Extracting real-life workflow models from relational data and using these to generate field-based usability testing scenarios at Philips Healthcare

Hoornaar, T.J.

Award date:
2017

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Extracting Real-life Workflow Models from Relational Data and Using These to Generate Field-based Usability Testing Scenarios at Philips Healthcare

Master Thesis

T.J. (Timen) Hoornaar

Supervisors:
dr. ir. N. (Natalia) Sidorova
ir. A. (Angelique) Brosens-Kessels, PDEng *(Philips Healthcare)*
dr. ir. H. (Rik) Eshuis
ir. M. (Maikel) L. van Eck *(Advising member)*

Final version

Eindhoven, July 2017

Abstract

In this thesis, which was written in a project performed at the research & development department of Philips Healthcare, we have used process mining to give insights in the complex workflow in which iXR machines are used and we have proposed a way to include field-data into usability testing scenarios for testing these machines. In order to do this, a few goals needed to be achieved.

First of all, the available data, which was of a relational nature, was translated into artifact-centric process data. This includes defining relevant artifacts and states, transforming the data to these states and linking them to artifacts, and including a process instance in the data.

When the relevant process data was present, an artifact-centric process model was mined using the CSM miner. An overview of relevant practical insights, as well as a thorough review of the CSM miner itself, are presented in this thesis.

Finally, three different measures were defined to combine domain knowledge with data insights to generate usability scenarios. These are transition probabilities (looking at the overall likelihood of occurrence of a given trace, compared with the model), completeness (looking at the completeness of a trace through the model compared to the model) and reachability (guidance while generating a scenario, giving the probability of reaching state X from state Y in an infinite number of runs of the model).

Some of these measures were tested on the real available data and suggestions for inclusion of these measures in an interactive scenario generation tool were given.

Preface

First of all, I would like to thank my supervisors for all the help, guidance and feedback. I think I drove Natalia to the top of her discussion abilities by never taking anything I didn't agree with for granted. Especially the work in chapter 5 would not have been presented in the current status without her (although sometimes merciless) feedback.

Maikel has helped me very well by helping in the daily activities of performing the research and has spent numerous meetings (where the clock somehow ticked quicker than our perception of time) providing feedback and discussing new ideas.

I will not forget the weekly "Meet in the middle" progress meetings over some nice coffees with Angelique, who has always provided the necessary feedback and was always able to provide a good Philips view on my work. These meetings sometimes also took a bit longer than expected, I'm starting to see a pattern here...

I would also like to thank Bart, while still looking at the many models Bart and I drew on the whiteboard. The discussions we had were always very insightful.

I would finally like to thank the third committee member, Rik Eshuis, for the time he spent guarding the grading process and reading the thesis.

Of course I would like to thank my parents and family, who have always supported me in the choices I made during my 6-year study period, even if this choice was sometimes difficult for them (such as moving 3 hours away from my parents' house to study in Groningen). Thank you for all the opportunities you provided me with, and the interest you showed in my (study) life.

Moreover, I would like to thank all my friends. Some of them a remainder from my Dordrecht-period, a lot of new ones in Groningen and some new ones in Eindhoven as well. Without you, I would have had a much more boring life than I have right now.

I won't forget all my colleagues at Philips who have provided mental energy in the form of ping-pong breaks, canteen lunches, car drives, savory times, or just chats.

Finally, a small thank you goes to mr. Frédéric Chopin, whose beautiful music has gotten me through long coding or writing sessions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research described in this thesis has been executed within the IGT systems (Image Guided Therapy) department of Philips Healthcare. This department makes iXR machines, which stands for Interventional X-ray and is a specific type of Interventional Radiology. Interventional Radiology refers to a range of techniques which rely on the use radiological image guidance (X-ray fluoroscopy, ultrasound, computed tomography (CT) or magnetic resonance imaging (MRI)) to precisely target therapy [1]. This type of X-ray systems is used to perform minimally invasive treatments [2] in which it is not necessary to open up a patient for a procedure, but for which it is enough to make an incision in a wrist or a groin through which it should then be possible to navigate an introduction element, such as a catheter.

More specifically, this research has been executed within the Research & Development department of IGT systems, where new machines are designed and tested. The outcomes of this research will therefore be applied in a new development and testing process.

The iXR machines are used in a highly complex setting. The most important difference with other X-ray machines is that they are used in an interventional, surgery-like setting. There is not only a machine, but also staff members (hereafter referred to as "resources"), and most importantly a patient, and they work together quite intensively with a lot of dependencies between them. Please refer to figure 1.1 for an impression of the number of people involved in such a procedure.

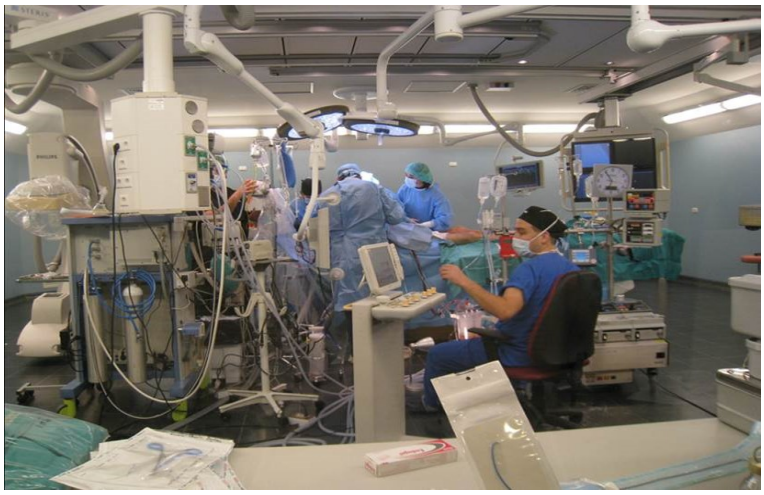

Figure 1.1: Impression of a procedure with the relevant resources, the patient and the machine

The workflow within a specific procedure room is a tight orchestration between the machine, the resources and especially the patient, since every patient is unique, resulting in each patient pro-

cedure having its own possible complications. Philips defines a workflow as follows. "A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information [3]. It can be depicted as a sequence of operations, declared as work of a person or group [4], an organization of staff, or one or more simple or complex mechanisms." [5]. This same definition will be used throughout this thesis.

In order to improve the current machines, it is necessary to know what this workflow looks like and gain more knowledge on the clinical setting of the usage of the machines. This knowledge and understanding of the clinical setting can be used for different purposes. One of these purposes is the substitution of physical validation and verification by virtual validation and verification (hereafter called *virtual testing*): testing new features of a Philips iXR machine before the product is physically built. Furthermore, this knowledge can be used to improve the current ways of physically testing the newly developed machines.

To be able to design and produce good machines that support the doctors in their way of working, it is necessary to know what this way of working looks like in real-life settings. Philips works closely together with actual users to get better insights into this way of working. This is done by observing actual procedures, but also by collaborating with the actual users during the design of new machines. These observations and collaborations are, however, based on small samples and a small subset of the total number of users and usages of the machine. Expanding the reach of these studies to include all users is not feasible, taking the total costs and time this would take into account.

Since every resource has their own point of view and acts from their own perspective, and since every patient procedure and resource is unique in their way of working and complexity, it is very difficult to get a complete view of the workflow when only focusing on a small number of observations and user studies, compared to looking at the total usage. Using a small group of resources and patient procedures to get input for designing and testing new systems will likely show a biased and incomplete view of reality.

One way to include more observations in a feasible way is to look at the data that is generated by the machines in the field. This data is available for the whole lifetime of the machine and for many hospitals around the world. This data is currently not included in the design process. However, being able to use the field data to generate useful insights that can be used for the design process would provide a much better understanding of the workflow and will base the idea that Philips has about this workflow on a much higher number of observations and different procedures.

When it comes to designing new machines, focusing on usability is an important step. Poor usability would not only result in expensive redesigns, poor customer experience, high volumes of complaints, lost user productivity and lost revenue, but international standards require manufacturers of medical devices to follow a systematic usability process [6]. A part of this standard is about testing the usability with representative users.

Usability testing is defined as "evaluating products by testing it with representative users, with the goal of identifying any usability problems, collecting qualitative and quantitative data and determining the participant's satisfaction with the product" [7]. Currently, an observer (usability engineer) will use a scenario-based protocol to tell the participant (usually a doctor or X-ray technician) with a certain role within the scenario which steps to perform. This scenario-based test protocol aims to reproduce a situation as close as possible to the workflow the final product will be used in. The observer checks how well each task in the scenario was performed by the participant.

The central goal of a usability test is to see how intuitively the design works by looking at the interaction between the user and the machine. The goal is not to validate parts of the machine (for example whether pressing button A performs the task that is coupled to pressing this button), this is something that is done by performing functional tests. Usability tests are held in a later

stage of the development and assume that the functional tests have already been performed.

The scenario-based usability protocol is currently based on clinical knowledge present in Philips' employees, and obtained by performing observation studies and collaborating with the users. No field data on the workflow is included.

The usability engineer tries to incorporate incorporate critical and frequently used functions (so-called red routes [6]) of the machine in a scenario. Analyzing these red routes is performed by using all the domain knowledge that is available, but is again only based on domain knowledge and not on the data that is generated in the field. It is therefore not possible to see whether these red routes are really observed as red routes in the field as well, since the data from the field is not used to check this. The real-life data that shows how the machines are used and the whole workflow of a procedure is the information that can help to make the testing scenarios more realistic.

Using field data to get relevant insights into the workflow could reveal ways of working that a usability engineer did not think of. Furthermore, using this data allows for a bigger view of the whole workflow that is based on more observations than currently possible. Deducing the workflow from field data, and using it to create usability protocols, guides usability engineers into the inclusion of relevant functions that are used in the field in the usability test. It also ensures that a task that is given to the participant is performed (in the field) by the same role as the participant has in the scenario. We assume that basing the usability tests on field data will make them more representative and allow for more thorough testing, which will give insight into more usability issues that can be tackled by Philips.

Moreover, new EU regulations also prescribe that some form of risk-tracking should be incorporated into the process and clinical evaluations requirements should be increased [8]. Using field data to enhance the current way of generating testing protocols is one step towards this goal and these regulations are another reason for basing these protocols on field-data.

In order to achieve the inclusion of this real-life information, Philips has collected data from a specific hospital in The Netherlands. Since the workflow is a tight orchestration between the machine, the resources and the patient, three different sources of data are collected. These are:

- Location data: data from a Real Time Location System where each participant in a procedure wears a certain tag that can be tracked to a specific room. This data is logged for one specific hospital in The Netherlands and can be used for the workflow models.
- Manually entered event-data: data collected by using a tablet during actual procedures. Certain procedure steps (events) are recorded when a nurse presses the button for this specific event. This data is logged in the same hospital as the location data.
- Machine log data: data from the actual machines that is uploaded to a Philips database on a daily basis. This data is available for a large portion of the machines throughout the whole world.

However, in order to generalize across multiple hospitals and multiple clinical procedures, data sources from other hospitals must be included as well. This is something that was not possible, given the time that was available for performing this research. Therefore, this will be out of the scope of the research in this thesis.

The available data was collected in a period before we performed the research in this thesis. It was not possible to include any requirements or suggestions for the purpose of this research. Any suggestions on the data might be included in future studies.

1.1 Research goals

The goal of Philips is twofold. Firstly, they want to transform their data into a workflow model to provide data-driven insights in the workflow. These insights would not only be based on the limited number of observations that they are currently based upon, but could be generated from many different hospitals across the world and could be based on many different clinical procedures.

Secondly, Philips wants to use these obtained insights and the field data as a basis for usability testing protocols.

In order to get to these goals, we have come up with a few steps that need to be taken from a technical perspective. To understand the actual workflow, it is necessary to first obtain a graphical model of it, so stakeholders can interpret and discuss them. The data that Philips has contains process data (albeit implicitly, as will be clear from the remainder of this thesis). We want to obtain workflow models, but a realistic view of the process is required. In order to deduce many different insights in a time-efficient and precise way, it is desirable to make this model in an automatic manner. This is where process mining comes in as a good technique [9].

Process mining is a field that allows for automatic discovery of process models that explain the behaviour captured in event data [9]. The input to process mining is data that is structured as a so-called *event log*. The output from process mining is a (visual) model that tries to explain the underlying process of this data as well as possible. This model will serve as the input for an interactive scenario-generation tool, that allows usability engineers to combine their domain knowledge with insights from the data to make a field-based usability scenario.

This section will describe and explain the more technical research goals defined to get to the final goal of Philips.

Process mining requires event data [9], but the data that is available is not structured as such. The very low-level event data that Philips stores in relational databases is not easily interpretable, as it is not related to single activities. This data could be seen as a state in which all the resources (machine parts, but also locations and status of the procedure) are at any given moment in time, rather than sequences of activities. State changes are logged, but the reason for these state changes are not. We can therefore not translate this data to events and it makes sense to keep this state-like nature. This also means we have to apply a process mining approach that incorporates states into the discovered models. It is, however, still necessary to transform the data that is available to data that can be used for these specific process mining techniques.

This brings us to the first research goal, which is defined as follows.

Research goal 1. *Transforming Philips' available data of a relational nature to process data, that can be used as input for process mining techniques.*

As described before, Philips has different data sources that give an insight into the behaviour of different entities within the process. After fulfilling research goal 1, these data sources are transformed into data that can be used for process mining and it should now be possible to extract a usable workflow model. Just as described before, this model should not only give an insight into the behaviour of the machine, but in the whole workflow. This complicates the extraction of a good model, but does not make it impossible. Furthermore, since usability testing scenarios usually try to focus on the mainstream behaviour, the model should also focus on this mainstream behaviour. This brings us to the next research goal.

Research goal 2. *Turning the event data into a workflow model that allows for a good overview of the standard behaviour of the process.*

When the model is created, the question "what does the workflow look like" is answered. However, the main goal of Philips is to derive a usability scenario that reflects the use of the product in the field and includes the tight orchestration between machine, resource, and patient.

From a technical perspective, a usability scenario is not something that is immediately derived from a process / workflow model. Translating this practical requirement to process mining terms would mean that a single execution sequence of the process captured by the model (a trace) must be derived.

In the process of getting from the data to a single trace through the workflow model, certain measures that say something about the quality of the test scenario are defined and are reported to the domain expert. This domain expert should then be able to generate the best field-based

usability scenario that reflects the goal of the specific test the best. This research goal is defined as follows.

Research goal 3. *Deriving a usability scenario (in the form of a single trace) from the data and comparing how well this scenario reflects choices and completeness in this data.*

1.2 Approach

Figure 1.2 graphically represents the approach that is taken in this thesis. The steps that are shown in this figure are described in the following section.

Many of the process discovery techniques available produce a model with a monolithic view [9, 10], whereas process instances may involve multiple interacting process objects or artifacts, each with their own life-cycle [11, 12]. This is also the case in the context of the research performed for this thesis, where the different artifacts consist of the multiple parts of the machine, but also the different resources and the patient. All these artifacts are loosely-coupled, meaning that the artifacts do not have to interact with the same other artifacts. It is, for example, possible that some nurses are working on the preparations for patient C while the other nurses are moving patient B back to his room and the doctor is still working on a procedure of patient A in a different room. However, there is a lot of interaction between artifacts that are working together. For example a specific doctor, that is assisted by a specific nurse to work on a patient in exam room A.

Due to the presence of these artifacts, we use an artifact-centric process mining approach. In order to get from the different data sources of a relational nature to process data that can be used for this artifact-centric process mining, all the data sources are reviewed individually and relevant artifacts are identified. Some artifacts follow easily from the data (for example the location of the different resources), but others need more preprocessing (for example the different parts of the machine). The different data is then linked to one of these artifacts and translated into relevant states to include in the model. When these steps are completed, research goal 1 is fulfilled.

In order to get to a model, the state-like process data that is created in research goal 1 is enriched with the right process instance for the goal of this thesis. This means that process data is created into a full state log that can be used for process mining. The model is mined using the CSM miner [13], allowing for a model that contains the artifacts involved in the process, as well as their mutual interactions, and fulfilling research goal 2.

Finally, to fulfill research goal 3 and to come up with a good field-based usability scenario, we propose a hybrid way in which a usability engineer composes a scenario that is in turn compared to the actual workflow model by looking at the likelihood of seeing such a trace in reality and checking the completeness of the testing scenario. This is done since the domain knowledge that is present in the usability engineers is difficult to capture, and can therefore not be included in the automatic generation of test scenarios. There are many constraints and choices that need to be made in a usability testing scenario and many of these cannot be defined beforehand since they appear trivial to the usability engineers. They only realize these constraints are present when they are contradicted, for example when the model suggests a task that is not expected in the scenario due to another reason.

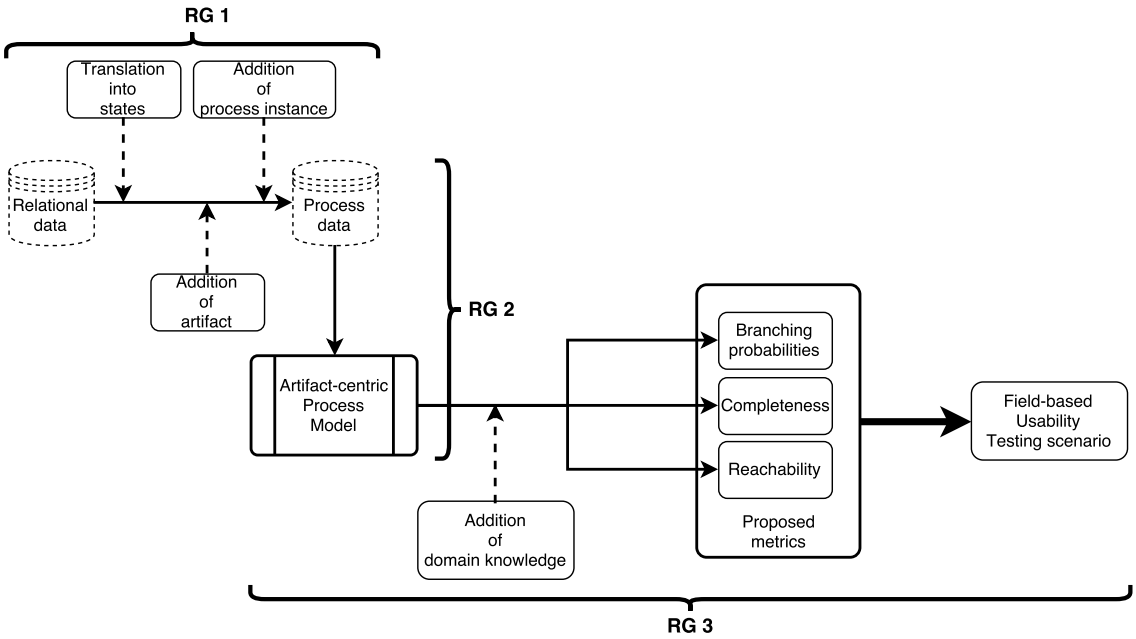

Figure 1.2: Graphical overview of the approach

1.3 Outline

The remainder of this thesis is structured as follows. Chapter 2 gives an overview of the different data sources and their data models. Chapter 3 explains the data preprocessing steps that are taken to clean and process the data to a usable format, after which chapter 4 shows how this cleaned data is used to get to the eventual workflow model, fulfilling research goals 1 and 2. This chapter also explains the models and shows interesting findings, as well as an evaluation of the tool that is used to get to these models (the CSM miner). Finally, chapter 5 fulfills research goal 3 by explaining how a scenario is deducted from the workflow model created before and tested for completeness and transition probabilities. The thesis is concluded with conclusions, limitations and future work in chapters 6 and 7.

Chapter 2

Data Description

This chapter gives an overview of the iXR machines, as introduced in chapter 1, in order to give the reader some background on their complexity. Furthermore, an overview of the different available data sources, as also introduced in chapter 1, and their data models is given. The chapter ends with some conclusions and practical recommendations based on the findings.

2.1 Overview of iXR machines

When looking at an iXR machine from a high perspective, it can be divided into seven main parts. See figure 2.1.

1. C-arm: the part of the system that is used to position the detector and X-ray tube to the right place and angle of the patient.
2. Detector: the actual place where the X-ray image is captured.
3. Table: adjustable table on which the patient is positioned during a procedure.
4. FlexVision: the screen on which physicians, nurses and other medical staff can (re)view images and position the system.
5. Touch Screen Module (TSM): touch screen on which the physicians can choose certain settings, zoom in on certain images and give commands to the system.
6. Control Module (TSO): module that can be used to position the C-arm and adjust the table positions.
7. Foot switch: pedals to enable Fluoroscopy and Exposure (the different dose intensities).

The iXR machines log a lot of data. The most important data refers to the seven parts of the machine as described before. The exact logging of the data is described in section 2.2.3. All the data is stored in relational databases, and is uploaded to a Philips server on a daily basis. This enables Philips to access the data and perform certain analyses on this data. For example, it is possible to predict when certain X-ray tubes will need replacement. This is very useful, since a field service engineer can replace these tubes before they break, which ensures no lab down-time, or scheduled down-time at a convenient time. Furthermore, the error messages in the logs can currently be analyzed to determine the root cause of the error, making the work of field service engineers that have to repair a machine on location a lot easier. The same data that is used for these predictions and field service orders will be used for the creation of the workflow models.

In order to get good images, physicians use different doses. Fluoroscopy uses the least radiation and is usually used to guide wires and catheters through the body. Exposure provides the best images, but also uses the most radiation. These images are usually used for detailed archiving images and comparisons of situations before and after the procedure [15]. There is always a trade-off between image quality and the amount of radiation (dose) that the patient is exposed to.

Figure 2.1: Philips Azurion system[14], with manually annotated numbers

In general, a procedure would start by the nurses preparing the room (making sure the necessary parts are sterile) and collecting the right equipment. The patient is then called to the room and put on the machine table. In the meantime, the doctor (physician / fellow) is called and the exam can start upon arrival of this doctor. This exam starts with the insertion of a guide wire, which acts as a sort of "highway" within the body. This guide wire is guided to the right area (the machine is used for finding the correct vessel and the right way to navigate through the body), after which the diagnostic part of the exam starts. This means that the doctor will investigate the exact steps that need to be taken for the specific patient, all by using the machine to look inside the patient. This is followed by the treatment part, where the actual intervention is performed and for example a stent is placed in a certain part of the body. This stent is placed via a catheter that can be sent over the guide wire. When the intervention is successfully completed, all the catheters and wires are removed and the small incision in the patient is closed. The patient is taken off the table and the doctors can leave. When the patient has left the room, the nurses prepare the room for the next procedure.

2.2 Available data sources

As described in chapter 1, different data sources are used for the creation of the workflow model. Since the model should not only look at the iXR machines, but at the whole workflow, the machine log data is enriched with two other data sources that will be described in the following section. All this data has been collected for research purposes in a hospital in The Netherlands.

The available data was collected in a period before we performed the research in this thesis. It was not possible to include any requirements or suggestions for the purpose of this research. Any suggestions on the data might be included in future studies. The only way to change the data is by preprocessing it, but missing data cannot be obtained anymore.

2.2.1 Location data

The first available data source is location data. For determining the location of the selected resources, data from a Real Time Location System, developed by Centrak [16], has been used.

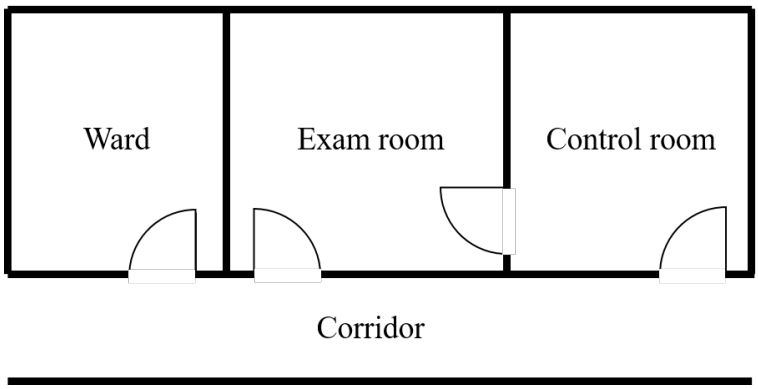

Figure 2.2: An example of a floor plan

0537-.txt - Notepad

File Edit Format View Help

```
1430469841|1601247|174751|false|false|false|false|true|false|1
1430469853|1601247|174751|false|false|false|false|true|false|1
1430469862|1601247|174458|false|false|false|false|true|false|1
1430469865|1601247|174751|false|false|false|false|true|false|1
1430469877|1601247|174751|false|false|false|false|true|false|1
1430469877|455450|174010|false|false|false|false|true|false|1
1430469883|455450|174016|false|false|false|false|true|false|1
1430469886|455450|174010|false|false|false|false|false|false|1
```

Figure 2.3: A sample of the raw location data

Every resource is given a tag that holds his/her role within the procedure. These roles can be physician (the doctor that is formally in charge of the procedure and is either performing the procedure, or supervising a fellow), fellow (a doctor that is still in training, but can perform certain procedures himself) and nurse (a person that assists physicians and fellows during the procedure, but also the role of X-ray technician).

When setting up this system, a few things have to be done. The area to be studied is divided in several regions. An example of this is shown in figure 2.2. These regions are equipped with a beacon (called monitor) that sends out a signal that can be picked up by tags worn by resources. These tags report their location, the ID that is transmitted by the monitor, and the corresponding time stamp every twelve seconds, unless a change occurs. That is, if a new monitor signal is picked up by the tag, the time stamp is reported immediately. Tags will also indicate whether they are in motion at the time of the pulse. The tags also have buttons on them that can be used for different purposes, but these are not used in this thesis.

Figure 2.3 shows a sample of the raw location data input. The first column refers to the epoch time stamp. The second column is the ID of the tag belonging to a certain role, and the third column is the ID of the monitor that this tag picked up. The other columns refer to the motion of the tag, the different buttons that can be pressed and the status of the battery.

The location data is granular up to one specific room. This means that it is possible to see that a certain tag role is in a particular room, but it is not possible to see where the role is located within the room itself. Furthermore, due to anonymity reasons, the different tags are changed between the resources with identical roles. A certain tag can therefore only be linked to a particular role, not to a single person with that role. The time stamps are accurate to 1 seconds.

For preprocessing of the data, please refer to chapter 3.

procedureInternID	PP5761730301
procType	PTA/Stent Bekken-Benen
Patient is besteld:time	2015.08.26 08:12
Patient is besteld:completion type	2
Patient arriveert in wachtkamer:time	2015.08.26 08:28
Patient arriveert in wachtkamer:completion type	1
Patient arriveert in kamer:time	2015.08.26 08:28
Patient arriveert in kamer:completion type	1
Patient ligt op tafel:time	2015.08.26 08:28
Patient ligt op tafel:completion type	1
Arts wordt gebeld/geroepen:time	
Arts wordt gebeld/geroepen:completion type	3
Arts arriveert in kamer:time	2015.08.26 08:44

Figure 2.4: A sample of the raw CRF data

2.2.2 CRF data

The second data source is Case Report Form (CRF) data. This data is collected using a tablet that was present in the control room of the specific hospital. This tablet ran an application in which the administrative responsible resource (usually a nurse) would indicate whether certain activities had taken place. Please refer to figure 2.5 for a screenshot of this application.

The activities that are recorded are highly structured and an overview of the activities can be found in table 2.1, as well as a sample of the raw data in figure 2.4. There are four completion types for these activities:

- Completion type 1: Time stamp is recorded when the event is clicked. Corresponds to tapping the activity name in figure 2.5.
- Completion type 2: Time stamp is entered manually (for an event that was missed in the past, but the time stamp is known). Corresponds to tapping "Handmatig" in figure 2.5 and adding a manual time stamp.
- Completion type 3: Event is skipped, since it was not necessary to perform the event (for example when the doctor was already in the room, so he did not have to be called again). Corresponds to tapping "Niet gebeurd" in figure 2.5. The time is not logged, the completion type is logged. See "Arts wordt gebeld/geroepen" in figure 2.4.
- Completion type 4: Event happened, but the administrator was unable to record the event. Time stamp is not known. Corresponds to tapping "Gemist" in figure 2.5. The time is not logged, the completion type is logged, the same way as "Arts wordt gebeld" is logged in figure 2.4 (only with completion type 4 in stead of 3).

Furthermore, some general information about the procedure is known, which is also based on what the nurse has entered. These are things like the type of procedure, the type of doctor that performed the procedure, whether it was an emergency procedure, what extra equipment was used during the procedure, whether the doctor was late / the procedure had to wait for the doctor, etc. These are not included in the eventual log, since incorporating them into the model would not be feasible within the time-span of this thesis.

The time stamps are accurate to minutes, that means that it cannot be determined at what second an event took place. Moreover, the registration of the events is a secondary task for nurses and is not something that is routine behaviour for their work. Therefore, the time stamps of the activities can only be trusted up to a certain extent. Furthermore, the way the application is built might have a negative effect on how the time stamps are recorded. Please refer to section 2.3 for more information on this issue.

What can be observed from the data in table 2.1 is that all of the activities, except "aanprikken" and "sluit_aanprikpunt" relate to the time before or after the actual procedure. The procedure, in which the machine is used and the patient is lying on the table, happens between "aanprikken" and "sluit_aanprikpunt". This means that no CRF data is available during the period of the actual exam.

Original activity name	English translation
patient_is_besteld	Patient is ordered
patient_arriveert_in_wachtkamer	Patient arrives in waiting room
patient_arriveert_in_kamer	Patient arrives in procedure room
patient_ligt_op_tafel	Patient is lying on the table
arts_wordt_gebeld	Doctor is called
arts_arriveert_in_kamer	Doctor arrives in procedure room
patient_is_klaar_voor_procedure	Patient is ready for the procedure
aanprikken	Start injection
sluit_aanprikpunt	Close injection
gebeld_om_patient_op_te_halen	Called to collect patient
arts_verlaat_onderzoeksruimte	Doctor leaves exam room
arts_verlaat_bedieningsruimte	Doctor leaves control room
patient_van_tafel_naar_bed	Patient is taken off the table and put on a bed
patient_verlaat_onderzoeksruimte	Patient leaves exam room
patient_wordt_overgedragen	Patient is transfered
einde_onderzoek_kies_sluiten	End of examination, choose “close”

Table 2.1: Overview of all CRF Activity labels

2.2.3 Machine log data

The third available data source is the log data from the machines that are used during the procedure. This data is very detailed and low-level, since a lot of things are logged during the procedure. However, Philips has already developed a parser that structures this low-level data into usable tables. An example of this is the exams + acquisitions table that can be found in figure 2.6. The highlighted part in the table shows that there was a three-second acquisition (imaging sequence), belonging to the exam (patient) in the night of April 18th - April 19th. The X-ray setting of this acquisition was “Left Coronary 15 fps” and it deals with a cardio procedure. This data is quite easy to interpret.

Figure 2.7 shows an example of the Geometry events data table, where all the information regarding the positions and geometry values of the machine is logged for every movement. This table reflects the state-like nature of the data as described in research goal 1: the table logs the state in which all the parts of the machine are at the time of a movement and these states can therefore be determined at any given point in time.

All the data that is observed is accurate to seconds. It is difficult to link the machine data to actions taken by resources within the exam room, since the machine is performing a lot of steps for different user actions. Mapping these user actions to the exact steps in the machine log is not trivial. Abstraction and synchronization of these activities and steps are difficult. Furthermore, it is not known who initiated a certain action in the machine. It could be that a nurse pressed a timer for timing how long it takes until a certain person arrives in the room, but it could also be that a doctor started the same timer for interventional purposes, for example to time how long it takes for the contrast fluid to reach the legs of a patient. Another example is the table height, which can be heightened to allow a better ergonomic working position for the doctor, but can also be heightened to increase image quality.

Figure 2.5: Screenshot of the tablet application used to collect the CRF data

ExamStartTimeUTC	ExamEndTimeUTC	AcqStartDateTimeUTC	AcqEndDateTimeUTC	AcqDurationSecs	AppName	ProcedureName	ApplicationGroup
4-18-14 23:38:03	4-19-14 0:59:05	4-19-14 0:23:42	4-19-14 0:23:45	3	Cardiac	Left Coronary 15 fps	Cardio
4-18-14 23:38:03	4-19-14 0:59:05	4-19-14 0:06:07	4-19-14 0:06:16	9	Cardiac	Left Coronary 15 fps	Cardio
4-18-14 23:38:03	4-19-14 0:59:05	4-18-14 23:59:19	4-18-14 23:59:25	6	Cardiac	Left Coronary 15 fps	Cardio
4-18-14 23:38:03	4-19-14 0:59:05	4-18-14 23:59:17	4-18-14 23:59:19	2	Cardiac	Left Coronary 15 fps	Cardio
4-18-14 23:38:03	4-19-14 0:59:05	4-18-14 23:58:06	4-18-14 23:58:13	7	Cardiac	Left Coronary 15 fps	Cardio
4-18-14 23:38:03	4-19-14 0:59:05	4-18-14 23:52:28	4-18-14 23:52:31	3	Cardiac	Right Coronary 15 fps	Cardio
4-18-14 14:07:29	4-18-14 17:39:45	4-18-14 16:18:13	4-18-14 16:18:24	11			
4-18-14 14:07:29	4-18-14 17:39:45	4-18-14 17:01:20	4-18-14 17:01:28	8	Cardiac1 2D	Left Coronary 15 frs	Cardio
4-18-14 14:07:29	4-18-14 17:39:45	4-18-14 16:59:53	4-18-14 16:59:57	4	Cardiac1 2D	Left Coronary 15 frs	Cardio

Figure 2.6: A sample of the Exams + Acquisitions table

EventTimestamp	FrStand_Blo	FrStand_Ca	FrStand_Dt	FrStand_Pr	FrStand_RD	FrStand_Sa	FrStand_ZR	Table_Ht	Table_Lt	Table_Lo	Table_Tl	Table_cr	Table_pv	Table_tia	Table_pva	BrakeEngaged	AbnormalBehaviorLAT	AbnormalBehaviorLNG
01-01-15 10:48:25	2820	-1.7	1190	0	0	PARK	-86.9	821	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:25	2820	-1.7	1190	0	0	PARK	-86.9	821	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	874	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	886	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	883	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:29	2820	0	1195	0	0	PARK	-86.9	908	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:29	2820	0	1195	0	0	PARK	-86.9	908	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE
01-01-15 10:48:29	2820	-0.1	1195	0	0	PARK	-86.9	906	70	1074	0	0	-1.4	HORIZONTAL	WORK	TRUE	FALSE	FALSE

Figure 2.7: A sample of the Geometry events table

2.3 Conclusions and recommendations

This chapter has given an overview of the machines and has described the complex environment in which the machines are used. This setting is complex due to the nature of the procedures: a procedure in which guide wires and other insertion objects are inserted into the body with the help of the images generated by the machines. The tight interaction between machine, patient and staff members is very important for this nature.

Furthermore, this chapter has described the three available data sources and has shown some samples of these sources. The findings of each data source will be discussed below.

Recommendations regarding the location data would deal with the precision: currently the location data is precise to the level of one specific room. This means it is not possible to see where exactly in the room a resource is located. It could be desirable to have this extra precision, especially in the actual exam room where the procedure is performed. This would allow a better insight into what actions are taken within the room and presumably also into which resource, or at least which role, performed which tasks.

Regarding the CRF data, recall the application in figure 2.5. It is possible to configure the default sequence of the activities, but this is not possible from the user interface. This makes it difficult for people with limited programming knowledge to perform such a task.

The current sequential or linear ordering of activities suggests a certain fixed sequential execution of the tasks. It is possible to select another activity, but the application gives strong suggestions on the activity that is next in line. This could be seen as a potential flaw of the application, since it is more difficult for users to select the actual activity that was executed when this was not the same as the "default" next in line activity. It is likely that users make mistakes in the time stamps because of this. Changing the application to allow for more freedom by the user would likely result in less mistakes in the recording of the data.

Finally, looking at the machine log data, the main concern is the amount of low-level activities that are recorded. It is very difficult to interpret this data for process mining purposes, especially since it is not known who performs certain changes in the state of the machine. It could be that a nurse pressed a button (resulting in a state change of the machine) for some non-interventional purpose, but it could also be that a doctor pressed the same button for interventional purposes. This also makes it very difficult to get to the underlying nature and reasoning of performing certain activities.

When it comes to the appropriateness and potential for process mining, a few observations can be made.

Including location data in the final model will provide an insight into the location of specific resources when certain activities are performed within the room. Since it is not possible to see where a user action originated from in the machine log data, it might be possible to link the location artifacts to other artifacts and find correlations between them. This is something that has not been done in this thesis, but could be included in further research.

Including CRF data in the model allows for better insights into the status of the procedure. This data shows, for example, whether patients have been called to the room, whether they are put on the bed, and whether an exam is going on. This data is also used for determining the process instance, as will be discussed in chapter 4. The data has some quality issues, but this is not presumed to be a very big problem for the purpose of this research.

Finally, including the machine log is an obvious step since this is one of the central artifacts during a procedure. It provides insights into generating images, geometry movements of the machine, status of a procedure etc. This data needs some processing steps in order to make use of the high potential, but the advantage it brings outweighs these efforts.

Chapter 3

Creating Process Data

The data, as described in chapter 2, is the raw data input that is handled in the process mining approach. In order to apply process mining on this data, the data must first be preprocessed and translated into something that can be interpreted by the process mining algorithm.

The CRF data is already structured as process data, but the other data sources need to be transformed from their relational nature to this process data format. This chapter looks into the preprocessing of this data, which means cleaning the data, but also translating the data into states that can be incorporated in the state log. This last step is especially important for the machine log data, which is difficult to interpret due to its very high amount of low-level activities and state changes.

3.1 Challenges

[9] defines a process as "a collection of activities such that the life-cycle of a single instance is described. Event logs contain data related to a single process. Each individual event must refer to a single process instance and events can be related to some activity".

This definition differs from the data that is currently available at Philips (apart from the CRF data, this data is already in an event log format). First of all, the data does not contain activities, but it mostly contains states: the position of the table is known at any point in time, as is the location of a specific resource. Secondly, no reference to single process instances is made. The state of resources and the machine is known, but the process instance that is currently being handled is unknown.

The CRF data is already in an event log format, but the location and machine data need to be transformed to such a process-oriented structure.

The location data is one long stream of locations, so it is known for every resource where it is at any point in time. One could look at this as a state of each resource at any given time. However, this is not process data yet. In order to get to process data, the addition of a specific process instance and the translation into a state-like data source should be performed.

The machine data is quite alike. This data source is, however, much bigger and something is done to make sure the eventual model is still interpretable by humans. Mining the model without doing something about this results in a spaghetti-like diagram, due to the complexity of the machine. Just like the location data, the state-like nature can be kept, but some discretization must take place to limit the number of possible states. Furthermore, a specific reference to a process instance should be added to this data.

Many of the process discovery techniques available produce a model with a monolithic view

[9, 10], whereas process instances may involve multiple interacting process objects or artifacts, each with their own life-cycle [11, 12]. Recent developments allow for the discovery of such artifacts [10, 12, 17] and will discover these as an individual model. Additional mining of the interaction between these artifacts and the visualization of these interactions allow process analysts to better understand models that contain different artifacts with their own life-cycles [13].

This type of process mining is very suitable for the kind of challenges that were described before. The sensor data in the machine all has its independent life-cycle model, but since they are all related to the same machine their interactions are limited. Interactions between these artifacts are very important, since they capture the whole behaviour of the system. We will therefore use an artifact-centric process mining approach.

3.2 Related work

Although underlying relational databases are loaded with data, there are no explicit references to events, cases, and activities. Instead, there are tables containing records and these tables are connected through key relationships. Hence, the challenge is to convert tables and records into event logs. Obviously, this cannot be done in an automated manner [18].

A difference between [18] and our situation is that [18] takes the viewpoint that the database reflects the current state of one or more process and defines all changes of the database to be events. We, on the other hand, take the state-like nature of the process into account and tries to capture these different states in the event log, where (due to the artifact-centric approach) each state change in one artifact does not necessarily have to mean a change in other artifacts as well. In other words: process mining usually works around the idea of events (that happen at a certain moment in time), whereas we use data that shows a certain state in which (part of) the entities are during a longer period of time.

Inherent to the choice of artifact-centric process mining, each artifact has its own artifact instances, which allows for the creation of separate artifact types and models. This yields different models of different sub-processes. These sub-processes interact with each other, and it is the combination of the sub-processes and the interaction that models the behaviour of the whole system.

According to [18], all events require a single case notion. An exception to this are proclets and artifacts; they can contain multiple case notions. Each trace in an event log describes the life-cycle of a particular case in terms of the activities executed. This is also the case for the process in our case: even though the data is divided into separate artifacts (each of them with their own life cycle), all of these describe the same single case.

3.2.1 Location data

Incorporating a constant stream of location data into a process model is something that has been done before. [19] transforms sensor data (mainly data from a RTLS) to process instances by using interaction mining. A certain amount of process knowledge is a prerequisite. Since Philips wants to make a workflow model to get insight into the process, this knowledge is not sufficiently present. Furthermore, the paper assumes a 1:1 relation for co-locating interactions, meaning that a certain activity and a certain event are highly correlated. This is also not the case for the data used in this thesis, since our locations are quite sparse, but the activities that can occur in these locations are very broad. Hence, knowing with a certain degree of certainty that an activity occurred when an actor was at a specific location is not possible.

Other authors have focused on mining location data as only data source, see for example [20, 21, 22], and draw conclusions on the location data only, whereas the approach described in this thesis is to enrich other data sources with this location data.

Since no approach in related work can be used, we have decided to incorporate the same state-like nature as applied to the machine data to the location data and come up with our own approach. This approach will be described in section 3.4.

3.2.2 CRF data

As described in chapter 2, the CRF data is recorded in an event log format, but some time stamp values are missing. The issue of missing data must be addressed since ignoring this problem can introduce bias into the models being evaluated and lead to inaccurate data mining conclusions [23]. Numerous data mining techniques exist for handling missing values. Some of them focus on missing values in general [24, 25, 26], whereas others focus on missing time values and predicting time series [27]. A way to partially address the problem of time stamps in general is to "guess" the order based on domain knowledge or frequent patterns across days [28].

We expect that all these approaches could have worked for our data. However, due to time constraints, we decided not to include any of the proposed preprocessing techniques. Please refer to section 3.5.1 for more information.

3.2.3 Machine log data

There are many different approaches to handling low-level event data. [29] use activity mining to cluster low-level events into higher-level logs. This approach requires configuring the algorithm with appropriate parameters that require domain knowledge. Furthermore, this approach neglects the state-like nature of the data that is used in our case. The same goes for [30], who apply event abstraction using behavioral activity patterns, but with the same limitations. The articles by [29] and [30] could have been used in our case, but implementing this would require more time that was not available. Therefore, these could be included in potential future work, which would take away the manual conversion step that we performed and replace this with an automatic step. This could possibly give better results, but since this was not tested we don't have evidence that proves this.

When applying machine learning to perform activity recognition, the quality of annotated data is very important for the performance of learning algorithms [31]. Algorithms can either be trained by using large amounts of prerequisite knowledge, or by applying domain knowledge to relate the sensor data to activities [32]. Since no translation from low-level events to higher-level activities exists for the available machine data, activity recognition is not suitable for our problem.

A search for unsupervised activity recognition (where no reference data is required) yielded no satisfactory results either. [33] proposes an unsupervised approach to activity recognition, but they use web mining to find relevant patterns on the web. This cannot be applied to our data, since the activities are machine-specific and are too specific to be published online or taken from other sources. Furthermore, [34] provide a way to apply activity recognition with only sparse labeled data. However, since the use of the machine and the logging of the data is not known to the fullest extent, it is expected that this would result in a happy-flow of activities only, as most of the "uncommon" activities are not labeled when asking a domain expert.

To summarize, the search for related work did not result in a successful approach that can be used in this thesis. Therefore, we came up with our own approach, which will be described in section 3.6.

3.3 Definition of artifacts

The artifact-centric approach, as described in the beginning of this chapter, allows for modeling the workflow as different artifacts. Determining these artifacts is an important step in preprocessing the data. Artifacts are interacting process objects that each have their own life-cycle [11, 12]. We are therefore looking for interesting "parts" of the data that could compose their own model, but also play an important role in understanding the system and process as a whole.

When looking at the process that is under study, that of a procedure within a procedure room, three main types of artifacts are expected. These are the machine, the people that use this machine, and the patient for whom this machine is used. Since the machine is composed of several

parts, looking at these individual parts would be a nice starting point for defining artifacts. The people that interact with the machine are the resources that are involved within the procedure.

The machine division in its separate parts reveals different machine artifacts. The division in the separate parts, as well as the inclusion of extra necessary information, has been done by talking to a domain expert and seeing what this expert would want to include in the model.

The same approach has been taken for incorporating the people in the model. The different roles have been defined, and each of these roles has been given some attributes based on domain knowledge.

This approach leads to the different potential artifacts as shown in table 3.1.

After comparing the desired artifacts to the available data, the conclusion was drawn that a lot of desired behaviour is not logged, or not logged to the extend that we desire. Table 3.2 shows which of the artifacts were included in the actual event log and the reason for not including certain others.

Artifact group	Artifact	Possible states
Machine	Table height + movements	{working position, bed position, ideal imaging position, moving up-/down, panning (searching)}
	Table position	{pivoted, not pivoted, tilted, not tilted, cradled, not cradled}
	X-ray tube	{on, off, unavailable}
	Radiationdose	{number of images, radiation used for images}
	Radiationtype	{low fluoro, medium fluoro, normal fluoro, exposure (strength etc. depending on EPX settings)}
	Location of detector	{head, torso, abdomen, upper arm, lower arm, upper leg, lower leg}
	Number of TSO's + TSM's	{one, more than one}
	Ultrasound present?	{yes, no}
	Anesthetics present?	{yes, no}
	Extra sterile table present?	{yes, no}
	Hemo values	{blood pressure and heartbeat values}
	Layout of the monitor (exam + control room)	{different screens, position of screens }
People	Patient location	{on a bed, on the table, fallen}
	Patient status	{in procedure, not in procedure}
	Patient type	{scheduled, entering with emergency, emergency on table}
	Nurse location	{exam room, control room, corridor, hallway, storage}
	Nurse sterile status	{sterile, not sterile}
	Nurse working type	{normal nurse, X-ray technician, "omloop", administration, not working}
	Fellow location	{exam room, control room, corridor, hallway, storage}
	Fellow sterile status	{sterile, not sterile}
	Fellow working type	{off, performing procedure, observing physician}
	Physician location	{exam room, control room, corridor, hallway, storage}
	Physician sterile status	{sterile, not sterile}
	Physician working type	{off, performing procedure, supervising fellow}

Table 3.1: Potential artifacts for the workflow data

Artifact group	Artifact	Data included
Machine	Table height + movements	No, abstracting the table geometry to for example the specific working height of a doctor is difficult due to the anonymity of the data.
	Table position	Yes
	X-ray tube	No, this information was not deemed necessary for an initial version of the model
	Radiationdose	No, abstraction amongst this dimension is very difficult (according to a radiation expert) and should be further investigated
	Radiationtype	Partly, only Fluoroscopy versus Exposure was included, not the intensity
	Location of detector	Not exact, abstraction depends on a large combination of values and should be further investigated. EPX settings are included, which give an indication of the location of the detector
	Number of TSO's + TSM's	No, this is unknown for the setup
	Ultrasound present?	No, as the machine log is only based on the actual iXR machine, it is difficult to include ultrasound information (which is an external machine), unless it is connected to the iXR machine. This machine can also work without the iXR machine.
	Anesthetics present?	No, same reason as ultrasound
	Extra sterile table present?	No, these values are not logged
	Hemo values	No, these values are not logged
	Layout of the monitor (exam + control room)	No, these values are not logged
People	Patient location	No, patient information is not logged
	Patient status	No, these values are not logged
	Patient type	No, these values are not logged
	Nurse location	Yes
	Nurse sterile status	No, these values are not logged
	Nurse working type	No, these values are not logged
	Fellow location	Yes
	Fellow sterile status	No, these values are not logged
	Fellow working type	No, these values are not logged
	Physician location	Yes
	Physician sterile status	No, these values are not logged
	Physician working type	No, these values are not logged

Table 3.2: Inclusion of the artifacts in the actual event log

3.4 Location data

Since the artifacts are now known, it is possible to look at the preprocessing and inclusion of the different data sources into the model. The location data will be cleaned with the help of a tool that has been developed by Philips.

3.4.1 Cleaning the location data

Since the tags and monitors are not 100% accurate, the data that is generated needs to be cleaned. Philips has developed a cleaning program, which is re-used and enhanced in this thesis. This program cleans the data in three different ways:

- Remove hopping events (reused from Philips)
- Replace unknown locations (zeroes, reused from Philips)
- Indicate when the tag is out of the monitored area (extension made by us)

These three cleaning procedures will be explained in the following sections.

Remove hopping events

When a tag is constantly switching between two monitors, a lot of events are generated at the same time, which is called hopping. This can happen when a tag is near a boundary between two zones, which means it sees both monitors alternatingly during the two halves of each detection period. This results in a sequence of rapidly alternating location reports, where the period between two reports is typically 0 or 3 seconds but occasionally longer (up to 12 seconds) [35]. Since hopping events rarely occur on their own, a hopping sequence is defined as follows: *Please note that this definition is the definition Philips used to implement this in the tool. The parameters are therefore fixed and we cannot change their values.*

Definition 1. *A hopping sequence is a sequence of alternating location readings such that the length of the readings $\geq$ 4 (A-B-A-B), and a time difference between each reading <13 seconds.*

Not all these hopping sequences are potentially impactful.

Definition 2. *Let a static hop be a hop where the tag was motionless during the hopping. Then a hopping sequence is considered as innocent hopping sequence when it consists of predominantly static hops, that is <25% of the events had motion.*

If a hopping sequence fulfills definition 2, it is not removed from the data. If it does not fulfill definition 2, it is marked as a harmful hopping sequence and the reported monitors between the start and end of the hopping sequence are replaced with 0.

Replace unknown locations / 0-events

If the system cannot identify the location of a tag, its location will be indicated as "0". This happens when the tag either does not receive any monitor signal, or when it receives two different monitor events at the same time (without hopping occurring). Not getting a signal could be because the tag is out of the monitored area (which is dealt with in more detail in section 3.4.1) or because the tag is covered by other equipment or not within the reach of a specific monitor in the same room.

One of the inputs to the cleaning tool that Philips has developed is an annotated map of the area that is under study. This map can be interpreted by the tool. Whenever a 0-event occurs, the tool looks at the shortest path from the previous known location to the next known location and interpolates the 0-event with the necessary locations according to this shortest path calculation. Dijkstra's algorithm is used for calculating this shortest path [35].

Indicate out of the monitored area

The current toolkit does not take into account whether resources are out of the monitored area (for example because of a break) or not. If resources are out of the monitored area, 0-events will occur. In the configuration Philips initially had, this would be interpreted as an unknown location that needs to be interpolated, in the sense of section 3.4.1.

However, this is not a behaviour that is desirable. It could be that a resource is on a break and the last known location was the exam room. The whole period of his break will be interpolated as if this resource was always in the exam room. This gives a scattered view of the data and we have therefore come up with a simple measure to exclude these events.

The minimum communication frequency of active tags is 5 minutes. When lying still for some time, the tags go to sleep mode, in which they also report their location once per 5 minutes [35]. If the tag is not in the monitored area, it is not able to send its location and this means there will be a gap in the timing of ≥ 5 minutes since the last reported location.

This threshold of 5 minutes is used to determine the tags that are out of the monitored area: if the last reported location was a 0, and the last recorded 0 time stamp was ≥ 5 minutes ago, the tag will be marked as "Out of the monitored area" until a non-0 location is recorded.

3.5 CRF Data

The CRF data is already very close to a usable event log for our purposes. However, this data has some data quality issues due to the nature of the collection. This section looks into these issues and their possible solutions. It is, however, not possible to use one of these solutions due to timing constraints, and therefore this data is left unchanged.

3.5.1 Cleaning the CRF data

Missing values must be replaced with extreme sensitivity for not disturbing the patterns in the data that already exist [26] and there is no one best way of handling missing time stamps. An approach would likely be related to predicting time series, as suggested by [27], since this is the closest to our problem. However, since developing and applying a good algorithm for our specific data would possibly require too much work, we decided not to interfere with the data and accept it as it is right now. This means that both skipped and missed CRF events are handled as if they did not occur, acknowledging the limitations this might imply. Our focus is to prove the feasibility and approach for the specific goals that have been described before, the perfect data quality is something that can be achieved in future research or future applications of the outcomes.

The specific patient reference is known for each case, due to the event log nature of the CRF data. The only operation that is needed to transfer the data from the current format to an actual event log are very trivial linking activities and therefore not based on related work.

3.6 Machine log data

The machine log contains a lot of very detailed and low-level information, which is way too complex for the purpose of this thesis. Mining a model of this data results in a big spaghetti-like diagram. It is therefore desirable to divide this data into separate parts, which will be done based on the parts of the machine, which can then be used as separate artifacts. This is described in detail before in section 3.3.

3.6.1 Incorporating states in the event log

After combining the desired artifacts with the available data, as described in tables 3.1 and 3.2, the states as shown in table 3.3 are defined. The following sections go into more detail on obtaining

these states and the reasoning behind including them in the event log.

Artifact	Possible states
Tilt	(Positive tilt|Negative tilt|Not tilted)
Pivot	(Positive pivot|Negative pivot|Not pivoted)
Cradle	(Positive cradle|Negative cradle|No cradle)
Imaging state	(Fluoroscopy|Exposure|No imaging)
Machinestate	(Working|Idle|Imaging)
EPX Setting	All the possible EPX settings, e.g. Iliac/Pelvis, One upper leg, Two legs.

Table 3.3: Overview of the machine log states

Tilt state

Tilting the table means that the feet of the patient are tilted downwards or upwards. Tilting could serve multiple purposes, but one of the most important ones is that tilting with the feet towards the ground will result in more blood flowing to the feet of the patient, and tilting with the head to the ground will result in more blood flowing to the head of the patient. This could influence the flow in gravity-oriented procedures, for example to quickly get contrast fluid to the feet of the patient. Please refer to figure 3.1 for a side-view example.

Incorporating tilt in the model would of course say something about the type of procedure, but it could also give domain experts an indication of the issues in flow that this specific patient had. Using tilt is usually connected to a flow problem in the arteries of the patient. The states that table tilt can be in are:

- Positive tilt: the table is tilted upwards
- Negative tilt: the table is tilted downwards
- Not tilted: the table is not tilted and is in a straight position

The machine log is translated into one of these three states by iterating through all the geometry movements in the machine log and seeing whether the value for tilt is positive, negative, or zero.

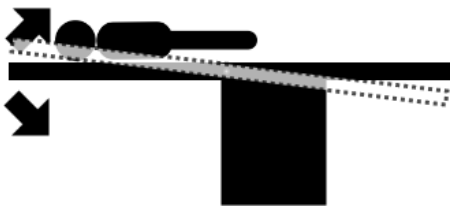

Figure 3.1: Side view from a table showing the tilt state

Pivot state

Pivoting the table refers to pushing the table outwards, so the head of the patient is not in line with the C-Arm anymore, but the arm of the patient is. Please refer to figure 3.2 for a top-view example. Since the C-Arm can only move in parallel to the table in default position, it is difficult to reach an arm that points outwards. This is especially used for radial access, where the insertion object is not inserted through the groin (femoral access), but through the wrist [36]. This is where table pivot comes in as a good feature. Incorporating this state in the model could say something about the type of access that is used (either radial or femoral) and the part of the body that is investigated by the physician. The states that table pivot can be in are:

- Positive pivot: the table is pivoted clockwise
- Negative pivot: the table is pivoted counter-clockwise
- Not pivoted: the table is not pivoted and is in a straight position

The machine log is translated into one of these three states by iterating through all the geometry movements in the machine log and seeing whether the value for pivot is positive, negative, or zero.

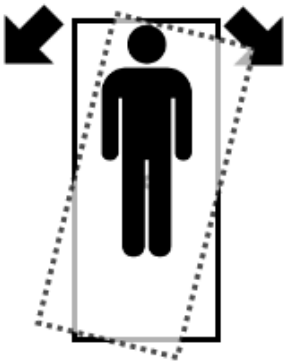

Figure 3.2: Top view from a table showing the pivot state

Cradle state

When cradle is enabled, the table's level is not parallel to the floor, but is twisted. Please refer to figure 3.3 for an example, as seen from the front of the machine. This feature is especially useful when a heavier patient is transferred from the bed to the table of the machine. It might be hard to lift this patient, and cradle could be used to assist the nurses in getting the patient on the table. Incorporating this state in the model could say something about the type of patients that are on the table (heavy vs. light). The states that table cradle can be in are:

- Positive cradle: the table is cradled clockwise
- Negative cradle: the table is cradled counter-clockwise
- No cradle: the table is not cradled and is in a parallel position with regard to the floor

Just as the pivot state, the machine log is translated into these states by iterating through the geometry movements and seeing whether the value for cradle is positive, negative, or zero.

Figure 3.3: Front view from a table showing the cradle state

Imaging state

As described in section 2.1, there are two intensities of X-ray. Since it is proven that radiation is bad for people (see for example the discussion in [37]), physicians have to minimize the amount of radiation that patients receive. There is always a trade-off between the amount of radiation and the image quality. Incorporating these choices in the model provide a good insight into the usage of the machine (since Fluoroscopy is used for different purposes than Exposure) and the amount of radiation patients are generally exposed to during a procedure. The states that Imaging can be in are:

- Fluoroscopy: the lowest intensity setting, usually used for guiding the insertion object through the body
- Exposure: the highest intensity setting, usually used for reporting purposes and comparing the pre- and post situations of a procedure
- No imaging: there is no imaging at that time and therefore no Fluoroscopy or Exposure pedal is pressed

The acquisitions table in the machine log shows all the different series of imaging chains. The imaging state is saved in this table as well and it is therefore a 1:1 mapping to get these states from the acquisition data.

Machinestate

The machinestate of the machine refers to the activity before, during, and after a procedure. Before the patient is entering the room, the machine is prepared for this patient. This involves some activities. When the patient is in the room, the actual procedure starts which of course involves a lot of activities with the machine as well. During this procedure, we distinguish between activities that are performed during an acquisition (making an image chain) and around these acquisitions. Then, after a patient has left the room, finishing the procedure, as well as making preparations for the next procedure will start. The machinestate of the machine can be in the following states:

- Idle: an exam has not started yet, so this refers to the preparation / finishing activities
- Working: an exam has started, but no acquisition is taking place
- Imaging: an exam has started and an acquisition is taking place

Please refer to figure 3.4 for a graphical overview of the machinestate and the position in the procedure.

The machinestate could be one of the most important states in the model, since incorporating it provides insights in the different activities that occur when a procedure / exam is taking place, when images are being taken and when preparations or finalizations are taking place. There could be major differences between the number of people that are present in these states, as well as the type of roles that are present and the activities that occur with the machine. It is, for example, not expected that radiation is used during the "Idle" state, since that means radiation is used without the purpose of treating a patient.

The acquisitions table in the machine log saves the start- and end times of an exam. The start time is saved as the time when the patient information (type of patient, previous images etc.) is downloaded into the machine. The end time of an exam is saved as the time when the new images and the report of the procedure for the specific patient are uploaded into the hospital information system. These times could be misleading, since patient data can be downloaded into the machine both long before the procedure or when the patient is already lying on the table. Likewise, the new patient data can be uploaded to the hospital information system when the nurses are still busy performing the final roundup activities to the patient, as well as long after the patient has already left the room (this for example happens when the case is complicated and the physician still has to look through the images and annotate them for the report). This is a limitation of our approach, however it is not possible to derive these states in any other way than by using this data.

The start time of an acquisition is the time the Fluoroscopy / Exposure pedal is pressed (starting the images), the stop time of an acquisition is the time this specific pedal is released (stopping the images). The start time of the exams is immediately derived from the acquisitions table in the machine log data. The three states are derived by iterating through the data. Whenever a procedure starts, the state "working" is saved. If an imaging chain occurs, "working" is changed to "imaging" and back to "working" again when the imaging chain is completed. When the exam stopping time is reached, the state is changed back to "idle".

Please note that it is not possible to distinguish between finishing time of the previous patient and preparation time for the next patient. These two types are both reflected in the state "idle".

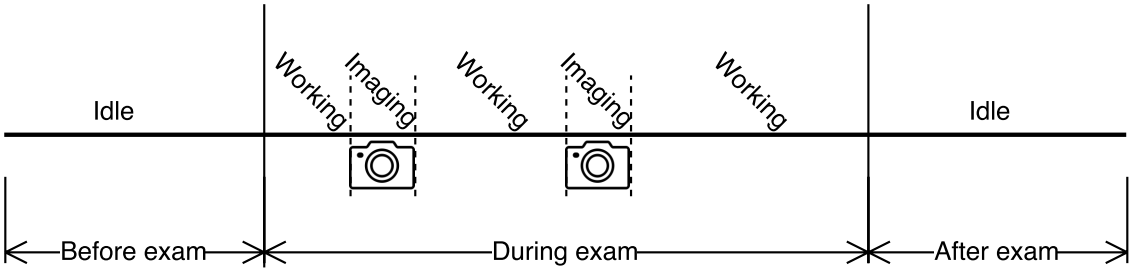

Figure 3.4: Overview of the "Machinestate" artifact. The x-axis represents the time (not to scale)

EPX Setting

The EPX setting refers to the setting of the machine when images are taken. A certain EPX setting optimizes the imaging parameters to provide the best image possible for certain parts of the body. For example, an EPX setting for a leg provides fundamentally different gray values, image intensity zones etc. than that of a heart.

The EPX setting tells something about the area that is under interest. However, it should be noted that not all physicians adapt the EPX settings to the right area under interest. An example of this could be a peripheral lower leg case where the physician uses an incision in the groin of the patient. To find the right artery in the groin, the starting EPX setting could be "Iliac / Pelvis". After the incision is made, the physician would move the detector of the machine downwards, first crossing the upper legs and then the lower legs. Since the upper legs are not very important for a lower leg case, it could very well be that the physician would only change the EPX setting from "Iliac / Pelvis" to "Lower leg", leaving the "Upper leg" setting out.

Besides this given limitation, incorporating the EPX setting in the model provides big insights into the parts of the body that are under investigation. Also, seeing that a heart setting is chosen for a lower leg case could provide a potential inconsistency in the system.

The EPX settings are again saved in the acquisition table. For each acquisition, the specific setting can be derived. The frames per second (fps) are also included in the EPX field in the database. Since these intensities are not relevant for determining the area under interest (the location of the detector), this value is removed. So "Lower legs 1fps" gets the same EPX state as "Lower legs 2fps".

3.7 Conclusions and recommendations

After the different data sources have been cleaned in the specific ways described in this chapter, they have been linked to different artifacts and incorporated in the event log. This brings an almost-complete event log: the only thing that is missing is a reference to a process instance, which will be discussed in chapter 4. However, it should now be possible to use the relational data of Philips for process mining purposes due to the translation to an event log format, independent

of the choice for a particular process instance. This chapter is therefore considered as a very important chapter when it comes to process mining, since it would not be possible to apply the necessary techniques without the preprocessing and conversion steps described in this chapter.

Even though this chapter has reported some important steps, there are still possibilities to improve the quality of the data and therefore the actual event log and model that will be created from this data.

Table 3.1 shows the desired artifacts, whereas table 3.2 shows which artifacts were actually included in the final event log. There is a huge discrepancy between these two tables, and the reason for this discrepancy is described in table 3.2 as well. Finding a way to include the desired data in the event log would presumably increase the quality of the eventual workflow model very much, and it is therefore highly recommended to look at this inclusion for future versions of the model.

The location data is preprocessed in a way that was specifically designed for this data and we believe that further preprocessing of this specific data will not result in a better model. Of course, including more precision would result in different results, but this has already been described in chapter 2.

When it comes to the machine log data, the articles by [29] and [30] could have been used in our research if more time would have been available. Looking into these articles and trying to apply the techniques proposed here could result in better event logs, and thus better models. Including their techniques, that are mainly aimed at finding automated ways of getting from the low-level data to higher-level data, would remove a manual definition step of the different states in the machine log data, and would therefore remove potential human errors. Of course, these techniques would bring other challenges as well, but further investigation might prove helpful.

As described in section 3.5.1, it was not feasible to clean the CRF data appropriately during the available time for this thesis. It is, however, recommended to perform some of the suggested cleaning methods to ensure that the quality of the data is high enough.

Chapter 4

From Process Data to Model

The available data, as described in chapter 2, has been cleaned, as described in chapter 3, and translated to state changes with time stamps. However, an event log consists of more than only state changes and time stamps. This chapter describes one of the most important aspects that make process mining a success for our data: the definition of a process instance. Multiple options are considered, in which the approach is chosen to start with the smallest possible process instance and generally increase the time-span of this instance to generate all the possible process instances. These are then evaluated, which results in choosing the process instance of "procedure". This is also the process instance that is logged in the CRF data, so this approach is also feasible from the point of the available data.

Furthermore, this chapter will also describe how we got from a log (with process instance) to an actual model and evaluates the insights these models generate on a structured basis.

4.1 Process instance

Process models always describe life cycles of instances. The model is instantiated once for each case. The notion of process instances is made explicit in process-aware information systems, but in most other systems the instance notion is implicit [18]. In order to generalize behaviour among these different process instances, the process instance is needed [9]. What is the process that needs to be modeled? If that is known, the best means to abstract across these process instances naturally follows.

Currently, only the CRF data refers to some instance (a specific patient procedure, which is not chosen as the actual process instance yet), the location- and machine data do not, due to their relational structure.

4.1.1 Related work

The related work on process mining and process mining methodologies that was found does not explicitly tackle the problem of choosing a correct process instance. This lack of related work is underwritten in the relatively recent work [38], in which a process mining methodology is developed. The authors acknowledge that there should be more guidance for process analysts, for example with choosing a process instance.

[39] also underwrites this problem and states that one of three main problems when dealing with sensor data is the segmentation of the logs. More specifically, the authors state that sensor logs do usually not contain any information on which user caused a certain sensor record. No specific solution has been defined for this problem.

A way to include the process instance in a sensor-oriented event log is proposed by [19]. However, the same limitations as in section 3.2.1 hold: the paper only focuses on co-locations, which are not trivial in our case. Furthermore, the paper "assumes the existence of a case function

that returns the case entities from a set of entities". This case function is exactly what needs to be found for our data.

The idea of process cubes ([40]) is somewhat related to process instances. Process cubes allow for multidimensional process mining, in which multiple process instances are included in a specific format. This allows for different views on the same process, for example looking at different type of orders in an order handling process. This approach would allow us to define multiple process instances and see which model fits the needs the best, but the process instances would still need to be defined manually. Furthermore, process cubes have not yet been used in an artifact-centric approach and the artifact-centric approach would require a separate cube to be defined for each artifact, with potential synchronization issues as a result.

4.1.2 Application to the available data

Due to the lack of a formal approach in related work, we come up with our own way of including the process instance in the logs. This approach is described in the following section.

Choosing the process instance for the log

The process at Philips can be described in numerous ways and from numerous perspectives. In order to make an overview of different possible process instances, we started with the smallest possible process instance and expanded the time-span of this notion with every step. The results of this analysis can be found in table 4.1.

Currently, a usability scenario starts when the doctor enters the exam room (and the patient is already on the table) and stops when the doctor leaves the exam room. The process instance of one exam would then be the most suitable. However, Philips wants to move to a story-like scenario in which a scenario entails the entering of a doctor + patient, treatment of that patient and leaving of the patient + doctor. The process instance that suits this goal the best is therefore that of one procedure.

The two problems of this process instance, as described in table 4.1, can be overcome as follows.

- Overlap in the data: if an event is relevant for multiple cases, it should be replicated when creating event logs [18]. Therefore, the data that could apply to both a new and an old patient can be replicated, with the respective case identifiers of both procedures.
- Boundary determination: with the help of CRF data, this problem can be overcome. This requires the availability of CRF data. This is not a problem for our case, since CRF data is available. However, it would be a problem for different hospitals or to incorporate new data in the model.

Process instance	**Pros**	**Cons**
1 acquisition ("imaging" in "machinestate" artifact)	- Small timeframes, so very detailed information of separate acquisitions - An exam consists of many individual acquisitions, so the available data is already a lot when only a few patients are recorded - Data quality is good, since it is exactly known when an acquisition starts and ends	- Not looking beyond acquisitions, whereas most of the activities happen around the acquisitions - An acquisition is a relatively short part of an exam, so the view is quite limited
1 exam (acquisition + "working" in "machinestate" artifact)	- Only the time when the patient is actually being treated is considered - Data quality is less good, since the start and end of an exam are somewhat prone to interpretation	- Usage of the machine outside the exam of the patient is discarded.
1 procedure (exam + "idle" in "machinestate" artifact)	- Hospitals are usually patient + appointment-centric, so looking at the treatment of one patient on a particular day makes sense - Ability to abstract amongst different types of procedures (cardio vs. neuro etc.)	- Some overlap in the data; it is unknown whether activities belong to finishing the current patient or preparation of the next one - Difficult to determine the boundaries of a procedure, currently CRF data is required for this
1 shift of a resource (nurse, physician, fellow)	- Good to get an overview of activities resources perform - Less overlap, since one can only focus on the specific resource and determine a time span - Can be very useful for resource optimization	- Might miss certain patient / procedure aspects - The data does not log which activities are performed by which resource, making it difficult to incorporate e.g. machine log data into the log - Resources change tags between days, making it difficult to compare different ways of working
1 patient treatment (all the different treatments over time necessary for one problem of a patient)	- Gives a good overview of all the steps that are required for one whole treatment of a patient - Abstracts from time differences: if activity A is performed in either the first appointment or the second one, they are treated the same according to this process instance - Ability to abstract amongst different types of full treatments (cardio vs. neuro etc.)	- Data does not hold the patient references, hence it is not known whether a certain patient was already in a procedure before - Data collection is not long enough to capture a whole treatment of a patient
1 life cycle of a machine	- When everything goes as planned, Philips should only be involved in the hospital when the life cycle of the machine is over (to sell a new machine). A high-level overview of this life cycle helps Philips to understand this process	- Since we are looking at one hospital only, there would only be one case for the whole model. Hence, the model does not generalize anymore - Model would be too high-level and it is difficult to say something about this for specific treatments

Table 4.1: Possible process instances, with their pros and cons

Timestamp	**Resource**	**State**	**Artifact name**	Case
03-08-2015 09:58:11	Nurse4	Hallway	Location Nurse4	A12GH
03-08-2015 09:58:20	Nurse4	Control room procedure	Location Nurse4	A12GH
03-08-2015 09:59:00		patient_arriveert_in_kamer	CRF	A12GH
03-08-2015 09:59:00		patient_ligt_op_tafel	CRF	A12GH
03-08-2015 10:20:04	Physician1	Exam room procedure	Location Physician1	A12GH
03-08-2015 10:22:25		Iliac/Pelvis	EPX setting	A12GH
03-08-2015 10:22:25		Imaging	Machinestate	A12GH
03-08-2015 10:22:25		Fluo	Imaging	A12GH
03-08-2015 10:22:27		No EPX / no imaging	EPX setting	A12GH
03-08-2015 10:22:27		Working	Machinestate	A12GH

Table 4.2: Small example of the final state log with process instance and preprocessed data

Including the process instance in the log

In order to add the process instance to all the states that the log currently exists of, the following steps are taken.

1. Determine the first and last CRF activity for every procedure
2. Record the time stamps of the respective activities
3. Iterate through the log and assign each state to the respective case identifier according to the time stamps recorded in step 2
4. If a state falls in two CRF intervals, duplicate the state and assign both case identifiers to the states
5. If a state falls outside a procedure, ignore it

Table 4.2 shows a small fraction of the log that is the result of all the preprocessing steps. The respective columns show:

1. Time stamp of the event (in our case state change)
2. If known: resource. Only resources for location data are known
3. The actual state
4. Specific artifact that the state belongs to. In the example, the CRF artifact is shown, as well as two different location artifacts and three different machine artifacts
5. Case identifier (process instance). The example only shows one identifier, since it looks at a very short time-span.

4.2 Mining

As described before, the CSM miner (see [13]) is used to deal with the state-like artifact-centric data that is available for this research. There are more artifact-centric process mining approaches, but to our knowledge there is only one other approach that takes the interactions into account. [41] has taken the tool of [42], which did not allow for the mining of interactions, and improved it with mining interactions. However, this tool is not publicly available. In the interest of time, we decided to focus on the other challenges and use the idea proposed by [13], which does take interactions into account.

The data has been preprocessed with the usage of this technique taken into account. Therefore, the log as shown in table 4.2 can be directly imported into the CSM miner, which is implemented

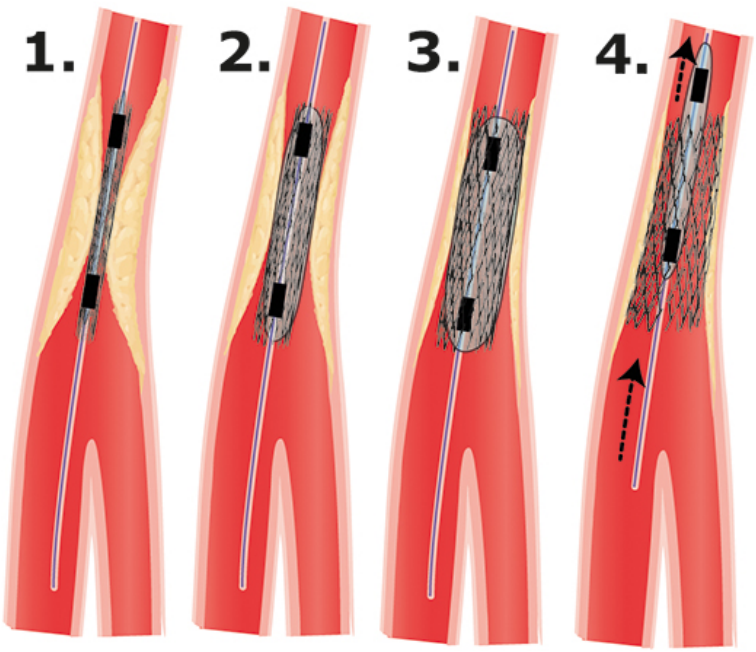

Figure 4.1: Example of the placement of a stent

as a plugin for the open-source process mining framework *ProM*. The version of the plugin that was used is version 6.7.60, therefore all results are based on this version.

For the remainder of this thesis, the specific clinical procedure that is used in the model is a Percutaneous Transluminal Angioplasty (PTA) with placement of a stent (see [43]) in the pelvis or hips. This means that a balloon is used to open up an artery, and a stent is place to keep this artery open. Please refer to figure 4.1 for a schematic picture of this procedure.

This procedure is chosen since most of the data that is available refers to this particular clinical procedure. In total, 65 cases over a period of a little over three months have been recorded. Table 4.3 shows some statistics on the 65 cases that were used for the model. This table shows that the data is quite scattered.

Measure	**Time (hh:mm:ss)**
Min	00:56:00
Max	09:30:00
Mean	02:28:52
Standard deviation	01:23:05

Table 4.3: Overview of the procedure "PTA/Stent Bekken-benen"

The maximal value of 9.5 hours is expected to be an outlier in the CRF data. However, since the CRF data was not preprocessed due to timing constraints, this will not be evaluated further.

When the data is loaded into the CSM miner plugin, no further settings need to be adjusted. The result of the mining is immediately shown as separate state models for each artifact. However, one problem did occur when trying to import the data into the miner, which was that the full log with all the artifacts appears too big and requires too much computational power to load. Since every tag of the location data has its own artifact, the number of artifacts (and especially artifact combinations) explode. It would be a nice thing to map all the 5 nurses tags to one or two artifacts: "main nurse" and "secondary nurse" (for example) and let the tool decide which nurse fulfills which role. However, it appears that the current version of the CSM miner does not support this, meaning that a solution had to be chosen.

Since the CSM miner has only been used on few data sets (see [10, 13]) we do not only want to use this miner to generate useful insights, but we want to evaluate this miner on our data set as well. Since it is not possible to test this with the whole log containing all the artifacts, we decide not to include the location data into the evaluation and generation of useful insights.

Also, the specific 65 cases that we looked at did not have any "Tilt" data. Therefore, the evaluation will be based on the data, as described before, with the following artifacts:

- Pivot
- Cradle
- Imaging
- Machinestate
- EPX Setting
- CRF

4.3 Evaluation and resulting findings

The CSM miner shows the individual artifact models. When a certain state is clicked, interactions with the most likely states in different artifacts are highlighted in orange. Please refer to figure 4.2 for an example of this visualization.

In this case, the figure shows a model of the "Imaging" and "Machinestate" artifacts. This model shows that when choosing the state "Fluo", the machine is always in state "Imaging". This makes sense, since "Fluo" is a type of radiation, and the state "Imaging" means that radiation is being used.

Besides the visual models and their interactions, [13] have implemented measures of interestingness to evaluate the interestingness of artifact interactions. These measures are (descriptions adapted from [13] and [10]):

- Confidence: strength of the prediction, expressing the estimated conditional probability of the occurrence of one, given the occurrence of the other [44].
- Support: frequency with which items occur in a set of transactions, which is an estimate of their probability of occurrence.
- Lift: ratio between probability of co-occurrence and expected co-occurrence under statistical independence. A value of 0 indicates two states never occur together, a value of 1 that two states are independent and a value above 1 that two states can be observed more often than can be expected under statistical independence.
- Conviction: similar to lift, but where lift is an undirected measure, conviction is a directed one.
- Cosine: geometric mean of lift and support.
- Jaccard: value 0 means two states never occur together, value of 1 means if the two states occur, they always occur together.
- Phi-coefficient: normalized difference between the probability of co-occurrence and expected probability of co-occurrence under statistical independence.

For the formal definition of these metrics, please refer to [13].

We will now formally evaluate the results of the mining algorithm in terms of these measures. The algorithm can look at state co-occurrences, transition co-occurrences and forward-looking co-occurrences. Please refer to [13] for an explanation and formal definition of these measures.

For each of the measures, we show the top 5 insights that were generated. These insights are then further evaluated and we describe how useful they are from a practical point of view.

The states in which the certain artifact has "Not started" yet are not included in these insights. These states occur due to the way the algorithm works: if state A starts (looking in time) and there is no value for state B for that specific case yet, the algorithm notes a "BNotStarted" state.

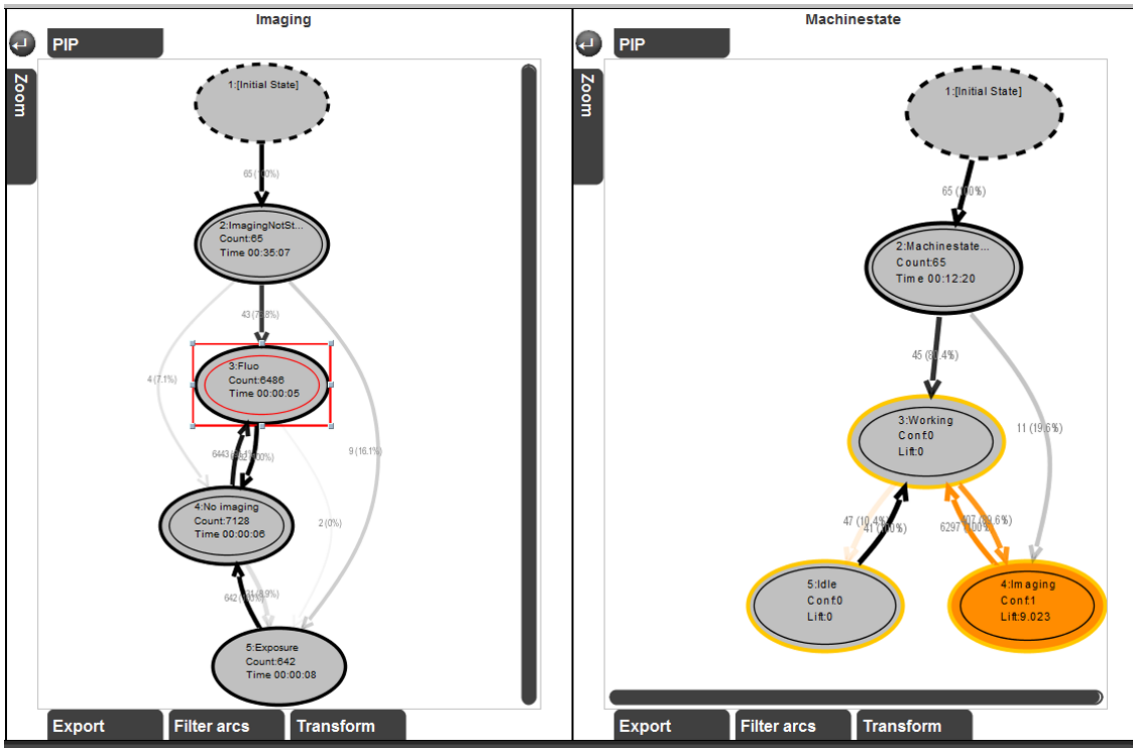

Figure 4.2: "Imaging" and "Machinestate" artifacts, including the interaction between states {Fluo} and {Imaging}

In our case, this is not applicable due to the state-like nature of our data: it is not possible that the machine has not started something yet, every artifact is always in a certain state. For example, looking at pivot of the table: this cannot be "Not started", since this would mean that the pivot element of the machine was removed from it. This would have to refer to "No pivot", which is also a state in the specific artifact.

Furthermore, we decided to exclude occurrences that were generated from the same relational tables in the database of the machine data, unless there is a weird or interesting explanation to them. For example, the same relational database table in the data is used to generate the "No imaging" states in the artifacts "EPX setting" and "Imaging", and the same relational database table is used to define that artifact "Machinestate" is in state "Imaging" when the artifact "Imaging" is in states "Fluo" or "Exposure" (since these are both a type of radiation). These relations are of course found by the algorithm (as can be seen in the screenshot example in figure 4.2) but showing that the algorithm can find these co-occurrences once is enough.

Finally, states with the same implication are left out of the evaluation. An example is that once the co-occurrence "Cradle:No cradle" with "Machinestate:Imaging" is found, the co-occurrences "Cradle:No cradle" with "Imaging:Fluo", "Imaging:Exposure" and "EPX:Foot" are not really interesting anymore. All of these co-occurrences imply that the table is not cradled when some sort of imaging is used.

4.3.1 Evaluation of the measures used for generating interesting state co-occurrences

In order to filter out very rare occurrences from the results table, we will set the minimum support level to 0.001 . This means that at least 0.1% of the total time of the process should be spent in combinations of states to be included in the results and is done to filter out states that only occur very rarely in the data. Taking an average case duration of 2.5 hours, and 65 cases in total,

Condition	Consequence	Confidence
EPX:Lungs	Cradle:Positive	1
EPX:Abdomen Prop Scan	CRF:Aanprikken	1
EPX:Lungs	CRF:patient_wordt_overgedragen	1
EPX:Abdomen Prop Scan	Imaging:Fluo	1
Machinestate:Idle	Imaging:No imaging	0.965
CRF:arts_verlaat_onderzoeksruimte	Imaging:No imaging	0.931
Imaging:Fluo	CRF:Aanprikken	0.913
Machinestate:Imaging	Imaging:Fluo	0.905
CRF:patient_van_tafel_naar_bed	Imaging:No imaging	0.889
CRF:arts_arriveert_in_kamer	Machinestate:Working	0.852

Table 4.4: 10 most likely state co-occurrences, looking at confidence

this means that in total roughly 10 minutes need to be spend in the combination of states to be considered an interesting co-occurrence. This is expected to be enough time to generate interesting results.

Evaluation based on the confidence measure

Table 4.4 shows the top 10 state co-occurrences when looking at confidence. A confidence of 1 indicates that the given state always co-occurs with a specific other state and can therefore be compared to concurrent dependencies between events or activities in traditional process mining approaches [13]. Where we looked at the top 5 co-occurrences for all the other metrics, we looked at the top 10 for confidence since the top 5 would yield almost only these confidence values of 1, which are also arbitrarily sorted and therefore do not provide a good ranking of results.

The four states that would always co-occur with 4 other states relate to 2 EPX settings (Lungs and Abdomen Prop Scan). The implication of these co-occurrences are, however, difficult to declare. Since we are looking at a case that has to do with the lower part of the body, an EPX setting "Lungs" would not be expected. Furthermore, combining any imaging-related activity with CRF activity "Patient_wordt_overgedragen" (patient is being transferred) does not make a lot of sense, since this activity would not be accompanied with any radiation at all (afterall: the patient has already left the table). This mistake could be explainable due to potential timing errors in the CRF data (see section 2.2.2), but this mistake would not have to occur over and over again.

Further investigation revealed that the support for these top 4 co-occurrences was 0.001 , which might mean that the relation did not occur often enough to draw sound conclusions upon.

The next co-occurrence, that of "Machinestate:Idle" with "Imaging:No imaging", is a relation that is much more expected and investigation revealed a support of 0.202, a factor 200+ higher than the other score. However, this co-occurrence has a slight unexpected aspect to it as well.

It is indeed true that there will be no imaging when the machinestate is idle, that is: the procedure has ended and the patient is not on the table anymore. However, we would expect the confidence of this co-occurrence to be exactly 1, rather than 0.965, since both artifacts are derived from the same relational table in the database. Further investigation in this issue reveals that the CSM miner is not able to deal with changes of two artifacts at the same time. When the exam is closed, two transitions occur: "Imaging:Fluo/Exposure" $\rightarrow$ "Imaging:No imaging" and "Machinestate:Imaging" $\rightarrow$ "Machinestate:Idle". It appears that the CSM miner interprets these two instant transitions as separate transitions, allowing the combination "Imaging:Fluo/Exposure" with "Machinestate:Idle" or the combination "Imaging:No imaging" with "Machinestate:Imaging". These combinations are not allowed when looking at the definition of the artifacts.

This is an issue that might need to be fixed in future versions of the CSM miner, but it does explain the value of 0.965 as confidence value, rather than the expected value 1 .

Condition	Consequence	Support
Imaging:No imaging	Machinestate:Working	0.324
CRF:Aanprikken	EPX:No EPX / no imaging	0.263
CRF:Aanprikken	Machinestate:Working	0.231
Machinestate:Idle	EPX: No EPX/ no imaging	0.202
CRF:Aanprikken	Machinestate:Imaging	0.101

Table 4.5: 5 most likely state co-occurrences, looking at support

The remaining co-occurrences make sense: when the doctor leaves the exam room, there should be no imaging anymore (since the doctor is the only one to perform the imaging); when Fluo is used, the procedure is taking place ("CRF:Aanprikken"); when the machine is making images, 90.5% of the time this is done by using the low-intensity radiation type "Fluo"; when the patient is taken off the table, there is no imaging anymore and finally when the doctor arrives in the room, the machine is already in working state in 85.2% of the times, meaning that another doctor or nurse has already loaded the patient data from the hospital database.

Evaluation based on the support measure

Table 4.5 shows the top 5 results when looking at support. A high value for support means that a lot of time is spent in these co-occurring states simultaneously.

The findings show some interesting results. First of all, it shows that 32.4% of the total duration of the total process (all the cases) is spent in the combination "Imaging:No imaging" and "Machinestate:Working". Interpreted to the workflow, this means that an exam is going on, but no images are made, in 32.4% of the total time that this model is based upon. This is not an unexpected finding, but it is interesting to see that a significant amount of time is spent without using the machine. This confirms the importance of including other data sources than only the machine log, since the machine log would not account for the activities in this 32% of the exam.

The next two co-occurrences ("CRF:Aanprikken" with "EPX:No EPX / no imaging" and "Machinestate:Working") both have a very high support, but we would expect the support to be equal for both values. "CRF:Aanprikken" means that a procedure is taking place, so the machine must be in machinestate "Working" or "Imaging". Of these two machinestates, "EPX:No EPX / no imaging" can only co-occur with "Machinestate:Working", since "Machinestate:Imaging" would require an EPX setting to be recorded. This could be an issue in the data quality of the CRF data, but it could also be the same issue that was described in section 4.3.1.

The final two co-occurrences show no remarkable findings. It is, however, good to note that 10.1% of the total time spent in the process is spent during a procedure ("CRF:Aanprikken") while making images ("Machinestate:Imaging"). The main goal of the machines is to support doctors during procedures while using images (hence, the name Image Guided Therapy was given to Philips' business line responsible for these machines). Test scenarios will mainly focus on this particular usage of the machine, whereas this usage only covers one tenth of the total usage throughout 65 cases. This again justifies the usage of multiple data sources for generating these test scenario's and looking at the whole workflow.

Evaluation based on the lift measure

Table 4.6 shows the top 5 results when looking at lift. It is good to note that all these results were not found in any of the other top 5 tables for other measures.

A high lift means that the co-occurrence is statistically strong and that the combination of states occurs much more often than would be expected by chance under independence assumptions [13].

The results are quite surprising. The table contains two CRF activities where the patient is being transferred ("CRF:Patient_wordt_overgedragen"), one of them with "EPX:Lungs" and the

Condition	Consequence	Lift
CRF:Patient_wordt_overgedragen	EPX:Lungs	20.034
Imaging:Exposure	EPX:One upper leg	15.288
CRF:Patient_wordt_overgedragen	Pivot:Positive	15.151
CRF:Einde_onderzoek_kies_sluiten	EPX:One lower leg	14.928
Cradle:No cradle	CRF:arts_arriveert_in_kamer	13.981

Table 4.6: 5 most likely state co-occurrences, looking at lift

Condition	Consequence	Conviction
CRF:Arts_verlaat_onderzoeksruimte	Imaging:No imaging	6.874
CRF:Patient_van_tafel_naar_bed	Imaging:No imaging	4.269
CRF:Arts_verlaat_bedieningsruimte	Machinestate:Idle	3.71
CRF:Arts_arriveert_in_kamer	Machinestate:Working	3.705
CRF:Patient_van_tafel_naar_bed	Machinestate:Idle	3.054

Table 4.7: 5 most likely state co-occurrences, looking at conviction

other with "Pivot:Positive". These are both quite unexpected: an EPX setting for lungs would not be expected in a lower body procedure, and especially not when the patient is being transferred, since this would not involve any imaging. The same goes for table pivot: this state would also not be expected when the patient is already off the table.

Something similar goes for "CRF:Einde_onderzoek_kies_sluiten" and any EPX setting: when the procedure is closed, no radiation is expected anymore.

The co-occurrence of "Imaging:Exposure" and "EPX:One upper leg" is not uncommon. Since we are investigating a procedure that implements a stent in the area around the upper leg, this setting is expected. Especially for radation type Exposure, that is mainly used to make a picture of the actual result of the procedure.

Finally, "Cradle:No cradle" and "CRF:Arts_arriveert_in_kamer" is not a weird co-occurrence, since "no cradle" can only be recorded when some form of cradle is recorded before (if there would be no cradle for the specific case, "CradleNotStarted" would be displayed). Since cradle is used to assist the nurses in putting the patient on the bed, and since the patient is usually already lying on the bed when the doctor enters, this is an explicable co-occurrence.

Further investigating the first relations reveals that all the top 5 lift co-occurrences show support values between 0.001 and 0.003 . Just like experienced before, it turns out that co-occurrences with such a low support result in some inexplicable state co-occurrences.

Evaluation based on the conviction measure

The top 5 co-occurrences when looking at conviction can be found in table 4.7. 3 of these 5 co-occurrences have already been found by the confidence measure, so we will only focus on the remaining two.

A high conviction value has the same implication as a high lift: the relationship is statistically strong. However, this measure is asymmetric in nature, where lift is symmetric. This means it does not look at the score of both directions of the relation ($A \leftrightarrow B$), but only to one direction ($A \rightarrow B$).

The first of these is the co-occurrence "CRF:Arts_verlaat_bedieningsruimte" with "Machinestate:Idle". This is a good relation and it shows no unexpected behaviour: when the doctor leaves the room, the machine is in state "Idle", meaning that there is no patient information loaded into the machine. This shows that, according to this co-occurrence, doctors upload the procedure information to the hospital database before leaving the control room.

The second relation is the co-occurrence of "CRF:Patient_van_tafel_naar_bed" and "Machinestate:Idle".

Condition	Consequence	Cosine
Machinestate:Imaging	EPX:Iliac/pelvis	0.683
Imaging:No imaging	Machinestate:Working	0.666
Imaging:No imaging	Machinestate:Idle	0.609
CRF:Aanprikken	Imaging:No imaging	0.543
Cradle:Positive	CRF:Patient_wordt_overgedragen	0.461

Table 4.8: 5 most likely state co-occurrences, looking at cosine

This is a more unexpected relation, since this would mean that the procedure information is uploaded to the hospital database before the patient is taken off the table already. A domain expert explained that doctors usually have to do some reporting activities before uploading this procedure information. This is likely due to the quality issues that the CRF data has.

Evaluation based on the cosine measure

Table 4.8 shows the top 5 co-occurrences that can be found when looking at the cosine. Cosine is defined as the geometric mean of lift and support. This could be the solution to our issue: our lift values were very high, but the corresponding support values were very low. On the other hand, support gave us very good insights into the data. Taking both measures into account is expected to generate better results.

Just like the case with conviction, only two new co-occurrences were found. The first one is the one between "Machinestate:Imaging" and "EPX:Iliac/pelvis". This is a good insight and is easy to explain: since most of the procedures start with an incision with femoral access (through the groin), the area around the iliac and pelvis must be imaged to find the right vessel. This would be done with the corresponding EPX setting. The support of the relation is 0.052, which is relatively high.

The second relation is the co-occurrence between "Cradle:Positive" and "CRF:Patient_wordt_overgedragen". This relation is a bit more difficult to explain, even though the table can be cradled to get a patient from the table to the bed, we would expect the cradle to be "no cradle" when the patient is off the bed and transferred to a different unit. However, the support of this measure is also relatively high (0.033), which indicates that our initial instinct might be wrong.

In general, this shows that the combination of the high lift with higher support generates useful insights into the co-occurrences of different states.

Jaccard and Phi-coefficient generated no new results compared to previously found co-occurrences. They especially showed equal results with cosine.

4.3.2 Evaluation of the measures used for generating interesting transition co-occurrences

After looking at the state co-occurrences, we will now look at the transition co-occurrences. The evaluation of the state co-occurrences showed some inexplicable relations when using minimum support level 0.001, so this value needs to be re-evaluated. Support for transition co-occurrence is also fundamentally different, since it does not look at the time certain transitions take ([13] assumes all transitions happen instantly), but looks at the total frequency of these transitions.

The state log that is generated by the CSM miner for our data consists of roughly 73,000 transitions for 65 cases. Setting the minimal value to occur in the results to 0.1% of these transitions would result in a minimal number of 73 removed transitions. With 65 cases, this is slightly more than 1 transition per case.

This number is not expected to give relevant results. Removing 1% of the transitions would result in a total removal of 730 transitions from the results, or on average around 11 transitions per case. This number seems reasonable, and since it is not possible to look at the implications of

this filtering, this value will be used for the minimum support level.

Tables 4.9 through 4.13 show the results of the evaluation. Every table shows the top 5 results for that measure, except for confidence (to include also values < 1) and conviction (to include also values different from ∞). Just as in state co-occurrences, jaccard and phi-coefficient delivered no new insights when compared to the rest of the measures (and provided almost the same insights as cosine).

This brings us to 35 transition co-occurrences that can be evaluated on interestingness. Evaluating these co-occurrences revealed some interesting results. Of these 35 co-occurrences, 14 were co-occurrences that show an error in the implementation of the CSM miner. It turns out that the CSM miner is, again, not able to handle concurrent transitions. Therefore, it is only able to show that a transition occurred while being in a certain artifact state, not while another transition occurred. This generates insights as: Transition "Machinestate:Working" to "Machinestate:Imaging" happens while in state "Imaging:No imaging". This makes no sense, since this transition would also imply that state "Imaging:No imaging" would make a transition to state "Imaging:Fluo/Exposure". Therefore, these transitions generate no useful insights into the process, since in reality they would be impossible.

This leaves us with 21 co-occurrences left to analyse. Comparing these co-occurrences to those generated by looking at the states in section 4.3.1 reveals that in 4 cases, both the transition "from" and "to" have already found a co-occurrence with the "consequence". These relations are already discussed and will not be discussed again.

Looking at the remaining 17 co-occurrences, it is striking to see that 12 of those have to do with the transition from "Pivot:Positive" to "Pivot:Negative". However, this transition only occurred once in the whole log. The support value for this transition is 0.071, which is higher than our threshold, but it turns out that the threshold is still too low to effectively filter out non-relevant transitions.

Since transitions are assumed to be instant (see [13]), it follows that when a "from" transition co-occurs with a certain state, the corresponding "to" transition will also co-occur with that state (and the other way around as well). In the remaining 5 co-occurrences, there is only one co-occurrence where either the "from" or "to" state has not been found in the state co-occurrences yet. This is the transition from "CRF:arts_wordt_gebeld" to "CRF:patient_klaar_voor_procedure", which happens 92.9% of the time while being in state "Machinestate:Working" (this relation was found by looking at the conviction). This makes sense, and it shows that the machine is in working state after the doctor is called (but not arrived yet), confirming our presumption that this is done by the nurses, rather than the doctor.

However, generating only one relevant co-occurrence when starting with 35 does not seem to be very feasible. Therefore, it can be concluded that after state co-occurrences have been analyzed, it does not add much to also analyze the transition co-occurrences, at least with the data that we have available.

Condition from	Condition to	Consequence	Confidence
CRF:Arts_verlaat_onderzoeksruimte	CRF:Gebeld_om_patient_op_te_halen	Imaging:No imaging	1
EPX:No EPX / no imaging	EPX:Abdomen Prop Scan	CRF:Aanprikken	1
EPX:No EPX / no imaging	EPX:Abdomen Prop Scan	Imaging:No imaging	1
EPX:No EPX / no imaging	EPX:Abdomen Prop Scan	Machinestate:Working	1
Pivot:Positive pivot	Pivot:Negative pivot	Cradle:Positive cradle	1
Pivot:Positive pivot	Pivot:Negative pivot	CRF:Patient_wordt_overgedragen	1
Pivot:Positive pivot	Pivot:Negative pivot	Imaging:No imaging	1
Pivot:Positive pivot	Pivot:Negative pivot	Machinestate:Working	1
EPX:Hand / foot	EPX:No EPX / no imaging	Machinestate:Imaging	0.995
Imaging:Fluo	Imaging:No imaging	EPX:No EPX / no imaging	0.99

Table 4.9: 10 most likely transition co-occurrences, looking at confidence

Condition from	Condition to	Consequence	Support
Machinestate:Imaging	Machinestate:Working	EPX:No EPX / no imaging	0.486
Machinestate:Working	Machinestate:Imaging	Imaging:No imaging	0.46
Machinestate:Imaging	Machinestate:Working	CRF:Aanprikken	0.438
EPX:Iliac/Pelvis	EPX:No EPX / no imaging	Machinestate:Imaging	0.247
Pivot:Positive	Pivot:Negative	Cradle:Positive	0.071

Table 4.10: 5 most likely transition co-occurrences, looking at support

Condition from	Condition to	Consequence	Lift
Pivot:Positive	Pivot:Negative	CRF:Patient_wordt_overgedragen	20.034
EPX:Abdomen Prop Scan	EPX:No EPX / no imaging	Imaging:Fluo	9.823
Pivot:Positive	Pivot:Negative	Cradle:Positive	9.661
Imaging:No imaging	Imaging:Fluo	EPX:One lower leg	9.217
Machinestate:Working	Machinestate:Imaging	EPX:Abdomen frontal	9.2

Table 4.11: 5 most likely transition co-occurrences, looking at lift

Condition from	Condition to	Consequence	Conviction
Pivot:Positive	Pivot:Negative	CRF:Patient_wordt_overgedragen	∞
Pivot:Positive	Pivot:Negative	Cradle:Positive	∞
EPX:No EPX / no imaging	EPX:Abdomen Prop Scan	CRF:Aanprikken	∞
Pivot:Positive	Pivot:Negative	Machinestate:Working	∞
CRF:Arts_verlaat_onderzoeksruimte	CRF:Gebeld_om_patient_op_te_halen	Imaging:No imaging	∞
EPX:Abdomen prop scan	EPX:No EPX / no imaging	Imaging:Fluo	61.628
CRF:Arts_verlaat_onderzoeksruimte	CRF:Arts_verlaat_bedieningsruimte	EPX:No EPX / no imaging	13.102
CRF:Arts_wordt_gebeld	CRF:Patient_klaar_voor_procedure	Machinestate:Working	7.691
Imaging:No imaging	Imaging:Fluo	CRF:Aanprikken	5.424
CRF:Arts_verlaat_bedieningsruimte	CRF:Patient_verlaat_onderzoeksruimte	Machinestate:Idle	5.136

Table 4.12: 10 most likely transition co-occurrences, looking at conviction

Condition from	Condition to	Consequence	Cosine
Machinestate:Imaging	Machinestate:Working	Imaging:Fluo	1.931
EPX:Iliac / pelvis	No EPX / no imaging	Machinestate:Imaging	1.481
Pivot:Positive	Pivot:Negative	CRF:Patient_wordt_overgedragen	1.196
Machinestate:Working	Machinestate:Imaging	CRF:Aanprikken	0.956
Pivot:Positive	Pivot:Negative	Cradle:Positive	0.831

Table 4.13: 5 most likely transition co-occurrences, looking at cosine

4.3.3 Evaluation of the measures used for generating interesting forward-looking co-occurrences

When looking at the forward-looking co-occurrences, all the measures, apart from lift, generate the same results when sorting for the highest values (1 or ∞). Table 4.14 shows the result of this analysis. The lift value was 1 for all these co-occurrences as well.

Investigation of the results shows that no useful insights were generated. The extra textual information of the CSM miner reads as follows:
"A transition from [Pivot: Positive pivot] goes 100.0% of the times to [Negative pivot] while being in [Cradle: Positive cradle] (compared to 100.0% on average)"

Unfortunately, this shows that these specific insights do not have any added value. The expected value would be 100.0%, which was also the observed value. This is further shown by the lift value of exactly 1, which shows that the relations are statistically independent.

Furthermore, recall that the transition "Pivot:Positive" to "Pivot:Negative" was only observed once in the data. Further investigation into the generated results show that all the transitions that are found in table 4.14 are infrequent transitions, even though the support for these transitions is very high (1).

Further investigation into the support metric for forward-looking co-occurrences showed that this support is calculated for the local co-occurrences, in stead of for the global co-occurrences. Therefore, looking at support to filter infrequent values from the results (as done with state- and transition co-occurrences) is not appropriate. A lift of 1 indicates a statistically independent co-occurrence. We want to filter out these statistically independent co-occurrences, and therefore we set the minimal lift value to 1.001.

However, this generates no new insights as well. All the co-occurrences (when looking at maximal support, confidence, conviction and cosine) are very infrequent or are related to the "NotStarted" states that are described before. Furthermore, it is impossible to see whether a co-occurrence is based on infrequent transitions or not. As an example, see 4.15. This co-occurrence holds a "NotStarted" event, which is ignored for this example.

This example appears promising, support is very high, all the other values apart from lift are maximal and lift is 4, which is far from statistical independence. However, the transition from "Cradle:NotStarted" to "Cradle:No cradle" is only observed once in the data. Therefore, even though the metrics are very high, this co-occurrence provides no significant insights in the process.

Table 4.16 shows the results for the analysis of forward-looking co-occurrences when looking at the only changing parameter: lift. Unfortunately, the insights that are generated by this measure are not very useful as well. The only transition that did not occur only once or twice (revealed after further investigation) is that from "CRF:Aanprikken" to "CRF:Patient_is_besteld". The forward-looking co-occurrence tells us that this happens 100% of the times while being in "EPX:One upper

Condition	**Consequence from**	**Consequence to**
Cradle:Positive	Pivot:Positive	Pivot:Negative
CRF:Patient_wordt_overgedragen	Pivot:Positive	Pivot:Negative
Cradle:Positive	EPX:Cerebral	EPX:No EPX / no imaging
CRF:Aanprikken	EPX:Bolus chase	EPX:No EPX / no imaging
CRF:Patient_wordt_overgedragen	EPX:Aortic Arch LAO	EPX:No EPX / no imaging

Table 4.14: 5 most likely forward-looking co-occurrences, looking at the maximum values for support, confidence, cosine, jaccard (1) and conviction, phi-coefficient (∞). Lift was also 1 for all co-occurrences.

Condition	**Consequence from**	**Consequence to**	**Support**	**Confidence, conviction, cosine, jaccard, phi**	**Lift**
CRF:Arts_arriveert_in_kamer	Cradle:NotStarted	Cradle:No cradle	0.25	max value (1 or ∞)	4

Table 4.15: Example of a single forward-looking co-occurrence with its corresponding metrics

Condition	**Consequence from**	**Consequence to**	**Lift**
Cradle:No cradle	CRF:Arts_arriveert_in_kamer	CRF:Sluit_aanprikpunt	63
Imaging:Exposure	CRF:Arts_wordt_gebeld	CRF:Aanprikken	26
Pivot:Positive	CRF:Arts_wordt_gebeld	CRF:Aanprikken	26
Cradle:Positive	CRF:Patient_wordt_overgedragen	CRF:Patient_klaar_voor_procedure	20
EPX:One upper leg	CRF:Aanprikken	CRF:Patient_is_besteld	12.6

Table 4.16: 5 most likely forward-looking co-occurrences, looking at lift

leg", compared to an average of 7.93%. This relation is therefore quite strong. Unfortunately, "CRF:Aanprikken" cannot happen when "CRF:Patient_is_besteld" did not happen yet (the patient must be present in order for the procedure to start), which shows that this insight is based on a mistake in the CRF data.

4.3.4 General evaluation remarks about the CSM miner

Some issues regarding the CSM miner have been described in the previous sections. However, we think it is good to summarize them in this section. This will be divided in the time before generating insights, during generating insights and actual insights.

Before generating insights

To start with a positive note: the creation of the required log works very intuitively. Documentation clearly specifies how the log should be divided into separate artifacts, and when the user did not manage to do this himself, the tool guides the user into doing this.

Following the easy generation of the log, the models appear in an intuitive manner. Different models are shown and the user can click a certain state to see the interactions with other states. The user interface is consistent with that of ProM, the general program in which the plugin works.

The first critical remark regards the inclusion of location data. Due to computational limitations, it turned out that including the location data was not possible. It would be nice if this location data could be mapped to one or two artifacts by the tool or a tool preprocessing step. After all, we are not interested in "Nurse 1" versus "Nurse 2", but we are interested in the nurses that are working during a certain procedure. Whether this is nurse 1, nurse 2 etc. is not interesting (especially since the real nurses change their tag on a regular basis).

Allowing this single mapping would result in a dramatic reduction of the possible combination of states and would likely result in the inclusion of the location data in the model.

During the generation of insights

After the model has been created, several settings can be chosen to generate useful insights. These settings will be evaluated next.

First of all, filtering noise is difficult, due to the nature of the underlying algorithm of the CSM miner. Filtering can be done by setting a minimal value (for example for support) for inclusion in the results, but the underlying results of this filtering will not be displayed visually. Traditional process mining approaches, for example the Inductive Visual Miner plugin in ProM, show a visualization of the model. When certain traces or activities are filtered away, the visualization immediately adapts to the new situation, making it very intelligible for the user to see the results of the filtering on the rest of the model. It must, however, be noted that these traditional process mining approaches can only filter on one specific model, whereas the CSM miner has to take all the artifacts and their interactions into account. The nature of the miner is therefore much more difficult.

Since changing a value in the CSM miner changes something in the underlying calculations, but the result of this underlying filtering is not shown to the user, it is very difficult to come up with a good filtering value. Our evaluation showed that this value was not chosen correctly in all of the cases, while the original article that explains these filtering techniques was used as a reference and calculations were made to determine the right values.

Furthermore, focusing on the support measure, it was observed that this minimal value looks at the overall log. That is: assume the minimal value is set to 0.001, corresponding to a minimum occurrence of e.g. 20 times. If a certain state occurs ≥ 20 times in a single trace, but never in any other trace, the relationships are still included in the result. Allowing the user to set this value per trace might be a potential solution.

Also, when trying to filter for forward-looking co-occurrences, it turns out that the metrics can all be very high, but the insight that is shown can still be based on very infrequent transitions. Therefore, a different way of filtering when using forward-looking co-occurrences is desirable.

When looking at the different measures in general, two things can be observed. The first is that it turns out that the measures "Jaccard" and "Phi-coefficient" never produced any new co-occurrences. The results of these were already included in the analysis of different measures. Especially the measure "Cosine" produced highly comparable results.

Secondly, [13] states that the measures have "an intuitive interpretation". However, we think this is only true to a certain extend. It is very difficult to interpret the results and see how well all the co-occurrences behave. Since the CSM miner lets you look at seven different measures, it is difficult to come up with the best co-occurrences. Especially when looking at only one of these measures, it could very well be that the best co-occurrence is still a very weak or bad one.

We understand that it is extremely complex to come up with a definition of "Good co-occurrences", but we also believe that, in order for the findings to be interpreted easily and intuitively, it might be worth the effort to give suggestions of good co-occurrences and letting the algorithm interpret all the measures, instead of leaving this task to the user.

Actual insights

The previous findings all had to do with actually modeling the data, selecting the right measures, and filtering them. We will now look into some results of the actual co-occurrences.

First of all, talking about "the actual co-occurrences" is difficult. We must make a choice between state-, transition- and forward-looking co-occurrences. The evaluation on our data showed that state co-occurrences captured almost all of the relevant relations. This could be because we started our evaluation with this category.

At least for our case, transition co-occurrences did not add many insights. Many of the insights that were given by this category were already given by the state co-occurrence category. Furthermore, some very infrequent transitions were also included, but this could also be a filtering issue, as described before.

Moreover, one of the biggest concerns when looking at the transition co-occurrences is the fact that it only looks at transitions co-occurrences with a state. Since our data has many concurrent transitions, this did not fit the nature of the data correctly. This also occurred with state co-occurrences, where two concurrent transitions in different artifacts were not logged at the same time. For example, suppose artifact 1 has states A and B, artifact 2 has states 1 and 2. In reality, only the combination A1 and B2 is allowed. However, a transition in both states at the same time can result in the "forbidden" state combinations A2 or B1.

Forward-looking co-occurrences did not add any relevant insights at all. Almost all the measures were 1 or ∞, depending on their limit, and showed no other information than statistical independence would do. Filtering for lift did generate some insights, unfortunately this resulted in a lot of infrequent insights. Also, filtering for a minimal lift value of 1.001 (excluding statistically independent results) did not provide any new insights.

Besides these critical remarks, it must be said that some relevant insights have been generated from the data. A lot of these insights have to do with the artifact-centric nature of the data and would therefore not have been found so easily with other process mining approaches. Even though the CSM miner has its flaws, it has provided these relevant insights which help us get a better overview of the actual workflow. Also, the textual description of the actual numbers helps to interpret these findings. The relevant findings will be described in the next section.

4.3.5 Relevant insights generated by the CSM miner

Since the relevant insights are somewhat mixed with tool evaluation results in the previous sections, we will give an overview of relevant insights. This will be divided in two lists: one with insights that already confirm our findings and show the correctness of the data, and another with surprising new insights.

Quite a few insights that already confirm our findings were found. These are listed below, with some practical implications:

- When the machine is in state "Idle" (no exam is going on), there is no imaging. This means that the usage of imaging is limited, which is a good thing since it is only required during exams.
- When the doctor leaves the exam room, no imaging is used anymore. This is a good finding that shows that the doctor is the only one to use imaging. Apparently, this hierarchy is respected in this particular hospital.
  - More specifically, when the patient is taken off the table, no imaging is used anymore. This shows that imaging is really only used for treating patients.
- Fluoroscopy is mainly used during "Aanprikken", which means the time that the actual exam is taking place. This is good to note, since it also shows that the CRF data (Aanprikken is a CRF activity) is not that bad afterall; this is behaviour that was expected.
- When images are taken, 90.5% the low-intensity radiation (Fluoroscopy) is used. This shows that doctors really make the distinction between image quality and amount of radiation: if the image quality would have been more important they would use more Exposure (causing more harm to them and the patients due to the higher intensity).
- The most likely EPX setting when using exposure is "One upper leg". Since we are looking at a clinical case that centers around the pelvis and upper legs, and since exposure is usually used to check the result of an intervention, hence only looking at the specific place where something happened (in our case the upper part of the legs), this setting is easily explained.
- The most likely EPX setting when using imaging in general is "Iliac / pelvis". This also makes sense, due to the nature of the clinical procedure.

Next, we will describe some less expected, or more special, insights that were generated from the evaluation.

- When the doctor arrives, 85% of the times the machine is already in state "Working". This means that the exam has already been started and all the patient information has already been loaded by the nurses.
- In 32.4% of the time during an actual exam, no imaging is going on. This shows that the machines are not used throughout the whole time that the patient is lying on the table.
  - In fact, imaging is only used in 10.1% of the total time of the 65 procedures. This is striking, since the main goal of the machines is to support doctors in their procedures by using images (Image Guided Therapy), but only 10.1% of the workflow is actually supported by the images from the machines.
- When the patient is transferred from the table to the bed, the machinestate is usually "Idle". This shows that the exam is already closed before the patient is taken off the table.

Since no location data could be included in the model, no relations with these have been found. However, some insights relating to location data have been provided by the CRF data as well (due to the activities doctor enters/leaves control/exam room).

4.4 Limitations of the models

One of the biggest limitations of the models is that they model the process in terms of different states. This is a design choice that is driven by the nature of the data, as described many times before in this thesis.

However, this design choice has quite some impacts for the actual users of the models, since they do not know how the model works. They mainly think in terms of single activities, whereas a single activity would mean a large combination of states in the models that are presented here.

This limitation shows that the very low-level data has been translated into a bit higher-level data, but is not on the same conceptual level as users think. This is a limitation that should be overcome, one possible solution is to create a translation layer between the model and the user activities.

Another limitation lies in the nature of state machines as modeling form. When looking at the mined models, the exact branching possibilities are not clear. That is; it is not made explicit in the model whether certain activities (or combinations of states) occur in parallel, or whether there is a choice between them.

If this turns out to be a problem for Philips, a different mining technique could be considered. However, this would probably be a difficult step, since currently the only available artifact-centric mining approach that mines interactions is used in this thesis.

The last limitation is that it is not possible to see all the interactions in one glimpse of an eye. Users have to click on one state in a certain artifact, after which the interactions with different artifacts will be highlighted. It is not possible to follow one path through the whole systems that goes across the borders of the different artifacts.

Even though it would be useful to do this to get a quick overview of the whole system, it would also result in a very complicated model with a lot of interactions. It is therefore more a limitation of the whole approach, or actually that of modeling a complex system in general, than of the specific mining technique.

4.5 Conclusions and recommendations

In this chapter, we have further extended the process data by adding the specific process instance to the data. For the purpose of these specific scenarios (usability scenarios), the process instance "1 procedure" is chosen. However, in order to add this process instance to the data, CRF data is currently still required. It is therefore recommended to find a way to include this specific process instance directly into the machine log data. This will ensure higher data quality (since no extra human step is involved) and will ensure that this process instance is known on a higher scale (since the machine log data is available from almost every hospital in the world).

Following this addition of the process instance, we have mined the model using the CSM miner. We have evaluated the CSM miner itself, but also generated some useful insights. One of the most important insights was that only one tenth of the whole workflow involved imaging, and when the patient was lying on the table this number was around 70%. Therefore, it is highly recommended that Philips keeps including different data sources that provide more valuable insights into the whole workflow as tight orchestration between patient, machine, and hospital staff, rather than focusing only on the machine log data.

Finally, some limitations of the models were discussed. One of the main limitations was that usability engineers think in terms of tasks and activities, while the model abstracts based on different states. It is highly recommended that this issue is further investigated, for example by creating a translation layer that acts as an interpreter between the usability engineer and the model. This issue is underwritten by [39], who talk about the difficulty of the gap that exists between sensor logs and actual tasks that are performed in process mining. However, they provide no solution to the specific problem.

Chapter 5

Creating Field-based Usability Testing Scenarios

In this chapter, we will address research goal 3: using field data to create real-life usability testing scenarios. Chapter 4 described the transformation of process data to a model. We discuss an approach that can use this model to evaluate the quality of a test scenario, looking at several measures. This approach provides factual information to assist usability engineers in creating test scenarios based on their personal domain knowledge.

Currently, usability engineers use their own domain knowledge to define testing scenarios that are aimed at the specific test and the specific feature that needs to be tested. These tests incorporate frequently used functions and primary operating functions, which can be interpreted as safety features. These scenarios consist of different tasks that are explained to a participant who has to execute the tasks in the scenario step-by-step. The observer (usually usability engineer) scores the tasks. See [6] for more information on the specific definition of the scenarios.

Even though these scenarios are based on a lot of experience, both from observations by the usability engineers, and from clinical domain experts, these scenarios are not based on the real data (usage of the system in the field). It is therefore good to provide the usability engineer with assistance in an interactive tool when generating the scenarios, making sure both the domain knowledge of the usability engineer, but also the data regarding the whole workflow from the field, are reflected in them, to ensure the scenario is as realistic as possible.

Usability engineers have a lot of domain knowledge. However, it is difficult for these engineers to realize that they have this domain knowledge until they are confronted with a contradiction to this knowledge. This could, for example, be because of an activity that is not intuitively placed within the workflow when looking at the model. Only when such a contradiction takes place, they know that something is not possible or does not lead to an intuitive scenario.

Since this knowledge is difficult to capture, it is not possible to include this knowledge in an automatic scenario generation method. However, the current way of working (by using only domain knowledge) lacks the insights that data might give. Focusing solemnly on domain knowledge is therefore not desirable as well.

For this reason, we came up with a hybrid approach that uses domain knowledge in combination with the actual data to get to realistic usability test scenarios. In this approach, the usability engineer would be the responsible person to come up with such a scenario. This is the same as the current process. This task of the engineer is steered and supported by the data in the model.

Section 5.2 introduces three different measures to guide the usability engineer by using an example. Before looking at these measures, we should first look at usability tests in general.

5.1 Usability tests and their testing scenarios

Usability tests are tests that are performed in a later phase of the development of the iXR machines. These tests are not aimed at formal validation and functional testing / unit testing ("does the machine stop when I press the "Stop" button?") but they are aimed at how intuitive a system is working and how intuitive the design of the features is for the user. The tests are not performed automatically but require a real user, the participant of the test, to perform the test. This participant is central in the whole test and the interaction between the user and the system is the subject that is tested.

Functional tests are, of course, also performed by Philips. This form of testing focuses on generating the highest completeness and testing all the features of the machine extensively. This is not the aim of a usability test, since these tests are usually limited in duration, and therefore only focus on the mainstream behaviour of the system. This does not mean, however, that completeness is not important for a usability testing scenario. The best thing is still to maximize the completeness (meaning that many different features are tested), but since we are dealing with real people and limited testing time this is not possible. The scenarios for functional tests are out of the scope of this thesis.

A specific type of test scenarios are task scenarios. These are scenarios that are realistic, encourage an action, and do not give away how the interface should be used [45]. In our case, this means the interface of the system and how to use this system. Task scenarios note the goals and questions to be achieved and sometimes define the possibilities of how the user can achieve these goals. Scenarios are critical both for designing an interface and for usability testing [46]. Philips also uses these types of test scenarios for usability testing purposes, and these are the scenarios that are to be made by the usability engineers.

As the name suggests, task scenarios are composed of different tasks. When a usability test is performed, these tasks will be read out by the observer, usually the usability engineer himself, and will be performed by the participant performing the test. Each of these tasks are scored individually, for example by looking how well the participant was able to perform the task without help from the observer.

One of Philips' goals is to make the usability tests more realistic. This is why we look at the whole model to define scenarios, even if the states that must be tested can be easily reached in isolation. For example, testing how intuitive a certain button works is perfectly possible by asking the participant to use this particular button straight away. Since this is something that the participant would normally only do in a specific workflow (as sequence of other activities) or potentially even only when certain resources are present, these extra steps need to be included in the scenario to reach this higher level of realism.

Furthermore, testing certain features in isolation might provide biased results, since participants are really focused on this particular task. When something happens in the setting that they are more used to (the normal workflow), they might feel more at ease and focus more on the whole workflow, rather than this single, isolated task.

When we talk about creating a test scenario, we mean looking at the different tasks that together form such a scenario. Each scenario may have a different purpose in mind, since each scenario may want to be used for testing a different new feature of the machine. The usability engineer will be guided by the interactive tool on the sequence and inclusion of the different tasks into the overall scenario.

Since the number of participants for a usability test is limited, and since the test must provide statistically significant results, Philips uses the same usability protocol for all the participants of a specific test. However, since every test has a different goal and different functionalities that need to be tested, the protocols differ between different tests.

Moreover, since the usability engineer himself is involved in creating the scenarios (due to the

hybrid approach that is described before and will be further elaborated in the remainder of this chapter), meaning that the usability engineer can steer the scenario to include specific tasks that he wants to include, and since only one scenario will be used for a specific test, we are looking for a way to come up with a single scenario for each test. Using more scenarios will not provide statistically significant results for Philips and is therefore not recommended.

5.2 Proposed measures

When looking at the domain of structural coverage of testing code, [47] have identified three types of coverage. These are statement coverage, branch coverage and path coverage. These measures mainly reflect the completeness of a scenario (in the form of the different types of coverage), but also focus on the coverage of specific branches. This can also be used in our case, but since the goal of a usability scenario differs from that of functional testing (see section 5.1), these measures must be adapted.

The notion of completeness is something that is applicable to our scenario as well, even though the goal is not to generate scenarios for which the completeness is 100%. However, including some form of completeness measure is desirable to see how many functions of the machine are tested in one single test. This is also desired by Philips, who want to quantify the total percentage of the system that is tested. They expect that, in the future, the total percentage of the interaction between user and system that is tested will be an important number to consider.

Since the goal of our scenarios is to be as realistic as possible, see section 5.1, the concept of the "coverage of specific branches" can be applied in our case as well. By choosing tasks that occurred more often in the data (hence, they are more "representative"), the scenario feels more familiar to the participant. Furthermore, Philips has indicated that the scenario should follow a natural flow and if this is not the case, they want to know this so they can prepare the participant on this deviation of normal flow. In order to do this, the different possibilities in the form of choices in the model must be quantified and translated to probabilities or frequencies. These data insights are usually not covered by the domain knowledge of the usability engineer, since they are obtained from the data.

Furthermore, we define a third measure that helps the usability engineer in balancing the completeness and the branching options that are taken. This is the reachability of certain states, which shows whether it is possible to get from state X to state Y and can be used to see whether the desirable states to be tested can be reached from the current state. If this reachability value indicates that your desired state cannot be reached when making a certain choice in the model, this choice can be reconsidered.

Our final goal is to create an interactive tool that can be used by the usability engineers to generate usability scenarios. In every step (selection of the next activity or task), the usability engineers would be guided by the three different measures described above.

The three measures will be formally defined in the remainder of this chapter. However, in order to intuitively illustrate these measures, we will use an example and demonstrate its use.

Consider a usability engineer that has to create a certain scenario. Figure 5.1 shows the underlying workflow model with its corresponding transition probabilities. In this scenario, the main focus on the usability test is on tasks I and M (highlighted in grey): these are tasks that need to be tested in any scenario. Every state in the model refers to a task for the scenario. This engineer starts at the first state (A) in the model. He can now decide which choice to make: in practice both B and C are observed. B happened in 10% of the observed times, C in 90% of the observed times (these are the respective transition probabilities). States I and M are both still reachable, no matter which task is chosen. The engineer decides to follow the most representative behaviour, choosing state C.

We are now in state C of the model. The only choice is D, which is the choice we make. This brings us to state D. Again, the only choice is task E, which is chosen. I and M are still reachable.

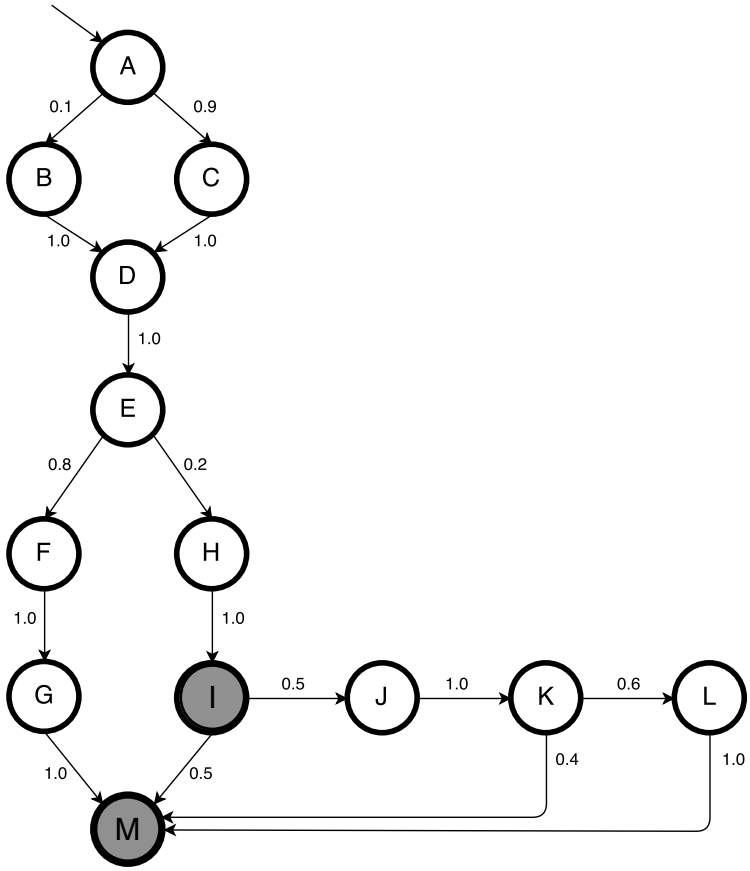

Figure 5.1: State model with transition probabilities

When we are in state E, the choice between F and H must be made. F is observed in 80% of the previous occasions, H is only observed in 20%. The reachability measure tells the engineer that state M will always be reachable, but state I is only reachable when the engineer chooses the (less frequently observed) path via state H. Since the goal of the test is to include I, state H is chosen.

We are now in state H of the model. No choice can be made, so we continue to state I. This state represents one of the tasks that we need to include in the scenario. We can choose to go directly to state M, which represents the other task that needs to be included in the scenario, but we can also go to state J. These states are both performed equally often in the past. Since they are both performed equally often, and since one of the target states M is reachable via both choices, the usability engineer looks at completeness. When state J is included, the total number of performed tasks, and thus the number of visited states in the model, increases. Including this state would provide a more complete scenario, so this state is chosen.

We are now in state K of the model. Two choices can be made: going to target state M, which happened in 40% of the past occasions, or going to state L, which happened in 60% of the past occasions. State M can be reached in both cases, since one choice is to go directly to state M and

the other choice provides another path to state M.

When looking at the measures, transition probabilities would say that the engineer needs to choose state L. This state would also increase the overall number of visited states in the model, so the completeness measure would also suggest this. However, the engineer knows that task L is tested enough in the previous version of the machine (so it is not necessary to test this again), but that doctors usually find this task very annoying to perform. The engineer therefore decides to use his domain knowledge and not to include this task (going against all the suggestions), since he does not want to annoy the doctor. He chooses task M, finishing the scenario.

Due to timing constraints, the scenario generation tool has not been developed yet, but the measures are described and some of them are tested in the rest of this chapter.

5.3 Related work

The next sections describe related work regarding test scenario generation and the proposed measures. Our practical problem is that we want to generate a scenario from a model, but when we translate this to a technical problem it is not possible to generate scenarios from models. It is, however, possible to look at traces of a certain model. In our case, these traces through the model would represent the different scenarios.

A usability scenario is the same for each participant and does therefore not contain any choices. The scenario must contain tasks that have been observed in the field. Therefore, a scenario in the terms of this thesis would not be a set of traces, but is one instantiation (trace) through this model. The model might contain choices, but the scenario does not.

Traditional conformance checking techniques focus solemnly on the match between the whole log (as a set of traces) and the model. See for example [48, 49]. This is somewhat related to our problem, but the main difference is that we want a certain score that shows how well a single trace (scenario) reflects the behaviour in the model. This differs from the traditional conformance checking approach. Related work that might be of use will be presented in the next sections.

5.3.1 Test scenario generation

Before getting to the specific measures described above, related work regarding test scenario generation in general is presented.

[50] and [51] have come up with an approach that uses Statecharts and/or Finite State Machines to generate automated test scenarios. However, the test scenarios are mainly focused on software, whereas our approach is based on user actions and more subjective usability testing, in which the completeness and total coverage is not as important as in software testing. This difference is also highlighted in [47], which is an article that focuses on software testing. Here, it is stated that "the concepts of structural coverage based on code exercised are well studied, for example, statement coverage, branch coverage and path coverage. We state that path coverage in a test ready UML statechart model is achieved if tests corresponding to all the paths or sentential forms are generated". It is clear that this focuses mainly on testing as much code as possible, without having to take other considerations, such as dealing with a real user or with limited available testing time, into account.

In [52], a comparison of different model-based testing techniques is made, and a taxonomy to better formalize their evaluation has been proposed. However, this article again only focuses on model-based software testing. [53] came up with an approach that uses activity monitors to generate tests. This approach is only used for closed-loop input testing, which means that both the input and the output of the test can be generated. This is not the case in our approach, since we do not have output of the particular test. We are only interested in the test protocol itself due to the different nature of the test.

More approaches exist, some of them focus on software testing, for example using UML diagrams [47, 54, 55, 56], and some on physical systems, for example to design and execute test

cases for the purpose of component and system-testing using UML sequence diagrams [57] and using decompositional model checking to test the design of a processor [58]. The problem with these techniques is that they are all focused on automatic system / software / unit testing, not on usability testing with a real user.

To our knowledge, no authors have come up with such a usability testing technique before.

5.3.2 Transition probabilities

Cost-based fitness analysis [59] allows for a single trace comparison to the model, but is unweighted in that it checks only whether certain traces are possible, given the model, but not how likely it is to see this trace in the model.

Some authors have incorporated a non-binary approach (i.e. a trace is not correct or wrong, but there is something in between) to conformance checking. [60] have handled temporal outliers with respect to a probabilistic model, where the overall likelihood of a single trace is the product of the temporal probabilities of the activities. We are not interested in the timing of the activities, but the computation of the overall likelihood of a single trace is something that can be used.

[61] focus on aspects that go beyond the control-flow to include contextual anomalies in conformance checking. They apply a certain probabilistic model to determine whether unexpected events are anomalous or not: unexpected events are detected as anomalous only based on a certain likelihood of occurrence. This likelihood value is, however, only used to determine whether an event is anomalous or not and the likelihood value is discarded after this determination, falling back to the binary approach to the problem.

[62] assigns an anomaly score to a log based on kernel based sequential data anomaly detection. Outliers can be detected by identifying the points with high anomaly scores or with a certain predefined standard deviation from the mean. This article does therefore not look at a trace in regard to the model, but looks at individual outliers within these traces. A trace with outliers can be seen as a bad scenario, but this does not necessarily have to be the case. Pressing the emergency button of the system, for example, could very well be an outlier in the model, but if the interface of the emergency button is tested, this is a very important task to include. Therefore, some more measures are needed and this approach will not be used.

The general idea of probabilistic automata, as described very well in the PhD dissertation [63], can be used. An example of this is that "each execution of a probabilistic automaton delivers a number of traces. Looking at each collection of traces, the hypothesis "This collection of traces is generated by the given probabilistic automaton" is tested". To our knowledge, these general ideas have not found any connection to the field of process mining yet.

5.3.3 Completeness

In process mining, the term completeness usually refers to how complete the model is compared to the whole event log (thus again: set of traces). The completeness is very high if all the behaviour that is seen in the event log can be executed in the model, implicitly meaning that all the behaviour of the process is contained in the log as well. See for example [64, 65, 66, 67].

Another usage of the term is found in [68] to refer to "the amount of information a log needs to contain to be able to discover the underlying process".

Searching for trace completeness yields no satisfactory results. Hence, it is safe to assume the completeness of one specific trace (in our case the scenario) has not been covered yet.

5.3.4 Reachability

Given a system and property p, reachability model checking is based on an exhaustive exploration of the reachable state space of the system, testing whether there exists a state where p holds [69]. Such property p could be "Is it possible to reach state X from state Y?" .

In terms of Finite State Machines, reachability means a state is reachable if a sequence of inputs forces the FSM to evolve from an initial state to that state [70].

[70] also states that the process terminates as soon as a fixed-point is reached, i.e., no newly reached states are found.

However, just like the transition probabilities, we are not interested in the binary approach (it is possible to reach X or not, as used by the work given in this section before), but more interested in whether it is always possible, never possible or sometimes possible to reach a certain state. For this, Markov chain theory seems to come in useful.

Determining least fixed points corresponds to finding the probability of ever getting from X to Y. An example of this can be found on page 35 in [71]. This theory seems applicable to our case and will therefore be further used to further develop our proposed measures.

5.4 Approach

To summarize, we have a model and we have a specific trace through that model. We want to score the specific trace on different dimensions. These dimensions tell how well this trace would perform as a usability testing scenario. These scores act as additional information for the usability engineer, who will always stay in charge of the scenario generation process.

The next sections will deal with the separate measures that are used to score the scenarios. But first we define a few general concepts that are used throughout these measures. These are the concepts of all possible states, a trace, a log, and the total set of states in a log.

Definition 3. *S is the total set of states,*
A trace is a sequence S^**,*
A Log is a multiset of traces,
Log: $S^* \to \mathbb{N}$ *,*
The set of states in a log $S^\bullet = \{s \in S | \exists \sigma \in Log : s \in \sigma\}$*,*
For a sequence $\gamma \in S^*$ *we define* $s \in \gamma$ *iff* $\exists 1 \leq i \leq |\gamma| : \gamma_i = s$ *.*

5.4.1 Transition probabilities

Transition probabilities consider the reflection of choices in the model into the specific trace. During a test, we want to prevent tasks that the user would not perform at the specific position in the scenario. If there is no choice on the next task in the model, the next task must be performed. But if there is a choice between different tasks, it is likely that the user will choose the most occurring task. Whether this most occurring task should be included in the scenario, or whether there is a different task, is up to the usability engineer upon the creation of the scenario. Transition probabilities can guide the usability engineer into the inclusion of such choices.

Transition probabilities describe the conditional probability that s' is the next state, given that we are currently in state s, in a certain trace (execution run), compared to the observed behaviour of all traces in the log. Formally:

Definition 4. *Given a log Log, trace* γ *and two states s and s':*

$$\textit{Transition probability(s, s', Log)} = \frac{\sum_{\gamma \in Log} \left| \left\{ 1 \leq i < |\gamma| \,\middle|\, \gamma_i = s \wedge \gamma_{i+1} = s' \right\} \right|}{\sum_{\gamma \in Log} \left| \left\{ 1 \leq i < |\gamma| \,\middle|\, \gamma_i = s \right\} \right|}$$

The overall likelihood score of a given trace then is the product of all the transition probabilities for the transitions in a given trace, compared to the transition probabilities in the log. Formally:

Definition 5. *Given a log Log and trace* γ*:*

$$\textit{Likelihood}(\gamma, \textit{Log}) = \prod_{i \in \{1,..,|\gamma|-1\}} \textit{Transition probability}(\gamma_i, \gamma_{i+1}, \textit{Log})$$

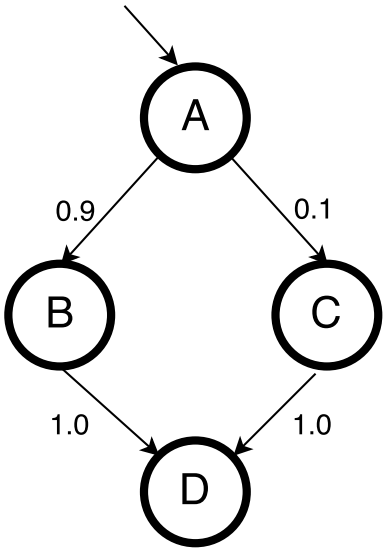

Figure 5.2: Simple state model with transition probabilities

This is illustrated with an example. Consider the Log
$[< A, B, D >^9, < A, C, D >]$
and figure 5.2 for the model of this example.

Figure 5.2 shows a very simple (state) model in which the process starts in state A. After state A, a choice can be made between either state B or state C. State B was chosen in 90% of the cases in the past (transition probability = 0.9), state C in 10% of the cases in the past (transition probability = 0.1). Since there is no choice after B, respectively C, both these transition probabilities are 1.0 . This shows that the sum of all outgoing probabilities from a state equals 1.0 in all cases, except for the final state.

As described in the section on related work, [60] has defined overall likelihood of a single trace as the product of the temporal probabilities of the activities. We will use the idea of calculating the overall likelihood of a single trace, but it will not be the product of the temporal probabilities, but that of the transition probabilities. That is, the probabilities of going from state "From" to state "To". Recalling this to our example in figure 5.2, this would mean that the trace $< A, B, D >$ would have an overall likelihood value of (0.9 * 1.0 = 0.9) and trace $< A, C, D >$ would have an overall likelihood value of (0.1 * 1.0 = 0.1).

Please note that a transition from A to D directly would result in a transition probability of 0 for that transition, since it has not been observed in the data in the past. Since the overall likelihood value is the product of all individual probabilities, a transition from A to D would result in an overall likelihood value of 0 for the whole trace as well, no matter how high the other probabilities are.

In order to apply this formula on other examples from the real data, we take the following steps.

1. Use state log as reference data: the input of the algorithm will be the state log of all the different cases.
2. Create a probability matrix with all the transition probabilities.
3. Transform the scenario / trace into individual transitions (tuples of states (From, To)).
4. Assign probabilities to all these transitions by looking up their values in the probability matrix created in 2.

5. Compute likelihood for scenario / trace by multiplying all the probabilities acquired in 4 to get the overall likelihood score.

5.4.2 Completeness

As described before, the goal of a usability testing scenario is not to be as complete as possible, since this would not be feasible due to the nature of the test (testing with a real user, who has limited time). However, it is good to perform a test that is as complete as possible, given the constraints that the usability engineer will keep in mind. Therefore, including completeness into the suggestions that the usability engineer gets is a relevant thing to do. The usability engineer can determine how complete the scenario must be, and potentially split the test in two different scenarios if the completeness would otherwise be significantly too small.

As mentioned before, the notion of completeness that we use is different than that usually used in process mining. It is good to note that with the term completeness, we look at the completeness of one specific trace in regard to the whole model. Translating this completeness to our case shows how many features of the machine have been tested in a single test.

A trace can contain states that are not observed in the log, since the usability engineer can include self-defined states in a trace that were not observed in the log. An example of this is when the usability engineer wants to test the emergency button, but the emergency button was not pressed in the observations in the log. The state in which the emergency button is used would then be an element of S, but not of $S^{\bullet}$.

We calculate completeness as compared to the observed states in the log ($S^{\bullet}$). Performing the calculation on S, thus including the emergency button example above, is misleading, since the given trace could only contain non-observed behaviour in the log and still show a high completeness value. The user would then think that he covered some behaviour observed in the field, whereas he has actually only included behaviour that was not observed in the field.

Completeness of a trace is calculated by dividing the number of states in the log by the total number of states in the log. Formally:

Definition 6. *Given a log Log and trace γ:*

$$\textit{Completeness}(\gamma, \textit{Log}) = \frac{\left|\{s \in S^{\bullet} | s \in \gamma\}\right|}{|S^{\bullet}|}$$

In order to illustrate this, another example is used. Consider the Log $[< A, B, C, E >, < A, D, E >^5]$ and its corresponding model in figure 5.3.

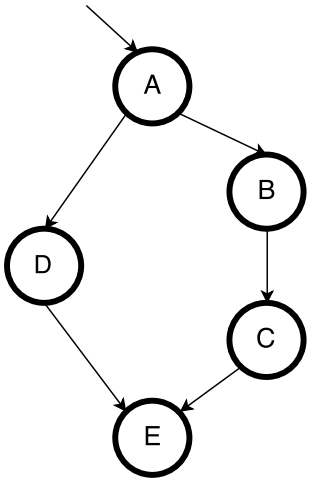

Figure 5.3: Simple state model, without transition probabilities

The trace $< A, B, C, E >$ would visit 4/5 of the total number of states, the trace $< A, D, E >$ would visit only 3/5 of the total number of states. The first trace can therefore be marked as more complete.

The completeness notion, as described before, does not take the frequency of states into account. If state A is visited nine times and state B is visited only once, it makes sense to add some weight to state A, since not including this state in the scenario would remove a lot of the mainstream behaviour. Furthermore, these weights could also be added by the usability engineers themselves. It is, for example, possible that a certain frequently used function or safety feature is always required in a test scenario. Giving this state a very high weight would distinguish it from potentially more frequent, but less important, features.

Weighted completeness takes the idea of completeness, as described in definition 6 and extends it by adding weights to the states. It is calculated by dividing the sum of the weights of all states in a given trace by the total sum of weights in the log. Formally:

Definition 7. *Let w_s be the weight of state s, for every $s \in S^\bullet$. Then we define, given a log Log and trace γ:*

$$\textit{Weighted completeness}(\gamma, \textit{Log}) = \frac{\sum_{\{s \in S^\bullet | s \in \gamma\}} w_s}{\sum_{s' \in S^\bullet} w_{s'}}$$

Please note that w_s can be specified based on domain knowledge or log properties. In our example, we specify w_s as the frequency of state s in the Log, as shown in the example below.

Adding weights to our example, states A and E are visited 6 times, D is visited 5 times and B and C are only visited once. Therefore, states A and E could get score 6 when included in a trace, state D score 5 etc. Please refer to table 5.1 for an overview of the scores.

State	**Visited** $< A, B, C, E >$	**Visited** $< A, D, E >^5$	**Sum / Score**
A	1	5	6
B	1	0	1
C	1	0	1
D	0	5	5
E	1	5	6

Table 5.1: Overview of weighted completeness scores

When taking a weighted approach the trace $< A, B, C, E >$ would get score (6 + 1 + 1 + 6 =) 14/19, whereas trace $< A, D, E >$ would get score (6 + 5 + 6 =) 17/19. Trace $< A, D, E >$ would therefore be considered as more complete, which differs from the unweighted approach.

The workflow has an enormous possible set of states. It is impossible, and presumably also not necessary, to test all states. Partitions of states could show the same behaviour. For testing purposes, it would be enough to visit one of these states in such a behaviorally equal partition. States could be partitioned based on one of the measures in the CSM miner (support, confidence, lift etc.), but they could also be partitioned by a usability engineer who defines state equivalence based on his domain knowledge. This approach could be used inside different artifacts (states A and B in artifact 1 are equivalent), but also across them (states A and B in artifact 1 are equivalent with states A and B in artifact 2).

For this purpose, we extend the definition of weighted completeness (definition 7) to include these partitions. This partition completeness is defined as the sum of the weights of the partitions in a trace, divided by the sum of the total weights of the partitions in the log.

Definition 8. *We have a partitioning α of S:*

$\alpha : S^{\bullet} \rightarrow S^{\alpha}$

We have a weight w_p for each partition $p \in S^{\alpha}$,

Given a log Log and trace γ:

$$\textit{Partition completeness}(\gamma, \textit{Log}) = \frac{\sum_{\{p \in S^{\alpha} | \exists s \in \gamma : \alpha(s) = p\}} w_p}{\sum_{p' \in S^{\alpha}} w_{p'}}$$

Please note that again, w_p can be specified based on domain knowledge or log properties. Also note that a partition size of 1 for each partition, and a weight of 1 for each partition would result in the same result as definition 6.

See the very simple example process of two artifacts, depicted in figure 5.4. The dotted circle around states "A" and "B" means that these are state equivalent and therefore show the same behaviour.

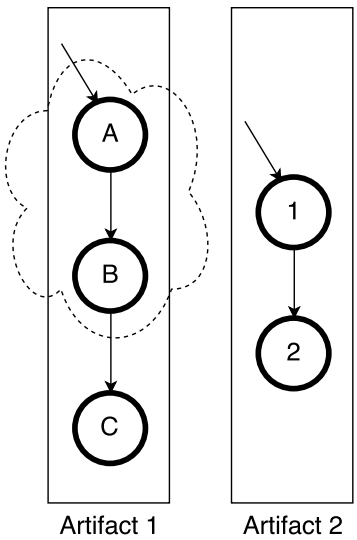

Figure 5.4: Simple state model of two artifacts, without transition probabilities. Dotted circle indicates state equivalence.

Without the dotted circle, looking at the composite state model of these two artifacts would result in six state combinations: A1, A2, B1, B2, C1 and C2. The completeness (whether weighted or unweighted) would be based on these six states.

However, domain knowledge in the form of behavioral state equivalence would help in reducing the number of states required for testing. In this case, knowing that states "A" and "B" behave equally reduces the required testing from 6 combinations (A1, A2, B1, B2, C1 and C2) to 4 combinations (A/B1, A/B2, C1, C2).

This is only a reduction of two possible state, but one can imagine that more domain knowledge would result in further reduction of states. *Please note: it is assumed that such domain knowledge is present for this specific extension to work.*

5.4.3 Reachability

Transition probabilities look at choices observed in the past from a local perspective. That is: they only look at the next step when a choice comes up. It could, however, very well be possible that this local best next step does not correspond to the global best step. Therefore, it is important to see what the impact of a certain choice would be on the rest of the scenario.

Take, for example, the very extreme (hypothetical) example where a machine is turned off right after turning it on in 80% of the cases. If the first step in the scenario would be "Turn on machine", the transition probabilities would suggest that the best next step would be "Turn off machine". However, when choosing to turn off the machine, no other choices can be made and the scenario would be completed.

In this specific example, using a reachability measure might be useful. This measure would show that once the usability engineer chooses the activity "Turn off machine", all the other states are not reachable anymore, meaning that the test is completed.

As described in the related work, we will use Markov chain theory to determine least fixed points, corresponding to the probability of getting from state X to state Y (in an infinite number of runs).

The calculation for this, as explained in [71], is as follows.
In general, $\underline{y}$ is a fixed point of function F if $F(\underline{y}) = \underline{y}$ and $\underline{x}$ is the least fixed point of F if $\underline{x} \leq \underline{y}$ for any other fixed point $\underline{y}$. Then we define:

Definition 9. *Given a log Log:*
T is a target set of states, such that $T \subseteq S^\bullet$,
Reachability(T) is the least fixed point of the function $F : [0,1]^{|S^\bullet|} \rightarrow [0,1]^{|S^\bullet|}$, *such that*

$$(F(\underline{y}))(s) = \begin{cases} 1 & \text{if } s \in T \\ \sum_{s' \in S^\bullet} \textit{Transition probability}(s, s', Log) * \underline{y}(s') & \text{otherwise} \end{cases}$$

Using the following algorithm will approximate Reachability(T):

Definition 10. $\underline{x}_0 = 0$,
$\underline{x}_{n+1} = F(\underline{x}_n)$,
Reachability(T) $= lim_{n \rightarrow \infty}\ \underline{x}_n$.
(in practice: terminate when the difference between two subsequent values of $n < \epsilon$, *for a user-defined tolerance value* ϵ.

The outcome of Reachability(T) is a vector which shows the probability of reaching the target state from each state in $S^\bullet$. Repeating this calculation for each state in $S^\bullet$ as target state gives all the probabilities of reaching any state in $S^\bullet$ from any other state in $S^\bullet$.

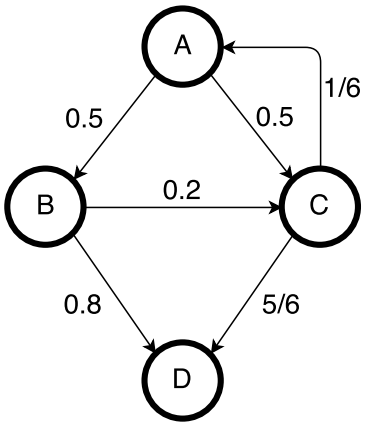

Figure 5.5: Simple state model with transition probabilities

Please consider the example shown in figure 5.5. This model is a model in which the final state can always be reached after an infinite number of runs. Applying definition 9 yields the results as shown in figure 5.6. In this result, the highest value (1) is highlighted in green, the lowest value (0) is highlighted in red. The values in between scale accordingly.

	Initial state	A	B	C	D	Final state
Initial state	1	1	0.54545455	0.6	1	1
A	0	1	0.54545455	0.6	1	1
B	0	0.03333333	1	0.2	1	1
C	0	0.16666667	0.09090909	1	1	1
D	0	0	0	0	1	1
Final state	0	0	0	0	0	1

Figure 5.6: Results of reachability calculations on the model of figure 5.5.

These results support the claim that the final state (state D) can always be reached in an infinite number of runs. Furthermore, it shows that state B can only be reached from the initial state (which is state A), state A and in very rare occasions from state C. More or less the same goes for state C.

It is good to know that a certain state is always reachable (has value 1.0), since this means that no attention needs to be paid to that state anymore and this provides the usability engineer with the most freedom in generating a scenario. It is also interesting to know that a certain state is not reachable anymore, especially when a certain choice in the scenario is made that excluded multiple other states.

However, the exact probability for reachability values between 0.0 and 1.0 is not interesting. A test scenario is created by the usability engineer, so if a certain state is to be included the usability engineer can always steer the scenario towards this state. Furthermore, providing reachability probabilities could be misleading when taking transition probabilities into account: it could be that a longer trace to a state gets a higher reachability score than a direct trace to that state. See for example figure 5.7, in which the reachability value of reaching C from A would be higher by visiting two extra states (B and D) in stead of going directly from A to C.

Because of this, we will abstract from values that are not 0.0 or 1.0, by giving the usability engineer the indication that this state is "sometimes reachable", meaning that he does not have the full freedom of making other choices in between, which a reachability of 1.0 would have. The "sometimes reachable" value implies that he will have to pay more attention if he wants to reach specific states.

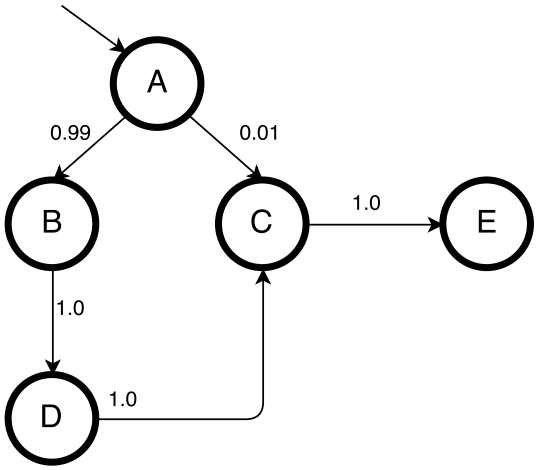

Figure 5.7: Simple state model with transition probabilities, showing potential misleading of reachability

5.5 Evaluation and suggestions for actual tool

Some of the proposed measures are implemented in a testing tool and can be used on the real data set. These measures include the transition probabilities and the reachability. Completeness is not yet included, both the unweighted, weighted and exclusion of different test cases due to state equivalence / partitioning.

Also, since no tool with a graphical user interface is available yet, some suggestions are made for including these measures in a scenario-generation tool. These measures can be useful for process mining tools in general as well, as long as the goal of this use is to create one specific trace and compare it to the model.

5.5.1 Evaluation with real data

In order to test these measures on the real data, a composite state log has been created from artifacts "CRF", "Imaging" and "Machinestate". The composite model combines the different artifact states into one state, resulting in a non-artifact-centric state model. This model is a real spaghetti-diagram, as can be seen in figure 5.8. This is not a problem, since the composite model is only used for calculation of the measures in the background and does not have to be shown to the user.

This model has been made with the same data as described in chapter 4, the 65 cases of type "PTA/Stent Bekken-benen". It consists of 137 unique states. The reference traces that are used for calculating the transition probabilities are the 65 actual traces that are observed in the field. Please refer to table A.1 in Appendix A for the results.

The highest likelihood score of the traces in the log is 0.000207937267403855 and the lowest likelihood score is 0.0 . Further investigation into these scores show that the number of states is almost directly linked to the value of the likelihood score and the last case has too many states, which leads to a score of 0.0 due to round-off differences and the precision of calculations. There are some deviations, but generally the higher the number of states, the lower the likelihood score.

The likelihood scores diminish very quickly. Looking at the trace with the highest score gives us $\sqrt[14]{2.079 * 10^{-4}} \approx 0.5458$, so the average transition probability for this specific trace was quite high. Nevertheless, the resulting likelihood score is already relatively small.

Since the results of the reachability analysis are mainly interesting during the generation of a

Figure 5.8: Composite state model of artifacts “CRF”, “Imaging” and “Machinestate”

scenario, it is difficult to show results on this measure. The initialization of the matrix with all the possible values is something that is shown in figure A.1 in Appendix A. This matrix shows all the possible reachability probabilities, where red represents a probability of 0, green a probability of 1, and orange all the values in between.

There are quite some states that cannot be reached from all other states. Knowing in advance that choosing a particular state will result in never being able to visit another is therefore still considered as a relevant measure for the generation of scenarios. Furthermore, there are only a few states that are green and can be reached from any other state. Most of the states can be reached, but not from every other possible state.

5.5.2 Suggestions for implementing these measures in a usable tool

Data visualization is a complex, but important, step of using data to provide valuable insights. Articles like [72] show that a lot of quality aspects determine whether visualizations are good or not. This is also shown by [73], who emphasize the importance of aesthetics on a good data visualization. Books, like [74], highlight that using a wrong visualization can dramatically “distort the “truth” ”.

Given this importance and the potential harm a bad visualization can do, the fact that our goal is not to describe a perfect visual tool, and simply the fact that the time to investigate different visualization techniques is not present, we will not come up with a visualization, but will only provide general relevant use cases for the proposed measures.

The measures for the user can be divided into two categories:

1. Measures that must be visible during the creation of the scenario
2. Measures that must be visible after a full scenario is created

Transition probabilities fall in both categories. During the generation of the scenario, it must be visible for the usability engineer what the possible next steps are and what their probability is. After a scenario is created, the likelihood score for the whole scenario must be displayed, to allow for comparison of scenarios. Also, it must be possible to see (in the actual model) where deviations from higher-value transition probabilities occur. If there is a state with two outgoing branches, one with probability 0.9 and the other with probability 0.1, and the branch with probability 0.1 is chosen, this must be shown at the end of the scenario generation. The threshold for showing a probability must be configurable.

Just like the transition probabilities, completeness of the scenario must be shown both during and at the end of the generation as well. During the generation of the scenario, it might be useful to see how many states are already covered. Also, superfluous tasks due to state equivalence, as discussed with the example in figure 5.4, must be shown and it must be the goal not to include multiple task from state equivalence partitions.

After the generation, the total coverage of the number of states (either weighted or unweighted) must be shown, both to allow for comparison of these numbers, but also to give an indication of the amount of test coverage this whole scenario generates. Furthermore, suggestions on how to improve this coverage must be given.

Finally, reachability is something that should only be displayed during the creation of a scenario, since it relates only to a specific state in which the scenario is right now. Reachability should give an indication into which path to take to get the highest possible completeness of the model, see the example described in figure 5.1.

5.5.3 Future work

The calculation of the transition probabilities currently does not take the length of traces and possible loops into account. Since a probability can be at most 1.0, and since the overall likelihood value of a trace is the product of the individual probabilities, the likelihood value can only remain the same (unlikely) or decrease when more states are added to the scenario. This is clearly visible in section 5.5.1, where the evaluation with real data shows that the likelihood values go down very quickly.

Some form of compensation for loops and the length of the traces is therefore desirable. This is something that we did not focus on and this should be further investigated.

Moreover, the memorylessness property that is currently assumed should be further investigated. Developing a real tool that allows usability engineers to generate scenarios could provide insight into whether this memorylessness property is perceived as a problem.

Furthermore, the completeness measures should still be implemented. Unweighted and weighted completeness are both trivial to implement. However, the partitioning of states, allowing for a reduction of the possible number of states, is something that could prove very important. The exact implementation should also be considered in more detail.

5.6 Conclusions and recommendations

In this chapter, we have provided insights into the current approach to generating usability testing scenarios and proposed a new approach to incorporate field-data into these protocols. An equality between these approaches is that the usability engineer is still in charge of this process. However, this usability engineer is guided by an interactive tool that gives suggestions of tasks to include in the scenario, based on the model generated in chapter 4.

In order to guide the usability engineer into making the right choices, three measures were proposed. These are: Transition probabilities, Completeness and Reachability.

Transition probabilities show how well the scenario reflects the choices that are contained in the model. Including this measure is important to say something about how realistic a scenario is. It is, however, recommended to think about the right visualization of this, especially when displaying an overall likelihood value for a scenario. Since the overall likelihood value is defined as the product of the individual transition probabilities, this number will go down very quickly. Furthermore, the assumed memorylessness property of this measure might be a limitation for the goals that Philips has, and will therefore need to be investigated in future work.

Completeness of a scenario says something about how many states are included in the scenario. Even though the goal of a usability scenario is not to provide coverage of all possible states, it is still recommended to include this measure, since it does say something about the test coverage of the usability scenario. Extending the completeness scenario with allowing state equivalence (either by using the data, or by using domain knowledge to define this equivalence) could drastically reduce the possible number of states of the tests. We recommend that Philips takes the time

to define these state equivalences / partitions of states carefully, since including them might save a lot of effort later.

Finally, reachability is a measure that can be used during the generation of a scenario and tells something about the probability of ever reaching state X from state Y. If this probability is 0, the usability engineer knows that he needs to take a few steps back if he wants this state to be included. If this probability is 1, he has all the freedom to define different activities in between. Anything in between means that he needs to pay attention during the generation of the scenarios.

This measure is used to balance potential contradictory results between transition probabilities and completeness, and makes sure that not only the local transition probabilities are optimized, but also the global ones.

These measures have shown to work when calculating a scenario based on a single measure, but the combination of them might generate contradictory results. Therefore, it is strongly recommended to only use the measures to guide the usability engineer into making the right choices, and not to generate automated testing scenarios. Respecting this recommendation will ensure that both data generated from domain knowledge and data generated from real-life data is included in the testing scenarios, making them as realistic as possible.

Chapter 6

Conclusions

In this thesis, we have come up with a way to generate field-based usability testing scenarios from relational database data. We have also used this data to get an insight in the complex workflow of the iXR machines under study. In order to achieve this, we have handled three approaches to the different research goals. Please refer to figure 6.1 for the graphical representation of these approaches.

In research goal 1, we have translated the relational data that Philips has into state-like process data that can be used for process mining. We have divided the data in different artifacts, mainly based on the component structure of the machines in the workflow under study, and we translated the available data to relevant state changes that can be linked to these artifacts. The main challenges lie in the nature of the data, which is very low-level and highly detailed. Including this in a model requires a lot of abstraction steps. We have tried to take some of these steps and translate this very low-level data into data that is of a higher level of abstraction. However, there is still a gap that needs to be bridged between the abstraction level of the actual models and the abstraction level at which the usability engineers (final users of the model) think.

Furthermore, the addition of a process instance to the data was discussed. A structural overview of possible process instances was given by starting "small" and carefully expanding the reach of these possible instances. This overview can also be used when applying process mining for a different goal than usability test scenario creation. For the purpose of this thesis, the process instance notion of 1 procedure was chosen, meaning that the model aims to generalize behaviour across different procedures.

Research goal 2 deals with the transformation of the event data (as generated by research goal 1) into the actual model that can be used for the generation of the testing scenarios. This is done by using the CSM miner plugin in ProM, which allows for artifact-centric process mining while also looking at interactions between the different artifacts.

A structural evaluation of the outcomes of the CSM miner has been described, but also the relevant findings that this miner has given into the process. Some of these findings were not expected. An example of this is that only 10% of the total workflow involves making images with the machines. Such a finding would not have been possible without using different data sources (CRF data and machine log data) and would presumably have been very difficult to find without using an artifact-centric mining approach that gives insight into the interactions between the different artifact, such as the CSM miner.

Finally, research goal 3 deals with the hybrid approach of using domain knowledge and insights from the data to generate realistic usability testing scenarios. The current process of generating these scenarios only focuses on the domain knowledge of the usability engineer, our approach also takes observations from real-life field data into account.

In order to provide these valuable insights, three measures have been defined. First of all, trans-

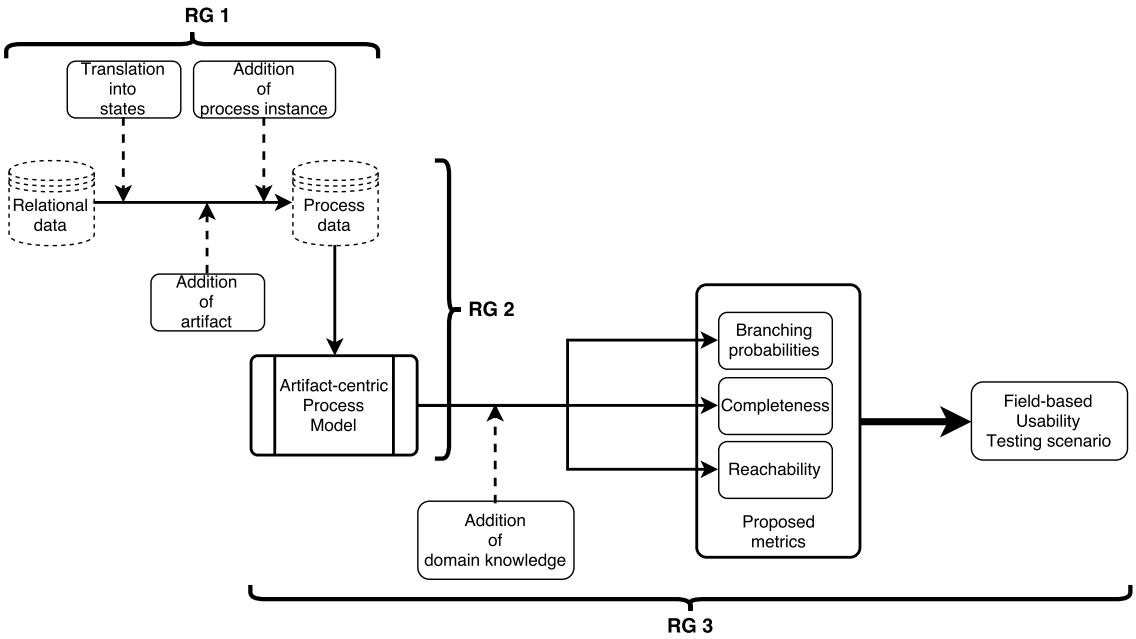

Figure 6.1: Graphical overview of the approach

ition probabilities show how well the scenario reflects the choices that are contained in the model. Including this measure is important to say something about how realistic a scenario is. Secondly, completeness of a scenario says something about how many states are included in the scenario. Achieving the highest possible completeness is not the main goal of a usability testing scenario, since this is simply not feasible due to the nature of the test (testing with a real participant and limited available time), but this measure remains an important measure when generating test scenarios, since it shows something about the total covered features in a test. Thirdly, reachability is a measure that can be used during the generation of a scenario and tells something about the probability of ever reaching state X from state Y. This measure is used to balance potential contradictory results between transition probabilities and completeness, and makes sure that not only the local transition probabilities are optimized, but also the global ones.

These measures are specifically tuned for generating usability testing scenarios. However, they can be used in a more general process mining setting where the user is interested in comparing a certain trace through a model to that actual model and seeing how well this trace reflects the choices that are obtained in the model and how complete the trace is.

Evaluation of the measures on real-life data showed the potential benefit of these measures. However, in order to make this approach work in a real-life setting, it would still be necessary to overcome the abstraction gap that was described at the beginning of this chapter.

Chapter 7

Limitations and Future Work

The limited time that was available for writing and executing the research in this thesis has lead to a few limitations of the work.

First of all, one of the most important limitations lies in the (lack of) preprocessing of the CRF data. As described in chapter 3, the CRF data showed some inconsistencies and needed to be preprocessed. However, there was no good way to perform the necessary preprocessing steps on the data in the time that was available. It is recommended to address this issue in future usages of the data.

Secondly, the limited time also did not allow to perform and test automated translation mechanisms that translate low-level data to higher level data. This could have been useful for incorporating the machine log data in the model.

Thirdly, not all necessary data was logged and could be used in the models. Please refer to table 3.2 for an overview of extra things that could have been useful. One of the main things to include in future models would be the location of the detector with regard to the patient. We expect that this would provide huge insights in the actual usage of the machine. Also, being able to tell which resource performs which tasks is expected to be of great value.

Furthermore, as underwritten by [39], there is an enormous difficulty of the gap that exists between sensor logs and actual tasks that are performed in process mining. This is something that was observed in this thesis as well. The level of abstraction from usability engineers (or end-users in general) differs very much from the level of abstraction used in the models. This is something that should be investigated in future work, to allow for better usages of these models.

Moreover, the models are currently based on data from one hospital and only look at one clinical procedure. In order to truly generalize across different hospitals and procedures, it is necessary to look at different hospitals and procedures as well. This should be included in future models.

Also, the measures that were presented need to be implemented in an interactive tool in order to test them properly with real users. This could also look whether the current limitations of the measures (for example the memorylessness property that is assumed for transition probabilities) is a problem for Philips. [75] have described a tool for scenario generation in requirements elicitation. Future work might look into this tool for inspiration on how to include our specific measures.

Finally, we have defined a few future usages of the model (more specifically process mining on the available data in general). These usages are: (for a description, see the rest of the chapter)

- Impact analysis within the workflow
- Replaying workflow in a simulated scenario
- Taking a role within the workflow in a simulated scenario
- Anomaly detection within the workflow
- Optimizing a workflow

Allowing to perform an impact analysis within the workflow is one of these new requirements. Given a certain workflow, what happens when we change something in the setup of the room or in the design of the machine? This could be very trivial changes, such as changing the controller of the machine from one side of the table to the other side of the table, but it could also be a more radical change, such as replacing an arc that cannot move with an arc that can move (this is of course a purely fictional example). Estimating (or preferably even calculating) the impact of these changes on the workflow before physically implementing the change (costing a lot of money and resources) could provide a huge competitive advantage to Philips.

Looking at replaying a workflow in a simulated scenario, the simulation environment must be able to execute the workflow model and replay what happened in this workflow. On top of this, the person interacting with the simulation must be able to take the role of a certain actor in the model. He/she must then be able to take over the steps that the actor in the workflow model would normally undertake and the simulation environment must wait for the right steps to be performed before continuing. These requirements are heavily related to the simulation environment itself, but could still be of great use for Philips.

Anomaly detection within the workflow relates to the improvement of the workflow in the real-life world itself. When certain activities differ a lot from the usual path, an alert could be risen, either to a doctor directly, or to a field service engineer that can determine whether further actions are required. An example of this could be the setting of the dose: when the intensity of the dose is too high for too long, this could result in damage to the patient. However, sometimes it is required to use high dose for a short period of time. If the dose intensity can be included in the workflow model, it could be possible to detect deviations from the normal pattern of dose intensity, so a hospital could be warned when their dose settings are too high for a longer period of time.

The last future usage is about optimization of the workflow. When a full view of a best-in-class workflow is present, Philips could advise their customers on the optimal way of designing their procedure rooms to get the best efficiency. Philips could also compare a workflow of a specific hospital to the typical workflow and see where the hospital deviates, and where they can gain. This would provide a competitive advantage to Philips as well, since they could add a new unique selling point (USP) in the form of room design and consultancy for the new machine. Examples of these improvements could be changing the setup of a room, use certain features of the machine that improve efficiency etc.

Bibliography

[1] What is interventional radiology? — bsir. http://www.bsir.org/patients/what-is-interventional-radiology/. (Accessed on 07/06/2017). 1

[2] Philips healthcare — interventional x-ray. http://www.philips.co.uk/healthcare/solutions/interventional-xray. (Accessed on 04/28/2017). 1

[3] Franchise tax board homepage. https://www.ftb.ca.gov/. Business Process Management Center of Excellence Glossary (Accessed on 10/27/2009). 2

[4] Iso 12052:2006 - health informatics – digital imaging and communication in medicine (dicom) including workflow and data management. https://www.iso.org/standard/43218.html. 2

[5] A. Brosens. Clinical workflow modelling. Confidential company presentation, March 2017. 2

[6] Usability for medical devices: A new international standard: Iso/iec 62366. http://www.userfocus.co.uk/articles/ISO62366.html. (Accessed on 07/01/2017). 2, 3, 47

[7] Usability testing — usability.gov. https://www.usability.gov/how-to-and-tools/methods/usability-testing.html. (Accessed on 05/01/2017). 2

[8] Eur-lex - 32017r0745 - en - eur-lex. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R0745. (Accessed on 06/28/2017). 3

[9] Wil van der Aalst. Data science in action. In *Process Mining*, pages 3–23. Springer, 2016. 4, 5, 14, 15, 27

[10] Maikel L van Eck, Natalia Sidorova, and Wil MP van der Aalst. Discovering and exploring state-based models for multi-perspective processes. In *International Conference on Business Process Management*, pages 142–157. Springer, 2016. 5, 15, 32

[11] Wil MP Van Der Aalst, Paulo Barthelmess, Clarence A Ellis, and Jacques Wainer. Proclets: A framework for lightweight interacting workflow processes. *International Journal of Cooperative Information Systems*, 10(04):443–481, 2001. 5, 15, 16

[12] Viara Popova, Dirk Fahland, and Marlon Dumas. Artifact lifecycle discovery. *International Journal of Cooperative Information Systems*, 24(01):1550001, 2015. 5, 15, 16

[13] Maikel L van Eck, Natalia Sidorova, and Wil MP van der Aalst. Guided interaction exploration in artifact-centric process models. *arXiv preprint arXiv:1706.02109*, 2017. 5, 15, 30, 32, 34, 35, 37, 38, 44

[14] View details of philips azurion 7 with 12" flat detector. http://www.philips.co.uk/healthcare/product/HCNCVD003/azurion-7-with-12-inch-detector. (Accessed on 04/28/2017). vi, 8

[15] IEC PAS. 61910-1 radiation dose documentation part 1: Equipment for radiography and radioscopy, 2007. 7

[16] Enterprise location services — clinical-grade visibility. `http://www.centrak.com/solution-overview/`. (Accessed on 03/28/2017). 8

[17] Xixi Lu, Marijn Nagelkerke, Dennis van de Wiel, and Dirk Fahland. Discovering interacting artifacts from erp systems. *IEEE Transactions on Services Computing*, 8(6):861–873, 2015. 15

[18] Wil MP van der Aalst. Extracting event data from databases to unleash process mining. In *BPM-Driving innovation in a digital world*, pages 105–128. Springer, 2015. 15, 27, 28

[19] Arik Senderovich, Andreas Rogge-Solti, Avigdor Gal, Jan Mendling, and Avishai Mandelbaum. The road from sensor data to process instances via interaction mining. In *International Conference on Advanced Information Systems Engineering*, pages 257–273. Springer, 2016. 15, 27

[20] Carlos Fernández-Llatas, José-Miguel Benedi, Juan M García-Gómez, and Vicente Traver. Process mining for individualized behavior modeling using wireless tracking in nursing homes. *Sensors*, 13(11):15434–15451, 2013. 15

[21] Carlos Fernandez-Llatas, Aroa Lizondo, Eduardo Monton, Jose-Miguel Benedi, and Vicente Traver. Process mining methodology for health process tracking using real-time indoor location systems. *Sensors*, 15(12):29821–29840, 2015. 15

[22] R Miclo, F Fontanili, G Marques, P Bomert, and M Lauras. Rtls-based process mining: Towards an automatic process diagnosis in healthcare. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, pages 1397–1402. IEEE, 2015. 15

[23] Marvin L Brown and John F Kros. Data mining and the impact of missing data. *Industrial Management & Data Systems*, 103(8):611–621, 2003. 16

[24] Jerzy Grzymala-Busse and Ming Hu. A comparison of several approaches to missing attribute values in data mining. In *Rough sets and current trends in computing*, pages 378–385. Springer, 2001. 16

[25] Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. *Classification, clustering, and data mining applications*, pages 639–647, 2004. 16

[26] Dorian Pyle. *Data preparation for data mining*, volume 1. Morgan Kaufmann, 1999. 16, 21

[27] S Sridevi, S Rajaram, C Parthiban, S SibiArasan, and C Swadhikar. Imputation for the analysis of missing values and prediction of time series data. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 1158–1163. IEEE, 2011. 16, 21

[28] Wil MP Van der Aalst. Getting the data. In *Process Mining*, pages 95–123. Springer, 2011. 16

[29] Christian W Günther and Wil MP van der Aalst. *Mining activity clusters from low-level event logs*. Beta, Research School for Operations Management and Logistics, 2006. 16, 26

[30] Felix Mannhardt, Massimiliano de Leoni, Hajo A Reijers, Wil MP van der Aalst, and Pieter J Toussaint. From low-level events to activities-a pattern-based approach. In *International Conference on Business Process Management*, pages 125–141. Springer, 2016. 16, 26

[31] Chao Chen, Barnan Das, and Diane J Cook. A data mining framework for activity recognition in smart environments. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 80–83. IEEE, 2010. 16

[32] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012. 16

[33] Tao Gu, Shaxun Chen, Xianping Tao, and Jian Lu. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data & Knowledge Engineering*, 69(6):533–544, 2010. 16

[34] Maja Stikic and Bernt Schiele. Activity recognition from sparsely labeled data using multi-instance learning. *Location and context awareness*, pages 156–173, 2009. 16

[35] Supriyo Chatterjea, Erik Bresch, Evert van Loenen, Roel Cuppen, and Teun van den Heuvel. Rtls analytics toolkit 2.0 documentation, December 2015. Confidential company information. 20, 21

[36] Sarah Hale and Tift Mann. Examining the appropriateness of radial or femoral access: evidence from the rival trial and clinical practice. *Interventional Cardiology*, 4(6):675–687, 2012. 22

[37] Jean L Marx. Low-level radiation: just how bad is it? *Science*, 204(4389):160–164, 1979. 24

[38] Maikel L van Eck, Xixi Lu, Sander JJ Leemans, and Wil MP van der Aalst. Pmˆ 2: A process mining project methodology. In *International Conference on Advanced Information Systems Engineering*, pages 297–313. Springer, 2015. 27

[39] Francesco Leotta, Massimo Mecella, and Jan Mendling. Applying process mining to smart spaces: perspectives and research challenges. In *International Conference on Advanced Information Systems Engineering*, pages 298–304. Springer, 2015. 27, 46, 66

[40] Alfredo Bolt and Wil MP van der Aalst. Multidimensional process mining using process cubes. In *International Conference on Enterprise, Business-Process and Information Systems Modeling*, pages 102–116. Springer, 2015. 28

[41] Xixi Lu. Artifact-centric log extraction and process discovery. *Unpublished masters thesis, Eindhoven University of Technology*, 2013. 30

[42] E Nooijen. Artifact-centric process analysis–process discovery in erp systems. *Unpublished masters thesis, Eindhoven University of Technology*, 2012. 30

[43] E YaJun, NengShu He, Yi Wang, and HaiLun Fan. Percutaneous transluminal angioplasty (pta) alone versus pta with balloon-expandable stent placement for short-segment femoropopliteal artery disease: a metaanalysis of randomized trials. *Journal of vascular and interventional radiology*, 19(4):499–503, 2008. 31

[44] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. 32

[45] Task scenarios for usability testing. `https://www.nngroup.com/articles/task-scenarios-usability-testing/`. (Accessed on 06/22/2017). 48

[46] Scenarios — usability.gov. `https://www.usability.gov/how-to-and-tools/methods/scenarios.html`. (Accessed on 06/22/2017). 48

[47] PVR Murthy, PC Anitha, M Mahesh, and Rajesh Subramanyan. Test ready uml statechart models. In *Proceedings of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools*, pages 75–82. ACM, 2006. 49, 51

[48] Anne Rozinat and Wil MP van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 2008. 51

[49] Fábio Bezerra and Jacques Wainer. Algorithms for anomaly detection of traces in logs of process aware information systems. *Information Systems*, 38(1):33–44, 2013. 51

[50] Valdivino Santiago, Ana Silvia Martins Do Amaral, Nandamudi L Vijaykumar, Maria de Fatima Mattiello-Francisco, Eliane Martins, and Odnei Cuesta Lopes. A practical approach for automated test case generation using statecharts. In *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, volume 2, pages 183–188. IEEE, 2006. 51

[51] Valdivino Santiago, Nandamudi Lankalapalli Vijaykumar, Danielle Guimarães, Ana Silvia Amaral, and Érica Ferreira. An environment for automated test case generation from statechart-based and finite state machine-based behavioral models. In *Software Testing Verification and Validation Workshop, 2008. ICSTW'08. IEEE International Conference on*, pages 63–72. IEEE, 2008. 51

[52] Mark Utting, Alexander Pretschner, and Bruno Legeard. A taxonomy of model-based testing approaches. *Software Testing, Verification and Reliability*, 22(5):297–312, 2012. 51

[53] Ilya Wagner, Valeria Bertacco, and Todd Austin. Stresstest: an automatic approach to test generation via activity monitors. In *Proceedings of the 42nd annual Design Automation Conference*, pages 783–788. ACM, 2005. 51

[54] Philip Samuel and Anju Teresa Joseph. Test sequence generation from uml sequence diagrams. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pages 879–887. IEEE, 2008. 51

[55] Andreas Heinecke, Tobias Brückmann, Tobias Griebe, and Volker Gruhn. Generating test plans for acceptance tests from uml activity diagrams. In *Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on*, pages 57–66. IEEE, 2010. 51

[56] Emanuela G Cartaxo, Francisco GO Neto, and Patricia DL Machado. Test case generation by means of uml sequence diagrams and labeled transition systems. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 1292–1297. IEEE, 2007. 51

[57] Sebastian Siegl, Winfried Dulz, Reinhard German, and Gerhard Kiffe. Model-driven testing based on markov chain usage models in the automotive domain. In *12th European Workshop on Dependable Computing, EWDC 2009*, pages 6–pages, 2009. 52

[58] Heon-Mo Koo and Prabhat Mishra. Functional test generation using design and property decomposition techniques. *ACM Transactions on Embedded Computing Systems (TECS)*, 8(4):32, 2009. 52

[59] Arya Adriansyah, Boudewijn F van Dongen, and Wil MP van der Aalst. Conformance checking using cost-based fitness analysis. In *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*, pages 55–64. IEEE, 2011. 52

[60] Andreas Rogge-Solti and Gjergji Kasneci. Temporal anomaly detection in business processes. In *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, pages 234–249, 2014. 52, 54

[61] Kristof Böhmer and Stefanie Rinderle-Ma. Multi-perspective anomaly detection in business process execution events. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 80–98. Springer, 2016. 52

[62] Ashish Sureka. Kernel based sequential data anomaly detection in business process event logs. *arXiv preprint arXiv:1507.01168*, 2015. 52

[63] Mariëlle Ida Antoinette Stoelinga. *Alea jacta est: verification of probabilistic, real-time and parametric systems.* [Sl: sn], 2002. 52

[64] Gianluigi Greco, Antonella Guzzo, Luigi Pontieri, and Domenico Sacca. Discovering expressive process models by clustering log traces. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1010–1027, 2006. 52

[65] Lijie Wen, Jianmin Wang, Wil MP van der Aalst, Biqing Huang, and Jiaguang Sun. A novel approach for process mining based on event types. *Journal of Intelligent Information Systems*, 32(2):163–190, 2009. 52

[66] W MP Van Der Aalst, Vladimir Rubin, H MW Verbeek, Boudewijn F van Dongen, Ekkart Kindler, and Christian W Günther. Process mining: a two-step approach to balance between underfitting and overfitting. *Software and Systems Modeling*, 9(1):87–111, 2010. 52

[67] Anne Rozinat and Wil MP Van der Aalst. Conformance testing: Measuring the fit and appropriateness of event logs and process models. In *Business Process Management Workshops*, volume 3812, pages 163–176. Springer, 2005. 52

[68] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004. 52

[69] Conrado Daws and Stavros Tripakis. Model checking of real-time reachability properties using abstractions. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 313–329, 1998. 52

[70] Gianpiero Cabodi, Paolo Camurati, and Stefano Quer. Improved reachability analysis of large finite state machines. In *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pages 354–360. IEEE Computer Society, 1997. 52, 53

[71] Dave Parker. Lecture 2: Discrete-time markov chains. In *Probabilistic Model Checking*. University of Oxford, Department of Computer Science, 2011. 53, 58

[72] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011. 61

[73] Nick Cawthon and Andrew Vande Moere. The effect of aesthetic on the usability of data visualization. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 637–648. IEEE, 2007. 61

[74] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications.* CRC Press, 2010. 61

[75] Jae Eun Shin, Alistair G Sutcliffe, and Andreas Gregoriades. Scenario advisor tool for requirements engineering. *Requirements Engineering*, 10(2):132–145, 2005. 66

Appendix A

Case ID	Likelihood	Number of states
PP5768829851	2.08E-04	14
PP5818191356	1.82E-04	16
PP5797154438	4.84E-05	15
PP5799054244	3.03E-05	16
PP5797285118	2.77E-06	18
PP5811140795	3.97E-08	17
PP5811198504	3.12E-08	14
PP5792920502	1.85E-09	21
PP5811059019	1.60E-12	19
PP5811008134	1.02E-12	14
PP5751492458	5.48E-18	100
PP5820599517	1.16E-18	28
PP5786981655	1.51E-24	180
PP5788048515	8.68E-28	124
PP5808563894	7.29E-32	221
PP5785930809	1.15E-33	222
PP5811909020	7.77E-38	305
PP5743627572	4.20E-38	167
PP5816363535	3.48E-38	336
PP5828323057	1.44E-39	316
PP5820792008	1.18E-39	200
PP5774768825	6.95E-44	337
PP5762596640	1.03E-44	273
PP5823193541	7.71E-46	352
PP5772983444	5.79E-49	350
PP5790328947	3.80E-49	353
PP5816166885	1.73E-49	376
PP5790252165	6.06E-50	379
PP5779098235	1.16E-50	322
PP5767821715	8.24E-51	330
PP5780921113	4.28E-51	376
PP5774817187	1.42E-51	248
PP5747913797	2.54E-53	639
PP5749814979	7.48E-58	410
PP5790469045	1.41E-58	314
PP5756652447	3.19E-59	372
PP5829203874	9.24E-61	372
PP5817038734	4.55E-62	349

PP5814493169	6.64E-63	442
PP5756703739	3.97E-63	214
PP5756552731	3.73E-70	431
PP5799986854	2.67E-73	495
PP5761730301	1.02E-74	499
PP5754034300	4.21E-76	539
PP5787987744	2.57E-76	370
PP5745486447	7.67E-77	585
PP5757613366	2.19E-79	592
PP5786851272	3.58E-81	637
PP5791185423	1.63E-81	652
PP5742755087	3.60E-82	530
PP5761778819	1.13E-84	371
PP5808395743	1.53E-86	672
PP5742803532	1.95E-91	654
PP5762774918	1.69E-93	788
PP5820518405	4.52E-95	982
PP5827503578	8.86E-96	710
PP5775705650	8.89E-103	750
PP5828585049	4.05E-103	883
PP5780061664	2.40E-113	697
PP5786028548	3.66E-121	876
PP5829328407	5.84E-135	1260
PP5741923301	5.33E-148	810
PP5828261668	2.79E-154	1139
PP5756770046	1.20E-168	1614
PP5787746740	0	3220

Table A.1: Results of running the branching probabilities algorithm on 65 cases, looking at artifacts “CRF”, “Imaging” and “Machinestate”. Sorted by likelihood value in descending order.

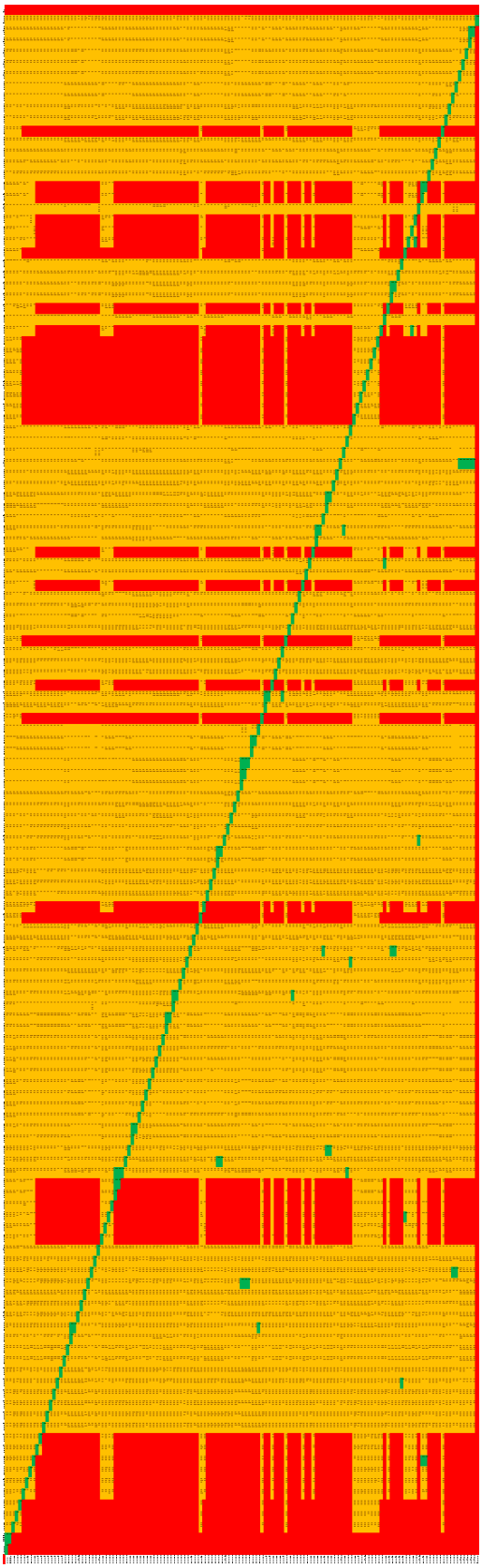

Figure A.1: Initialization of reachability matrix for artifacts “CRF”, “Imaging” and “Machinestate”. Rotated 90 degrees. Red = 0, green = 1, orange = in between