013	NULL
30.21..0501	1932	Man	04/03/2013	01/10/2012	NULL
30.11..0501	1933	Man	27/11/2013	30/09/2013	NULL
30.11.100.0501	1939	Vrouw	12/10/2016	03/10/2016	NULL

30.11..0501	1941	Vrouw	07/09/2015	23/07/2015	NULL
30.21..0501	1953	Man	28/07/2015	10/06/2015	NULL
30.21..0501	1942	Man	03/12/2013	18/04/2013	NULL
30.21..0501	1939	Vrouw	18/03/2014	13/09/2013	NULL
30.21..0501	1948	Man	06/01/2014	14/03/2013	NULL
30.21..0501	1953	Vrouw	29/09/2014	06/03/2014	NULL
30.11..0501	1970	Man	18/09/2015	09/09/2015	NULL
30.21..0501	1960	Man	06/01/2014	08/07/2013	NULL
30.21..0501	1943	Man	24/02/2014	18/06/2013	NULL
30.21..0501	1944	Man	12/02/2014	22/01/2014	NULL
30.21..0501	1948	Man	21/11/2013	03/01/2013	NULL
30.21..0501	1942	Man	01/09/2015	27/07/2015	NULL
30.21..0501	1938	Man	17/10/2013	28/08/2013	NULL
30.11.100.0501	1931	Vrouw	16/09/2016	16/09/2016	NULL
30.21..0501	1936	Man	04/06/2015	03/06/2015	NULL
30.11..0501	1948	Man	17/12/2014	23/09/2014	NULL
30.21..0501	1935	Vrouw	18/07/2013	24/11/2012	NULL
30.21..0501	1929	Vrouw	02/12/2013	04/10/2013	NULL
30.21.100.0501	1943	Vrouw	08/02/2017	04/01/2017	NULL
30.11.100.0501	1947	Vrouw	08/09/2016	17/08/2016	NULL
30.21..0501	1928	Man	10/05/2013	14/03/2013	NULL
30.11.100.0501	1939	Man	25/09/2017	25/09/2017	NULL
30.21..0501	1930	Man	05/02/2014	11/05/2013	NULL

30.21..0501	1952	Vrouw	10/02/2014	16/04/2013	NULL
30.11..0501	1930	Vrouw	24/12/2014	28/10/2014	NULL
30.21..0501	1937	Man	03/08/2015	19/07/2015	NULL
30.11..0501	1928	Man	23/01/2015	06/01/2015	NULL
30.21..0501	1946	Man	08/08/2013	27/08/2012	NULL
30.21..0501	1931	Vrouw	04/12/2012	04/07/2012	NULL
30.21..0501	1936	Vrouw	04/07/2014	15/04/2014	NULL
30.21..0501	1940	Vrouw	22/04/2015	29/06/2014	NULL
30.21..0501	1932	Vrouw	05/06/2013	31/01/2013	NULL
30.11.100.0501	1939	Vrouw	29/07/2016	27/07/2016	NULL
30.21..0501	1945	Man	04/06/2014	19/09/2013	NULL
30.21..0501	1958	Man	03/06/2015	14/12/2014	NULL
30.21..0501	1935	Man	04/11/2013	22/03/2013	NULL
30.21..0501	1936	Vrouw	24/10/2013	30/04/2013	NULL
30.11..0501	1926	Vrouw	29/09/2014	01/09/2014	NULL
30.11.100.0501	1936	Vrouw	14/07/2016	16/06/2016	NULL
30.21..0501	1934	Vrouw	11/10/2013	08/12/2012	NULL
30.21..0501	1938	Man	14/10/2014	22/12/2013	NULL
30.21..0501	1938	Vrouw	28/10/2014	02/07/2014	NULL
30.11..0501	1953	Man	25/06/2014	09/04/2014	NULL
30.11..0501	1938	Vrouw	01/09/2015	08/06/2015	NULL
30.21..0501	1946	Man	12/07/2013	19/09/2012	NULL
30.11..0501	1935	Vrouw	10/09/2014	24/06/2014	NULL

30.21..0501..111	1930	Man	31/08/2012	07/10/2011	NULL
30.21..0501	1949	Vrouw	24/01/2014	31/10/2013	NULL
30.11..0501	1954	Vrouw	24/09/2015	30/06/2015	NULL
30.11.100.0501	1946	Man	06/12/2016	18/11/2016	NULL
30.11.100.0501	1939	Vrouw	18/01/2016	18/01/2016	NULL
30.21.100.0501	1943	Man	28/12/2016	16/11/2016	x
30.21..0501	1950	Vrouw	04/06/2013	27/08/2012	x
30.11..0501	1957	Vrouw	05/12/2013	05/12/2013	x
30.11.100.0501	1960	Vrouw	19/01/2016	19/01/2016	x
30.11..0501	1943	Man	15/05/2013	18/03/2013	x
30.11.100.0501	1957	Vrouw	09/01/2017	28/11/2016	x
30.11.100.0501	1954	Vrouw	24/04/2017	10/04/2017	x
30.11.100.0501	1944	Man	26/04/2017	29/03/2017	x
30.11.100.0501	1961	Man	24/04/2017	20/04/2017	x
30.11.100.0501	1963	Vrouw	13/02/2017	22/12/2016	x
30.11.100.0501	1943	Man	16/01/2017	22/12/2016	x
30.11.100.0501	1959	Man	04/10/2017	23/08/2017	x
30.21.100.0501	1966	Man	02/10/2017	23/07/2017	x
30.21..0501	1956	Man	15/04/2013	17/06/2012	x
30.21..0501	1940	Man	21/08/2014	01/02/2014	x
30.21..0501	1956	Vrouw	17/10/2014	19/05/2014	x
30.11.100.0501	1958	Man	23/08/2016	20/07/2016	x
30.21..0501	1945	Vrouw	21/10/2013	18/08/2013	x

30.11..0501	1953	Vrouw	19/03/2013	19/03/2013	x
30.21..0501	1970	Vrouw	07/01/2014	21/03/2013	x
30.21..0501	1973	Man	28/01/2013	22/10/2012	x
30.11.100.0501	1946	Vrouw	19/06/2017	19/06/2017	x
30.21..0501	1943	Man	01/08/2013	19/08/2012	x
30.21..0501	1942	Man	29/04/2013	21/04/2013	x
30.21..0501	1936	Man	17/06/2013	01/07/2012	x
30.11..0501	1951	Vrouw	26/08/2014	05/08/2014	x
30.21..0501	1944	Vrouw	13/11/2014	06/12/2013	x
30.11..0501	1949	Man	16/10/2013	07/10/2013	x
30.11.100.0501	1936	Man	11/07/2016	13/06/2016	x
30.11..0501	1952	Man	19/03/2013	13/03/2013	x
30.21..0501	1961	Vrouw	04/03/2013	14/08/2012	x
30.11..0501	1948	Man	06/10/2014	16/09/2014	x
30.21..0501	1935	Vrouw	13/05/2013	08/04/2013	x
30.11..0501	1944	Man	23/06/2014	28/03/2014	x
30.21..0501	1932	Man	27/03/2014	17/02/2014	x
30.21..0501	1956	Man	16/12/2014	07/03/2014	x
30.21..0501	1930	Man	19/02/2014	28/04/2013	x
30.21..0501	1950	Vrouw	24/04/2014	04/08/2013	x
30.11..0501	1949	Man	07/07/2014	03/06/2014	x
30.21..0501	1954	Vrouw	02/01/2014	06/12/2013	x
30.11..0501	1942	Man	30/09/2015	21/09/2015	x

30.21..0501	1950	Man	10/11/2014	25/06/2014	x
30.11.100.0501	1956	Man	11/11/2016	11/10/2016	x
30.21..0501	1937	Man	31/01/2013	18/03/2012	x
30.21..0501	1945	Man	29/09/2015	23/09/2015	x
30.21..0501	1949	Vrouw	30/05/2013	02/11/2012	x
30.11.100.0501	1940	Man	19/05/2016	19/05/2016	x
30.11..0501	1969	Vrouw	06/12/2013	20/11/2013	x
30.21..0501	1938	Vrouw	09/04/2013	19/04/2012	x
30.11.100.0501	1936	Vrouw	04/10/2016	04/10/2016	x
30.21..0501	1944	Man	09/09/2013	08/04/2013	x
30.11..0501	1952	Man	05/06/2015	23/04/2015	x
30.11..0501	1950	Man	05/02/2015	02/02/2015	x
30.11.100.0501	1953	Man	10/04/2017	23/01/2017	x
30.21..0501	1947	Man	16/04/2014	04/10/2013	x
30.21..0501	1952	Man	14/05/2014	07/07/2013	x
30.21..0501	1938	Man	31/08/2015	12/11/2014	x
30.11..0501	1953	Vrouw	10/03/2014	17/02/2014	x
30.21..0501	1941	Vrouw	19/05/2014	18/03/2014	x
30.11..0501	1945	Vrouw	23/06/2014	27/03/2014	x
30.21..0501	1937	Vrouw	07/07/2014	31/08/2013	x
30.21..0501	1940	Vrouw	24/06/2013	01/07/2012	x
30.11..0501	1936	Man	09/04/2013	09/04/2013	x
30.21..0501	1946	Man	25/07/2013	08/11/2012	x

30.21..0501	1941	Man	26/02/2013	05/01/2013	x
30.11..0501	1941	Vrouw	09/01/2014	09/01/2014	x
30.11.100.0501	1958	Vrouw	01/03/2016	16/02/2016	x
30.21..0501	1947	Man	10/07/2014	20/05/2014	x
30.11.100.0501	1951	Man	23/08/2016	18/07/2016	x
30.21..0501	1936	Man	24/09/2013	26/08/2013	x
30.11..0501	1945	Man	22/12/2015	22/12/2015	x
30.21..0501	1939	Vrouw	04/03/2013	05/03/2012	x
30.11.100.0501	1952	Vrouw	28/01/2016	05/01/2016	x
30.21..0501	1948	Man	15/08/2013	07/02/2013	x
30.11..0501	1949	Man	11/11/2014	01/09/2014	x
30.21..0501	1948	Vrouw	07/07/2014	04/09/2013	x
30.11..0501	1951	Man	12/08/2014	08/07/2014	x
30.21..0501	1956	Vrouw	01/08/2013	05/04/2013	x
30.21..0501	1951	Man	15/05/2014	29/06/2013	x
30.11..0501	1950	Vrouw	17/01/2013	20/12/2012	x
30.21..0501	1952	Man	11/07/2013	17/02/2013	x
30.21..0501	1955	Vrouw	24/02/2015	24/03/2014	x
30.21..0501	1944	Man	27/02/2013	05/07/2012	x
30.21..0501	1944	Man	06/02/2013	15/04/2012	x
30.21..0501	1955	Man	21/03/2014	20/05/2013	x
30.11.100.0501	1940	Man	01/11/2016	26/09/2016	x
30.21..0501	1952	Man	13/08/2014	11/02/2014	x

30.21..0501	1935	Man	02/06/2014	23/06/2013	x
30.11.100.0501	1957	Man	01/08/2016	15/06/2016	x
30.21..0501	1943	Vrouw	30/10/2013	07/04/2013	x
30.21..0501	1944	Vrouw	02/03/2015	11/03/2014	x
30.11..0501	1943	Vrouw	15/06/2015	23/03/2015	x
30.21..0501	1936	Vrouw	25/06/2013	30/12/2012	x
30.21..0501	1945	Man	01/09/2014	10/09/2013	x
30.21..0501	1948	Man	12/08/2013	02/12/2012	x
30.11..0501	1955	Man	11/03/2013	11/03/2013	x
30.11..0501	1945	Vrouw	30/07/2014	18/06/2014	x
30.21..0501	1944	Man	23/05/2013	18/03/2013	x
30.11..0501	1954	Vrouw	10/04/2014	08/04/2014	x
30.21..0501	1950	Man	26/08/2014	23/06/2014	x
30.21.100.0501	1941	Vrouw	05/04/2016	11/02/2016	x
30.21..0501	1948	Man	17/02/2014	11/01/2014	x
30.11..0501	1945	Man	11/03/2013	11/03/2013	x
30.11..0501	1954	Man	09/08/2013	25/07/2013	x
30.21..0501	1951	Man	05/08/2013	19/01/2013	x
30.21..0501	1951	Man	10/12/2013	07/10/2013	x
30.21..0501	1956	Man	15/05/2013	22/03/2013	x
30.21..0501	1933	Vrouw	05/11/2013	07/05/2013	x
30.21..0501	1942	Vrouw	08/04/2015	25/08/2014	x
30.21.100.0501	1953	Man	04/04/2016	16/02/2016	x

30.21..0501	1953	Man	07/07/2014	07/10/2013	x
30.21..0501	1937	Man	25/01/2016	22/11/2015	x
30.11.100.0501	1951	Vrouw	14/02/2016	18/01/2016	x
30.21..0501	1961	Man	23/05/2013	14/06/2012	x
30.21..0501	1943	Man	23/05/2013	18/11/2012	x
30.11.100.0501	1967	Man	11/07/2016	08/06/2016	x
30.21..0501	1942	Vrouw	07/03/2013	22/12/2012	x
30.21..0501	1949	Man	26/05/2014	19/01/2014	x
30.21..0501	1946	Man	04/07/2013	26/07/2012	x
30.21..0501	1953	Vrouw	10/09/2013	27/09/2012	x
30.21..0501	1942	Vrouw	03/10/2013	12/07/2013	x
30.21..0501	1938	Man	13/03/2013	03/08/2012	x
30.11.100.0501	1962	Man	13/09/2016	23/08/2016	x
30.21..0501	1942	Man	02/04/2013	27/01/2013	x
30.21..0501	1938	Man	15/04/2013	30/08/2012	x
30.21..0501	1959	Vrouw	23/04/2013	04/02/2013	x
30.21..0501	1960	Man	21/10/2013	27/01/2013	x
30.21..0501	1931	Vrouw	20/08/2014	02/12/2013	x
30.11..0501	1946	Man	31/01/2014	30/12/2013	x
30.11..0501	1945	Man	12/06/2014	11/06/2014	x
30.21..0501	1949	Man	01/07/2014	25/08/2013	x
30.21..0501	1967	Vrouw	06/05/2014	18/05/2013	x
30.21..0501	1971	Vrouw	09/07/2014	07/07/2014	x

30.21..0501	1959	Vrouw	12/03/2013	11/04/2012	x
30.21..0501	1940	Man	22/03/2013	21/02/2013	x
30.21..0501	1946	Man	10/10/2013	01/11/2012	x
30.21..0501	1948	Man	18/02/2013	17/04/2012	x
30.21..0501	1934	Man	27/06/2014	28/08/2013	x
30.21..0501	1951	Vrouw	30/05/2013	07/07/2012	x
30.11.100.0501	1974	Vrouw	12/09/2016	01/07/2016	x
30.11..0501	1954	Man	19/08/2013	19/08/2013	x
30.11..0501	1953	Man	22/10/2014	22/10/2014	x
30.11..0501	1937	Man	14/12/2015	18/11/2015	x
30.21..0501	1946	Vrouw	19/06/2013	11/08/2012	x
30.21..0501	1947	Man	21/02/2013	08/04/2012	x
30.21..0501	1946	Man	09/09/2013	26/09/2012	x
30.11..0501	1964	Man	16/09/2015	28/08/2015	x
30.21..0501	1949	Man	09/03/2015	03/06/2014	x
30.21..0501	1956	Man	08/07/2013	21/12/2012	x
30.11..0501	1950	Vrouw	15/05/2014	10/04/2014	x
30.11..0501	1952	Man	14/07/2015	01/06/2015	x
30.21..0501	1953	Vrouw	26/03/2013	26/06/2012	x
30.11..0501	1944	Vrouw	14/10/2013	23/09/2013	x
30.11.100.0501	1948	Vrouw	16/05/2017	16/05/2017	x
30.21..0501	1940	Man	29/10/2013	30/07/2013	x
30.11..0501	1933	Man	13/07/2015	15/06/2015	x

30.11.100.0501	1961	Man	01/07/2016	21/06/2016	x
30.21..0501	1951	Vrouw	30/03/2015	31/12/2014	x
30.11.100.0501	1951	Man	17/06/2016	10/06/2016	x
30.11..0501	1952	Man	27/01/2014	27/01/2014	x
30.11.100.0501	1941	Man	30/11/2016	16/11/2016	x
30.21..0501	1941	Man	23/09/2013	11/11/2012	x
30.11..0501	1941	Vrouw	09/06/2015	11/05/2015	x
30.21..0501	1932	Man	12/08/2014	27/01/2014	x
30.21..0501	1959	Vrouw	22/09/2015	15/06/2015	x
30.11..0501	1946	Man	15/12/2014	15/12/2014	x
30.21..0501	1934	Man	27/05/2013	07/01/2013	x
30.21..0501	1950	Man	07/07/2014	15/10/2013	x
30.21..0501	1951	Man	17/11/2014	20/12/2013	x
30.11..0501	1956	Vrouw	19/03/2013	19/03/2013	x
30.11..0501	1944	Vrouw	06/10/2014	05/08/2014	x
30.11..0501	1957	Man	27/10/2014	09/09/2014	x
30.21.100.0501	1957	Vrouw	26/09/2016	26/09/2016	x
30.11..0501	1947	Vrouw	05/01/2015	12/11/2014	x
30.11..0501	1948	Vrouw	23/02/2015	23/12/2014	x
30.11.100.0501	1948	Vrouw	22/08/2016	11/07/2016	x
30.21..0501	1945	Vrouw	01/08/2013	20/01/2013	x
30.21..0501	1943	Man	18/11/2014	09/06/2014	x
30.11..0501	1952	Man	30/06/2014	06/05/2014	x

30.21..0501	1954	Man	12/09/2013	09/07/2013	x
30.21..0501	1945	Man	12/12/2013	15/04/2013	x
30.11..0501	1936	Man	08/12/2013	18/11/2013	x
30.21..0501	1969	Vrouw	13/06/2013	25/02/2013	x
30.21..0501	1945	Man	05/08/2013	18/10/2012	x
30.11..0501	1941	Man	05/07/2013	05/07/2013	x
30.11..0501	1956	Vrouw	17/12/2015	05/11/2015	x
30.21..0501	1950	Vrouw	24/01/2013	14/11/2012	x
30.11..0501	1943	Man	06/05/2014	06/05/2014	x
30.11..0501	1955	Vrouw	05/10/2015	18/08/2015	x
30.11..0501	1947	Man	22/12/2014	28/10/2014	x
30.21..0501	1944	Man	21/11/2013	11/01/2013	x
30.21..0501	1948	Vrouw	07/07/2015	23/09/2014	x
30.21..0501	1948	Man	13/04/2015	18/03/2015	x
30.21..0501	1954	Man	26/02/2013	08/07/2012	x
30.21..0501	1948	Man	30/03/2016	22/12/2015	x
30.21..0501	1939	Man	22/09/2015	24/09/2014	NULL
30.21..0501	1939	Vrouw	25/10/2012	09/01/2012	NULL
30.11.100.0501	1948	Man	25/04/2016	02/02/2016	NULL
30.21..0501	1943	Man	03/12/2014	19/02/2014	NULL
30.11.100.0501	1935	Man	02/11/2016	25/10/2016	NULL
30.11..0501	1940	Vrouw	31/08/2015	04/06/2015	NULL