Technical University of Denmark
Department of Applied Mathematics and Computer Science

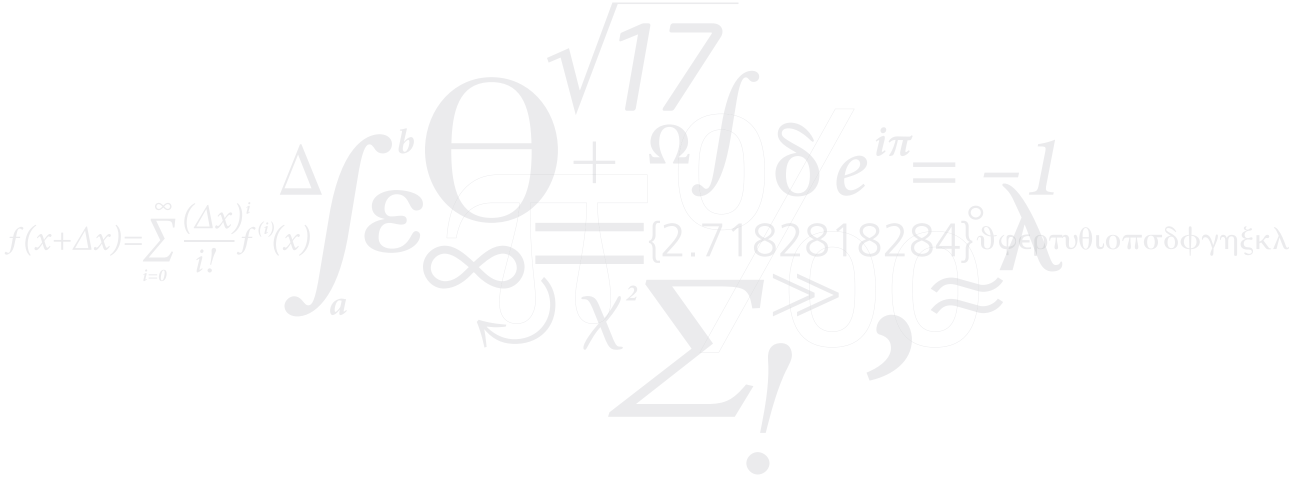
M.Sc. Thesis

Investigating The Use Of DCR Monocle For Creating Process Models Using Eye Tracking

January 2019

Author:
Jon Buch-Lorentsen

Supervisors:
Barbara Weber
Per Bækgaard
Andrea Burattin



Technical University of Denmark
Department of Applied Mathematics and Computer Science Richard Petersens Plads,
building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Abstract

Monocle is a novel tool developed to facilitate translation between natural language and declarative process modelling. The tool aims to help users create and maintain process models in the DCR environment. This thesis investigates Monocle by examining how users engage with the tool to create models. This has been done through an exploratory study consisting of two experiments with a total of 17 participants. To understand how users engage with the tool eye tracking has been used to gain insight to users viewing patterns when creating process models with the tool. The eye tracking data has been analysed using process mining to discover process maps showing how the participants gaze transition between the areas of interest in the interface. Verbal data has been recorded to complement the eye tracking data. This was done through semi-structured interviews that were analysed with qualitative coding.

The results show that the tool is useful for certain aspects of model creation but also have weaknesses. The tool shows great potential for creating activities and being used for getting started. However the results indicate that the tool is too cumbersome for adding constraints and relations and that the benefits of using it are not noticeable. It is however also found that this might be influenced by the experience of the user and the complexity of the process. Furthermore it is found that users might be adding undefined constraints to the model when using the modeller.

Preface

This thesis was carried out in the period August 1, 2018 to January 1, 2019 at the Department of Applied Mathematics and Computer Science at The Technical University of Denmark. The thesis was supervised by Professor Barbara Weber, Associate Professor Per Bækgaard and Assistant Professor Andrea Burattin. Additionally PhD student Amine Abbad Andaloussi assisted in the supervision. The thesis is part of the authors M.Sc. in Digital Media Engineering. This report serves as documentation of the work which has been done.

Content

1 Introduction

A critical aspect of any software is to design for the end user. It is therefore advantageous for developers to understand how users engage with their program. This knowledge can help them gain insight to improve the software and give inspiration for future implementations. Designing software tools, such as advanced case management systems, is a non-trivial task. To design well the developers must understand how their program is being used. A well made software tool can ease the workload of the user an enhance the quality of what they are creating with the application. Complex software therefore requires a deep investigation to get a clear grasp of how users are operating. In such softwares it is not just about designing a slick interface but about understanding which features can empower the users.

This thesis has been an investigation into a novel tool made for building process model. This tool is called Monocle and was made for the software DCR graphs. DCR graphs is an Industrial case management tool made for case workers without a background in process modelling[1]. DCR graphs is based on a flexible but complex modelling language, and for this reason it is at the core of DCR graphs to develop an intuitive and understandable tool[2]. DCR are aiming to bring a powerful modelling tool to domain experts, to do so they must implement the right features for making it comprehensible to the users. A recent tool added to DCR is aimed at addressing this. This tool is called Monocle. The tool gives the ability to create bi-directional linkage between textual process descriptions and the process models. This adds traceability and a more pronounced connection between the description and the model[3]. However this feature also requires the modeller to take effort and create these connections. Monocle creates a new way of modelling that is an addition to DCRs already established method of modelling. This creates some potential new dynamics of how users chooses to model.

In this thesis the Monocle tool is being investigated to understand how users engage with Monocle and how the interplay in the software is influenced by this new feature. It also investigates why users engage with Monocle in the discovered manner and what the benefits and challenges of Monocle are. To investigate these questions an exploratory study is conducted to examine how users create models. Eye tracking is used alongside interviews to collect data about their usage and experience. A novel method is then applied where process mining is used to analyse the eye tracking data to gain insight to users viewing patterns

1.1 Research Questions

The aim of this study is to investigate how Monocle is being used and why it is being used in the apparent manner. To drive the investigation two research questions has been established.

1.1.1 Research question 1

The main question to investigate is how Monocle tool is being used and how it is being used in relation to the remaining interface. This is done to understand which functionalities of Monocle is used and which functionalities of the remaining interface is used during modelling.

RQ1: How does users engage with a modelling task using Monocle tool?

1.1.2 Research question 2

Secondly it is interesting to understand why Monocle and the modeller are being used in the apparent manner. To examine this the benefits and challenges of using Monocle can be used to understand why users engage the way they do.

RQ2: What are the benefits and challenges of using Monocle?

For the remainder of this report the two research questions will be refer to as RQ1 and RQ2

1.2 Structure of Report

Chapter 2 will provide background knowledge and theory for the project. It will present relevant theory for the methods used and present the software under examination. Chapter 3 will go over the approach taken for designing an experiment. Chapter 4 and 5 will go over each of the conducted experiment in detail. Chapter 6 explains how the raw data was prepared and how it was analysed. Chapter 7 presents the results from the analysis. Chapter 8 will discuss the results and the methods used and suggest future work for the project. Chapter 9 will conclude and summarise the project.

2 Background

To delve further into the project it is necessary to investigate the underlying theory. This will be presented in this chapter to create a foundation for the project. The background presented is first business process modelling as it is a fundamental aspects of this project. Secondly the software DCR graphs will be examined to get an understanding of the software in focus. Finally the methods and tools for gathering and analysing data will be examined. These methods are eye tracking, process mining and qualitative research. Each topics will be presented in light of their relevance to this project.

2.1 Business Process Modelling

Business process modelling (or short process modelling) plays a central role for this project as the software to be investigated is a process modelling tool but also because process mining, used to evaluate the software, is related to process modelling.. The basics of process modelling are therefore relevant for understanding and diving deeper in both areas.

A process model is a formalised model representation of a business process[4]. The model is represented with a collection of process activities which are connected to achieve a common goal[4]. An example of a process model can be seen in figure 2.1. *Come up with an idea*, *write project proposal* and *refine idea* are examples of activities and the arrows are relations between them.

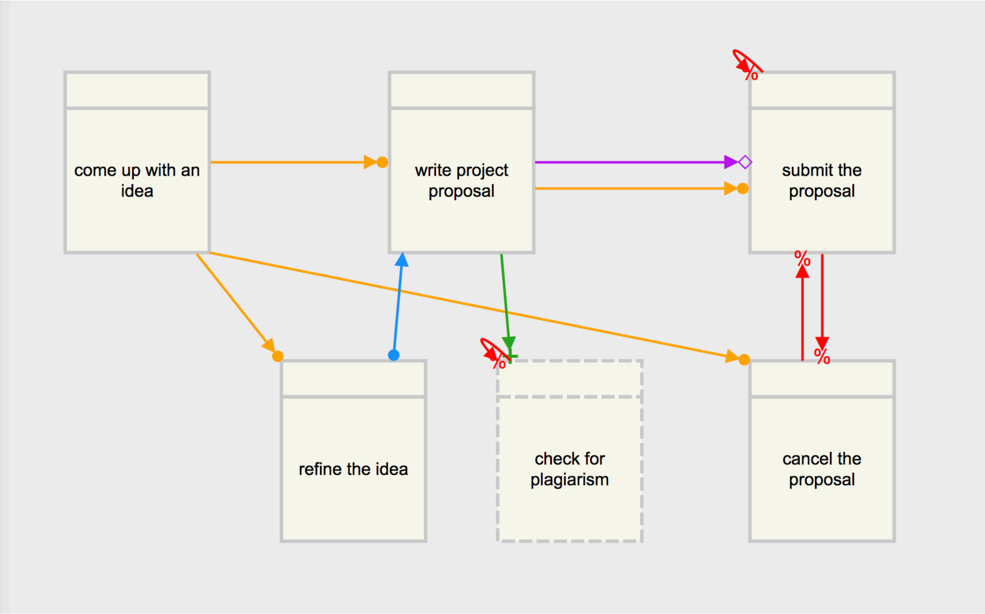


Figure 2.1: *A process model created in the DCR environment.*

The type of process modelled can be anything but as the name indicate the tool is often used for business processes or for software processes. Process models can be created by experts who investigates the business aspect to be mapped and then develops a model from this knowledge[5]. Process models can also be created in other ways such as automatic discovery which will be discussed in section 2.4. Process models can help people understand the process of businesses and be used to improve performance. It can also be used to simulating alternatives to the current process[4]. A process model must often capture several ways in which sequences of activities can be executed. It is therefore necessary to have a process modelling language which can define relationships between activities. A sequence of activity executions is known as a trace[5]. There exists many different process modelling language. Each language follow unique notations and rules. Depending on the language used models may look and behave differently and is therefore non-trivial to select a language[5]. Popular languages include BPMN and EPC[6].

2.1.1 Declarative and Imperative Models

In process modelling there is distinction between imperative- and declarative modelling languages. DCR graphs is based on a declarative modelling language which is an important characteristics of the software (See section 2.3). The differentiation has roots in programming where imperative focuses on how a program operates and declarative focuses on what the program should achieve, without being concerned with how it is achieved[7]. In process modelling imperative is characterised by an 'inside-to-outside' approach[7]. This means that all alternatives to how the model should be executed needs to be explicitly defined. Declarative modelling is characterised by an 'outside-to-inside' approach[7]. In such models it is not specified exactly how the model should work but only the most essential characteristics are explained. By default such models can be executed in any way and restrictions and constraints are added to the model to define its boundaries[7]. Historically imperative languages have been most widely used and the major languages can all be defined as imperative[6]. However the value of using declarative languages is emerging within the research area of process modelling[7]. Declarative languages are more flexible than imperative and can therefore capture more possible traces with a simpler model. The disadvantage of declarative languages is a general poorer understandability, this means that reading and understanding a declarative model is often more difficult than an imperative model[8].

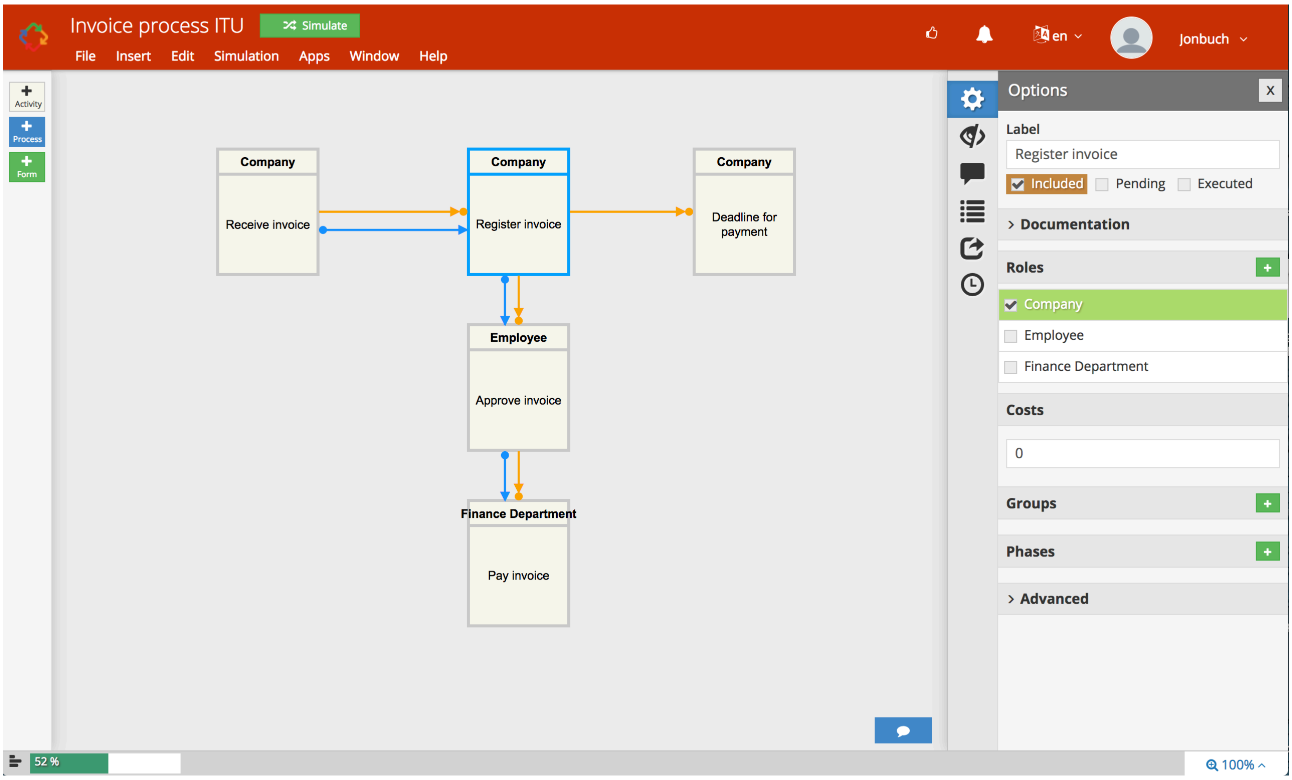
It has also been shown that the information in the process to be modelled can potentially influence which type of language is best suited[8]. A distinction can be made between sequential- and circumstantial information[8]. Sequential information is information about how one particular action leads to a particular outcome[8]. A sequential statement could be "Activity X must be executed before activity Y can be executed". Circumstantial information is information about the overall conditions and circumstances which produces an certain outcome[8]. It is therefore often about whether conditions have or have not occurred. An example of circumstantial information could be: “Whenever activity X has been executed the process can be terminated by executing activity Y”.

2.1.2 Hybrid Model Representations

As mentioned the issue of declarative modelling languages is that they are generally more difficult to read and understand than imperative languages. To alleviate this issue hybrid representations can be useful. A hybrid representation can be a combination of different modelling languages or it can be the combination of modelling notation with textual annotation[9]. DCR can be defined as a having a hybrid model representation as there are several notation methods to assist the user. One of these notation features is Monocle which will be explained in details in section 2.2.1.

2.2 DCR graphs

DCR Graphs (stands for Dynamic Condition Response graphs) is a software developed by the Process and System Models research group lead by Thomas Hildebrandt at the IT University of Denmark and Exformatics A/S[10]. The tool is created for knowledge workers to help digitise business processes and model workflows[11]. Specifically it is being developed as part of the EcoKnow project which is focusing on digitising case management processes in local governments[12]. By using the tool users are able to create process models through an interactive editor tool. The basic view of the DCR interface (called DCR Designer) can be seen in figure 2.2. The main window in figure 2.2 will be referred to as the modeller throughout this report. From the modeller a model can be created adding activities from the left column. The activities can then be name in the right side menu. Relations between the activities can be created by selecting an activity and then right clicking on another activity to create a relation to it. The modeller also supports other features, such as the ability to nest activities within other activities, however these will be disregarded as it has not been investigated in this project. Besides the modeller the software also features hybrid representation with the simulator for testing the model and Monocle for textual highlights. DCR is based on a declarative process modelling language. This is a central aspect of the tool since it is the belief of the developers that the flexibility can benefit case workers as it allows for more workflow executions in a model[2]. Processes often needs to be updated or changed throughout the lifetime of the process. Imperative languages such as BPMN are very rigid and require more process updates due to the nature of the language[13]. The flexibility of declarative notation can therefore limit the workload needed from the case workers. A high priority for DCR is therefore to compensate for the complexity associated with a declarative languages[2]. One way this is addressed is the hybrid representations; Monocle and The Simulator.



Options

Label

Register invoice

☑ Included □ Pending □ Executed

› Documentation

Roles +

☑ Company

□ Employee

□ Finance Department

Costs

0

Groups +

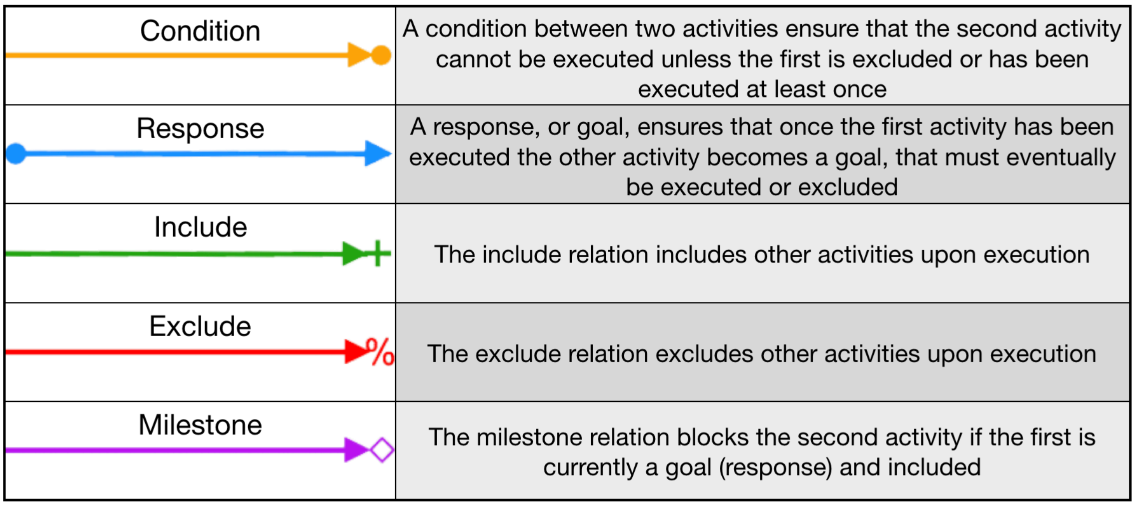
Phases +

› Advanced

Figure 2.2: *The basic interface view of DCR Designer.*

The DCR language's core notation is based on activities, activity stats, roles and relations[14]. Activities are the nodes which resembles actions performed in the model. An activity can either be in the state of included or excluded. While an activity is excluded it cannot be executed. Roles are the actors who perform the activities. A role can be attributed to each activity. Relations are the edges between activities that define the constraints and relationships between the activities. In figure 2.2 we can see each of the elements e.g. *Receive invoice* is an activity performed by the role *Company* this has two relations to the *Register invoice* activity. In DCR there are 5 main relations: condition, response, include, exclude and milestone. A relation must originate in one activity and point towards another activity. To define the relations this will be referred to as the first- and second activity

Condition is a commonly used relation which restricts the second activity from being executed unless the first activity is executed. The second activity can also become available if the first activity becomes excluded. The response relation makes the second activity pending after the first activity has been executed. When pending an exclamation marks appears on the activity which means the activity awaits execution. The include relation includes the second activity to the model when the first activity is executed. The exclude excludes the second activity from the model when the first is executed. The milestone relations restricts the second activity if the first activity is pending. The restriction can also be surpassed by excluding the pending activity. By default all activities can be executed and all traces are possible but by adding relations the process becomes restricted. In figure 2.3 each of the relation can be seen together with their description.



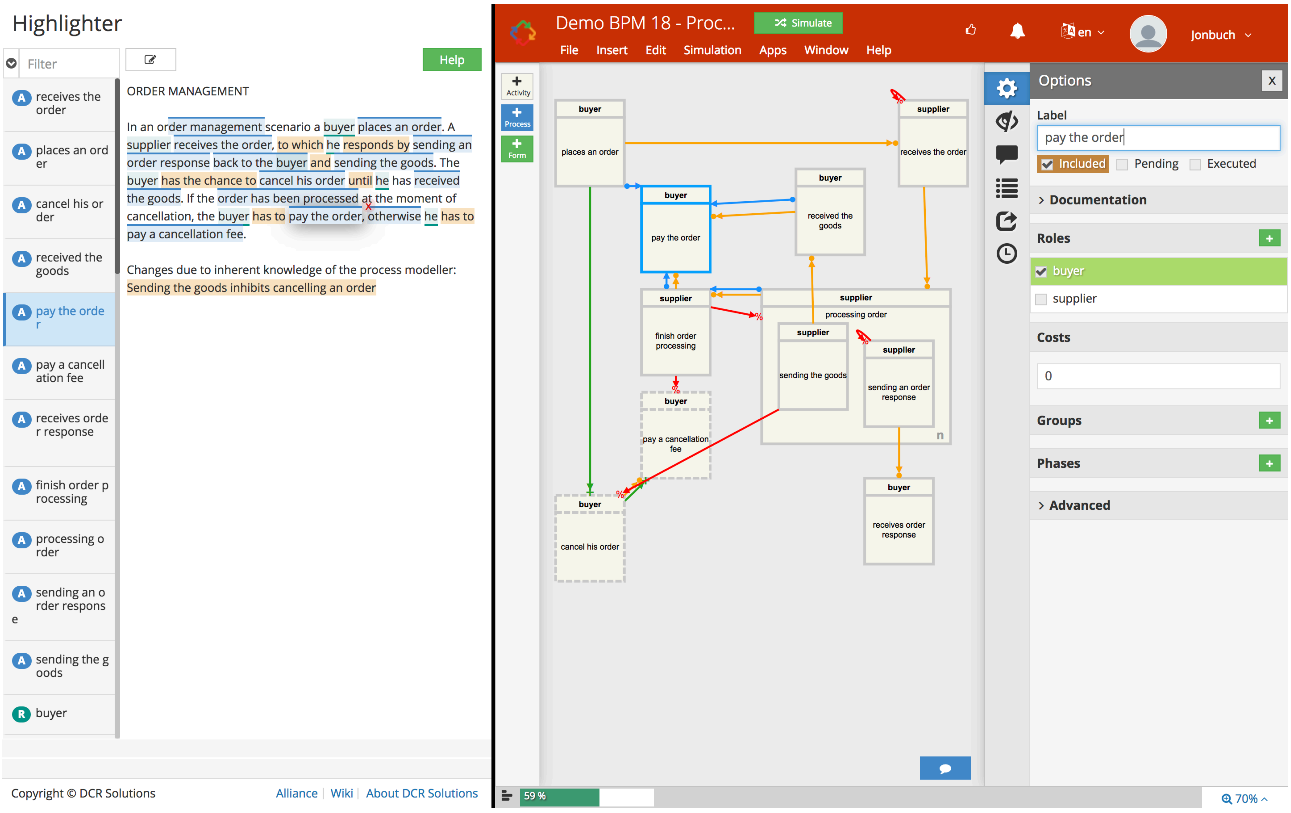
Condition	A condition between two activities ensure that the second activity cannot be executed unless the first is excluded or has been executed at least once
Response	A response, or goal, ensures that once the first activity has been executed the other activity becomes a goal, that must eventually be executed or excluded
Include	The include relation includes other activities upon execution
Exclude	The exclude relation excludes other activities upon execution
Milestone	The milestone relation blocks the second activity if the first is currently a goal (response) and included

Figure 2.3: *The 5 different relations together with their definitions taken from the DCR wikipedia*[1].

2.2.1 Monocle

Monocle (formerly known as Highlighter Tool) is a new feature introduced into the DCR graphs environment in the summer 2018. In an article concerning tool the developers state the following about the tool: “The purpose of the tool is to allow domain experts with some familiarity with the modelling notation, to easily construct and maintain declarative models”[3]. Monocle is a feature that can be opened in a window next to the modeller. Monocle can hold the textual process description and be used to build the model. By highlighting parts of the text activities, roles and relations can be directly generated into the model. This creates a permanent highlight of the text piece selected. Model parts created this way becomes bi-directionally linked to the highlight, this means that when selecting either the part in the model or the highlight in the text, the corresponding element also becomes selected. It is therefore a method to directly extract the process model and reduce the gap from the textual description to the model[3]. A view of the interface when having the tool opened can be seen in figure 2.4. With the tool it is also possible to link parts of the model that has already been created by assigning elements to text highlights. Defining the text to highlight is the job of user and can vary depending on their preferences. They can chose to highlight one word or a whole sentence for relation. Sometimes it can also be necessary to highlight text parts such as a comma when a constraint is only implicitly mentioned in the textual process.

[1] https://wiki.dcrgraphs.net/wiki/70/graph



ORDER MANAGEMENT

In an order management scenario a buyer places an order. A supplier receives the order, to which he responds by sending an order response back to the buyer and sending the goods. The buyer has the chance to cancel his order until he has received the goods. If the order has been processed at the moment of cancellation, the buyer has to pay the order, otherwise he has to pay a cancellation fee.

Changes due to inherent knowledge of the process modeller: Sending the goods inhibits cancelling an order

Label: pay the order

☑ Included □ Pending □ Executed

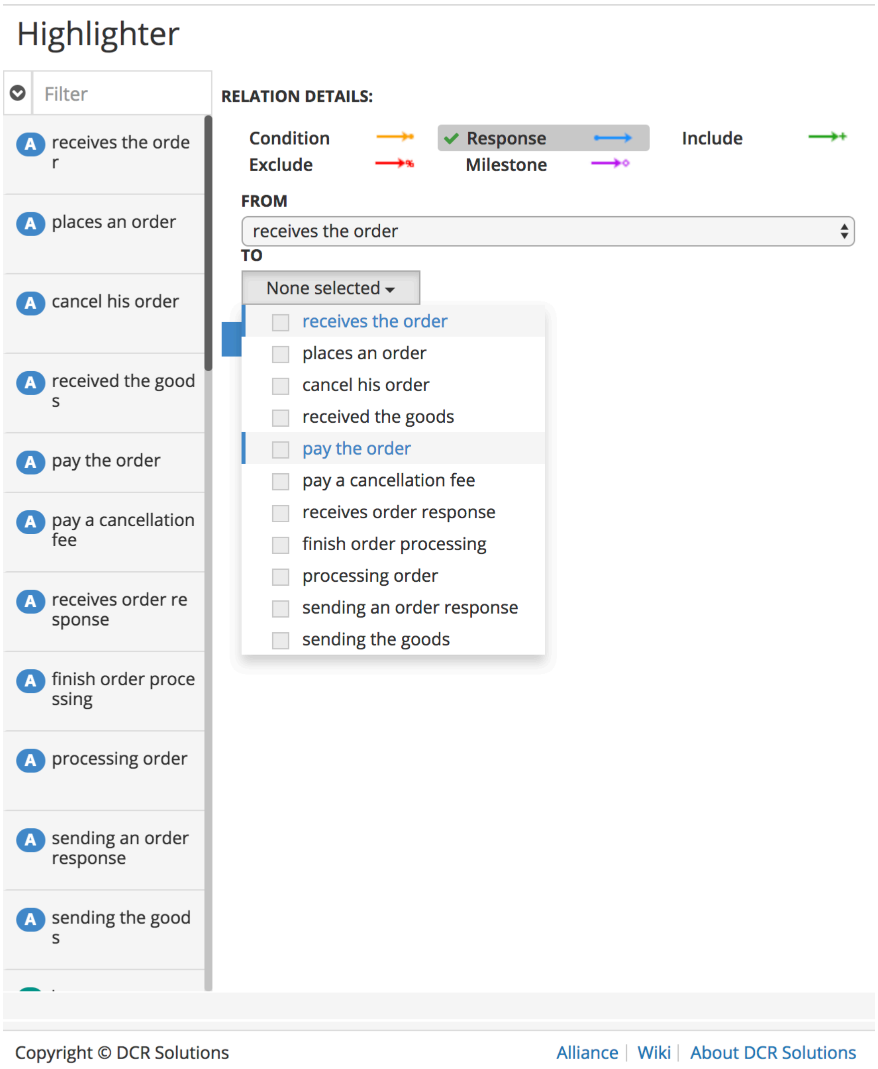
Roles: ☑ buyer □ supplier

Costs: 0

Figure 2.4: The interface with Monocle app opened. The highlighted words indicate the parts of the text that has been directly linked to the model.

When highlighting a text piece a pop-up menu appears where the user can select to either add it as an activity, role or relation. Creating activities in this manner makes the activity appear in the modeller. Roles become available in the menu to the right of the modeller. When adding relations there is an intermediate step where a new menu appears in place of the text. Here the user needs to select the relation she wants to add and the activities which the relation is coming from and going to. The screen view of this can be seen in figure 2.5. On the left in Monocle is the filter. This displays all the activities, roles and relations in the model. Using Monocle, the user is able to perform the same actions as with the modeller. Monocle therefore adds a choice to the user where she is able to define her own modelling strategy.

As mentioned Monocle is a tool developed to both help during model creation and model maintenance. A user may not create models in a sequential manner. Instead she might take an opportunistic approach which requires re-inspection and re-comprehension of his work[7]. It is imagined that Monocle is beneficial in such cases and potentially many others. However Monocle is a novel alignment for declarative modelling[3]. There is therefore little research directly related to be found about the impact of such a tool.



Highlighter

Filter

A receives the orde r

A places an order

A cancel his order

A received the good s

A pay the order

A pay a cancellation fee

A receives order re sponse

A finish order proce ssing

A processing order

A sending an order response

A sending the good s

RELATION DETAILS:

Condition — Response ✓ — Include

Exclude — Milestone

FROM

receives the order

TO

None selected ▾

□ receives the order

□ places an order

□ cancel his order

□ received the goods

□ pay the order

□ pay a cancellation fee

□ receives order response

□ finish order processing

□ processing order

□ sending an order response

□ sending the goods

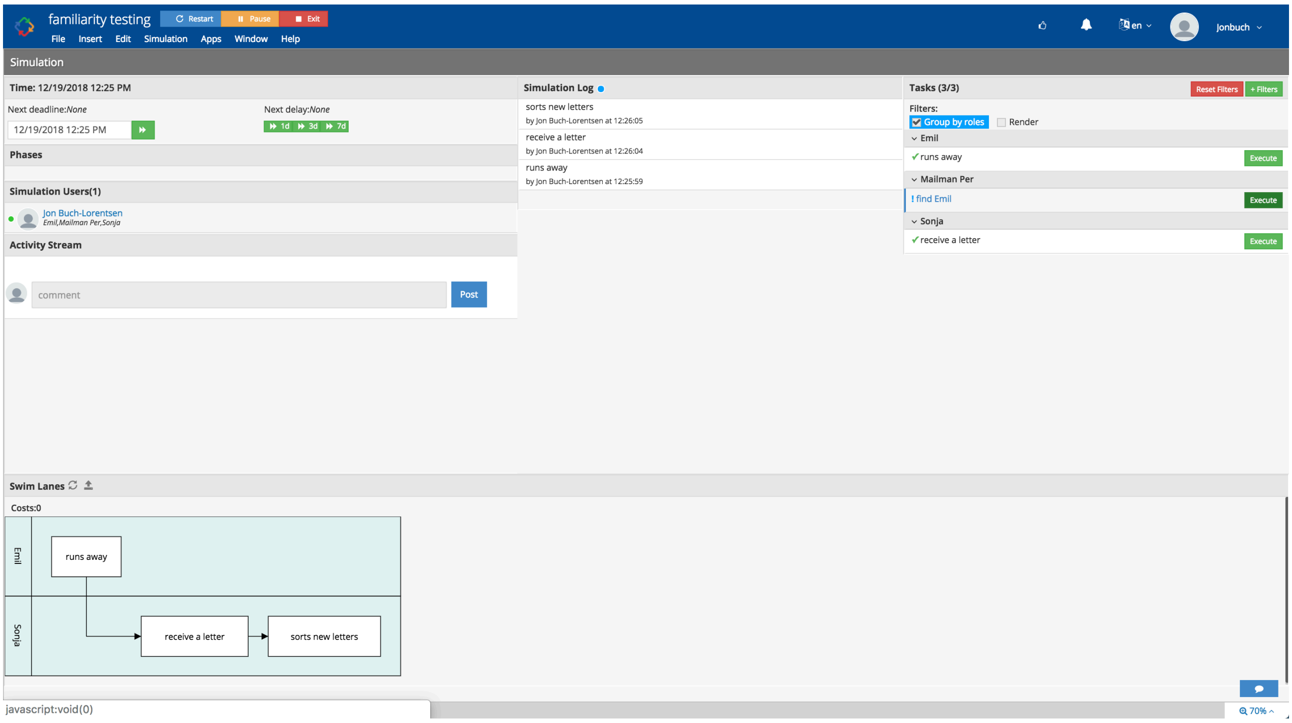
Copyright © DCR Solutions

Alliance | Wiki | About DCR Solutions

***Figure** 2.5: The view of the Monocle relation adder.*

2.2.2 The Simulator

The Simulator is a feature which allows the user to test their model. When running the simulator unrestricted activities of the model can be executed and the flow of the model can be tested. The simulator shows a log of activities by default however it is also possible to render the graph while simulating. The interface when running the simulator can be seen in figure 2.6.



Time: 12/19/2018 12:25 PM

Next deadline:*None*

12/19/2018 12:25 PM

Next delay:*None*

1d 3d 7d

Phases

Simulation Users(1)

Jon Buch-Lorentsen
Emil,Mailman Per,Sonja

Activity Stream

comment

Simulation Log

sorts new letters
by Jon Buch-Lorentsen at 12:26:05

receive a letter
by Jon Buch-Lorentsen at 12:26:04

runs away
by Jon Buch-Lorentsen at 12:25:59

Tasks (3/3)

Filters:
☑ Group by roles □ Render

Emil
✓ runs away

Mailman Per
! find Emil

Sonja
✓ receive a letter

Costs:0

Figure 2.6: *The view of the simulator without render turned on.*

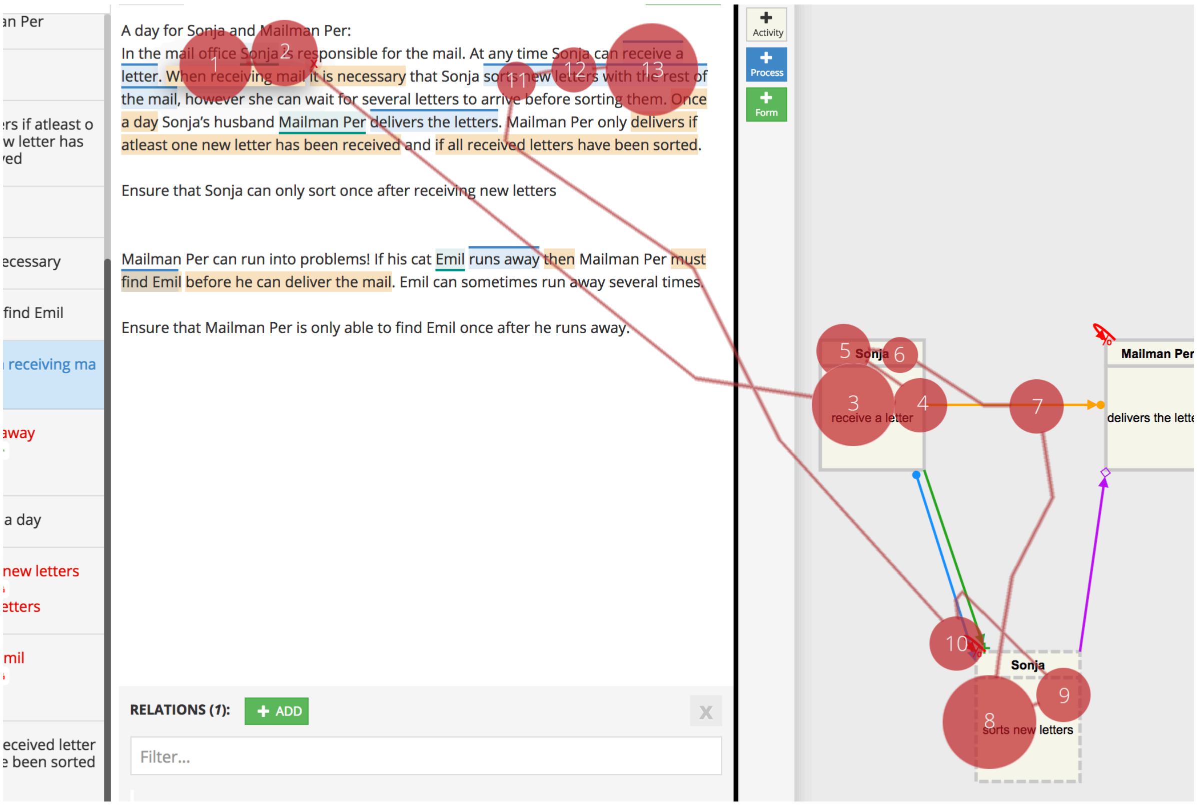
2.3 Eye tracking

Eye tracking is the process of measuring a persons eye movements. It has been showed that the innate characteristics of the eyes can reveal information about a persons physical and mental processes[15]. The method has been widely used in the field of human-computer interaction and for studying usability of websites and interfaces[15]. The most dominant method of eye tracking is done through pupil and corneal reflection tracking. This method is known as PCCR; pupil center corneal reflection[16]. Eye tracking commonly uses infrared- camera and light source since it is necessary to measure the pupil and corneal reflection with very high precision[16]. Using a regular light provides less contrast for identifying the pupil and corneal reflection. Additionally other light sources can interfere and cause issues for eye tracking if not based on infrared light. Eye tracking can be defined as a non-intrusive modality as neither the light or camera causes nuisance to the user.

In this project eye tracking will be used to evaluate and understand the viewing patterns and visual attention of the users. Eye tracking will be used to understand which parts of the interface the participants have been looking at and how they have been moving their gaze between different parts of the interface. To understand this Areas of Interests(AOIs) will be used. The following sections will focus on explaining AOIs and the relevant measurements related to AOIs.

2.3.1 Measurements

The book *Eye Tracking: A Comprehensive Guide to Methods and Measures* presents over 100 measures which can be used from eye tracking data[16]. The relevant measures to investigate will for any given study depend on the research question. This study will be focusing on extracting AOIs for creating process maps, this section will therefore only be focussing on the measures relevant to AOIs.



A day for Sonja and Mailman Per:
In the mail office Sonja is responsible for the mail. At any time Sonja can receive a letter. When receiving mail it is necessary that Sonja sorts new letters with the rest of the mail, however she can wait for several letters to arrive before sorting them. Once a day Sonja's husband Mailman Per delivers the letters. Mailman Per only delivers if atleast one new letter has been received and if all received letters have been sorted.

Ensure that Sonja can only sort once after receiving new letters

Mailman Per can run into problems! If his cat Emil runs away then Mailman Per must find Emil before he can deliver the mail. Emil can sometimes run away several times.

Ensure that Mailman Per is only able to find Emil once after he runs away.

Figure 2.7: *In the image we see both fixations and saccades with the circles being fixations and lines being saccades. A map like this is known as a scan path and the numbers indicate the order of fixations.*

2.3.1.1 Fixations

It is called a fixation when we focus our eyes on a narrow point in space. The focus area of our eye spans less than 2 degrees of our total visual field of over 200 degrees[16]. A fixation can last from between 100 ms to several seconds. It is generally considered that the attention of the person is put to the same object as the person is fixating on[17]. This is known as the eye-mind hypothesis. However this can only hold true as long as the information the person is looking at is relevant for the task at hand[17]. The length of fixations can be due to various internal reasons. In figure 2.7 we can see a set of fixations in a scene where each circle represents a fixation and the size represents the length of the fixation.

2.3.1.2 Saccades

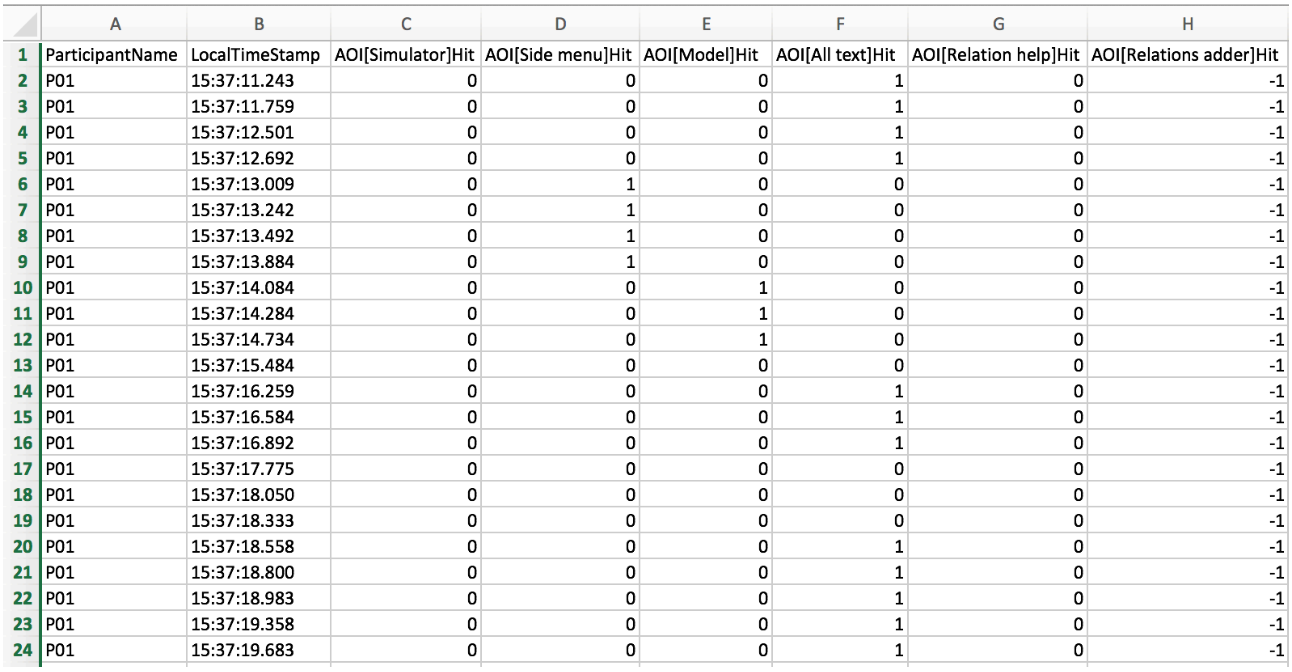
Saccades are quick movements of the eye from one fixation to a new. This is the fastest movement the body can produce and usually takes between 30-80 ms depending on the length of the saccade[16]. It is generally considered that the eye is blind during a saccade and unable to take in new information. It has been shown that saccades rarely take the shortest path between two points and that the path often have a curvature. Furthermore a saccade rarely hits right on the intended target and therefore a saccade is often followed by a smaller correcting movement called a glissade[16]. In figure 2.7 we can see the saccades between fixations represented by the lines.

2.3.2 Areas of Interest

AOIs are segregations in the stimulus defined by the researchers. AOIs are a commonly used since it is a useful tool which can improve the analysis of the eye tracking data[18]. The areas are defined by what is being investigated and by the user interface. Defining the AOIs is not elementary and how they are divided will have an influence on the analysis. It is therefore important to consider a meaningful division. AOIs should always be defined before the analysis as post-hoc analysis is subject to data dredging[16]. AOIs can either be assigned statically or dynamically depending on the interface and whether it is static or dynamic. A static interface will often be preferable as dynamic AOIs requires manual encoding. Implementing AOIs in your eye tracking study opens up to new ways of analysing the data. You can compare metrics between AOIs, such as the number of fixations or the average length of fixations in different AOIs. Central measures which will be used for creating process maps are dwells and transitions. The dwell is defined as the time from the users eye enter an AOI and until leaving it again. This means any number of fixations and saccades can occur within a dwell as long as the saccades land within the AOI. When a saccade lands outside the AOI the dwell ends. Note that a dwell is also referred to as a AOI hit. A transitions is the passing from one AOI to another. Again several fixations and saccades can occur before the users gaze enters an AOI. This means that if significant parts of an interface has not been assigned to AOIs some behaviours will be left unexplained in the analysis.

2.4 Process Mining

Process mining combines element from process modelling with elements from data science. Process mining allows for the automatic generation and identification of process models. It is called discovery (or Automated Business Process Discovery) when a new process model is generated by an algorithm from a dataset[5]. In order to discover a process model an event log is needed. An event log is a type of dataset extracted from a recorded process. An event log carries information about the executed activities and for each activity a timestamp for when it was executed. It also contains a case ID. The case ID relates to the instance of the process. These 3 attributes are considered as the basics of an event log to create a process model[5]. An example of an event log can be seen in figure 2.8. This event log is a simplified version of an AOI event log. In the log the AOIs are activities and the participant is the case ID. A 1 indicate which AOI is being looked at. In process mining this transfers to the activity being executed. 0 indicate AOIs not looked at and -1 indicate inactive AOIs. By discovering a process model from an AOI event log we can see the pattern of how the participant have been viewing the AOIs.



	A	B	C	D	E	F	G	H
1	ParticipantName	LocalTimeStamp	AOI[Simulator]Hit	AOI[Side menu]Hit	AOI[Model]Hit	AOI[All text]Hit	AOI[Relation help]Hit	AOI[Relations adder]Hit
2	P01	15:37:11.243	0	0	0	1	0	-1
3	P01	15:37:11.759	0	0	0	1	0	-1
4	P01	15:37:12.501	0	0	0	1	0	-1
5	P01	15:37:12.692	0	0	0	1	0	-1
6	P01	15:37:13.009	0	1	0	0	0	-1
7	P01	15:37:13.242	0	1	0	0	0	-1
8	P01	15:37:13.492	0	1	0	0	0	-1
9	P01	15:37:13.884	0	1	0	0	0	-1
10	P01	15:37:14.084	0	0	1	0	0	-1
11	P01	15:37:14.284	0	0	1	0	0	-1
12	P01	15:37:14.734	0	0	1	0	0	-1
13	P01	15:37:15.484	0	0	0	0	0	-1
14	P01	15:37:16.259	0	0	0	1	0	-1
15	P01	15:37:16.584	0	0	0	1	0	-1
16	P01	15:37:16.892	0	0	0	1	0	-1
17	P01	15:37:17.775	0	0	0	0	0	-1
18	P01	15:37:18.050	0	0	0	0	0	-1
19	P01	15:37:18.333	0	0	0	0	0	-1
20	P01	15:37:18.558	0	0	0	1	0	-1
21	P01	15:37:18.800	0	0	0	1	0	-1
22	P01	15:37:18.983	0	0	0	1	0	-1
23	P01	15:37:19.358	0	0	0	1	0	-1
24	P01	15:37:19.683	0	0	0	1	0	-1

Figure 2.8: *An example of AOI event log.*

An event log can contain various other attributes depending on what is being investigated from the process. Further attributes can be useful when analysing the model as it enables to filter the model based on these attributes. There exists various algorithms for discovering process models which also discovers in different process languages. The algorithm used is referred to as the miner. Currently there exists several tools made for discovering process models and analysing them. In this project the tool Disco will be used for discovering and analysing process models from the AOI event log.

2.4.1 Disco

Disco is a process mining toolkit which has been designed to easily import data. Disco allows for discovering models without expert knowledge of process mining and the underlying algorithms[19]. Disco creates graphical representations from the provided event log to models called process maps. The miner algorithm used in Disco to create models is based in the Fuzzy miner[20]. The Fuzzy miner is a mining algorithm that introduced the map metaphor and called its model maps[20]. The language used for process maps is rather simple. By default Disco displays the absolute frequencies of the activities and transitions. The transitions are indicated with weighted arrows. A higher weight indicates more transitions which is shown by a thicker arrow. An Example of a process map can be seen in figure 2.9.

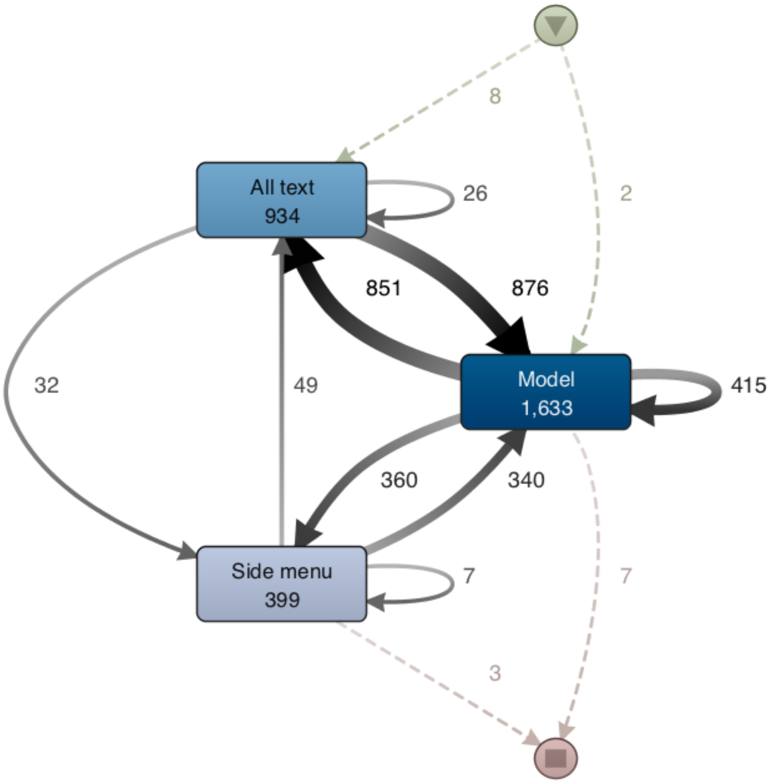
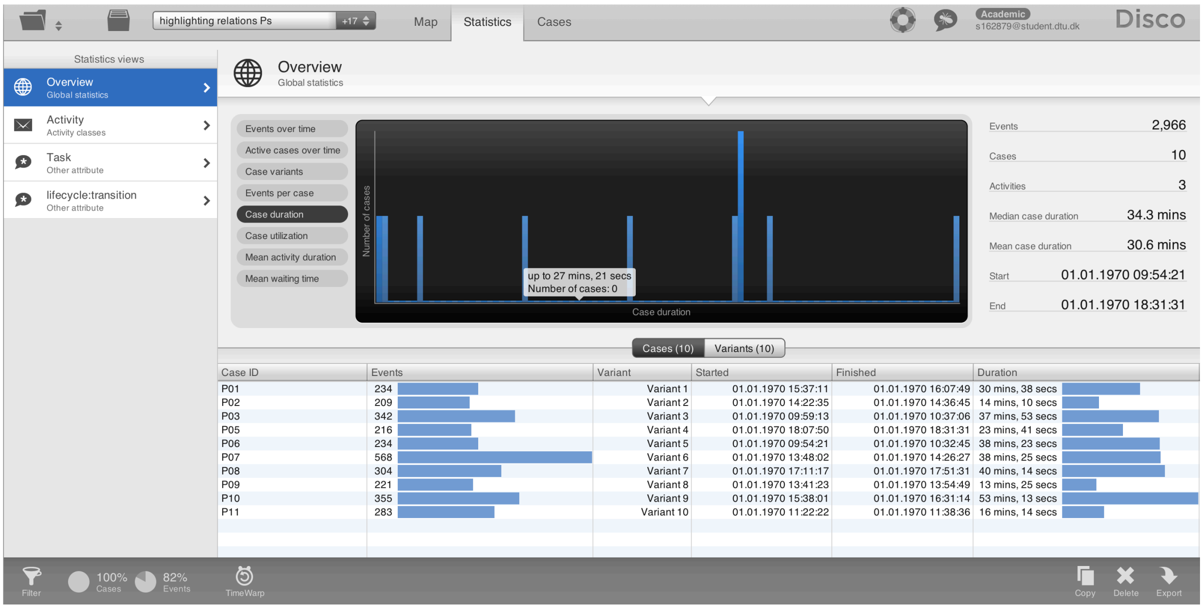


Figure 2.9: *Example of a process map discovered in Disco.*

Besides the graphical map representation Disco has several other features for analysing the data. The data can be filtered based on the different attributes such as case ID, activities, timeframes or any other attribute which has been defined in the event log. This enables for comparisons between various groups in the event log. Disco also includes statistics about the data. Here various information can be inferred about the different cases and activities to compare them to one another. The view of the statistics in Disco can be seen in figure 2.10. This is useful in an exploratory study as the data can be investigated from different angles.



Events	2,966
Cases	10
Activities	3
Median case duration	34.3 mins
Mean case duration	30.6 mins
Start	01.01.1970 09:54:21
End	01.01.1970 18:31:31

Case ID	Events	Variant	Started	Finished	Duration
P01	234	Variant 1	01.01.1970 15:37:11	01.01.1970 16:07:49	30 mins, 38 secs
P02	209	Variant 2	01.01.1970 14:22:35	01.01.1970 14:36:45	14 mins, 10 secs
P03	342	Variant 3	01.01.1970 09:59:13	01.01.1970 10:37:06	37 mins, 53 secs
P05	216	Variant 4	01.01.1970 18:07:50	01.01.1970 18:31:31	23 mins, 41 secs
P06	234	Variant 5	01.01.1970 09:54:21	01.01.1970 10:32:45	38 mins, 23 secs
P07	568	Variant 6	01.01.1970 13:48:02	01.01.1970 14:26:27	38 mins, 25 secs
P08	304	Variant 7	01.01.1970 17:11:17	01.01.1970 17:51:31	40 mins, 14 secs
P09	221	Variant 8	01.01.1970 13:41:23	01.01.1970 13:54:49	13 mins, 25 secs
P10	355	Variant 9	01.01.1970 15:38:01	01.01.1970 16:31:14	53 mins, 13 secs
P11	283	Variant 10	01.01.1970 11:22:22	01.01.1970 11:38:36	16 mins, 14 secs

Figure 2.10: *The statistics view of Disco.*

2.5 Qualitative Research

Qualitative research is a naturalistic type of scientific research that focusses on subjective data rather than numerical data[21]. It often takes an exploratory approach to understanding why a phenomenon occurs[21]. In this project it will be used to gain deeper insight to understanding why users engage with the interface the way they do. To do so verbal data will be collected and analysed.

2.5.1 Verbal Data

Complementing eye tracking data with verbal data is widely used for usability studies[16]. Having subjective data adds validity as it enables triangulation between data to ensure what is observed in the eye tracking data is also confirmed in the verbal data[16]. Two frequently used methods of collecting verbal data within human-computer interaction and usability studies are think-aloud and interviews.

Think-aloud is a method in which the users are asked to explain their thoughts as they are performing a given task. The participants are asked to remove their filter from though to speech and just articulate whatever pops into their mind[22]. This method captures the immediate response and reaction from the participants. However this method is very intrusive and will highly influence the test as the participants attention will be divided between the task and explaining their thoughts. It also requires certain effort from the participant as it might not come natural to speak your every thought. Furthermore in relation to eye tracking it has been suspected that the act of speaking can alter eye movement and also make recordings less accurate[16]. Because of these issues a retrospective think-aloud is often proposed for eye tracking. In this method the eye tracking recording is played back to the user after the test. While watching the participant is asked to recall their thought during the moments in the videos and explain them. It is considered that the participants memory of the moments will be enhanced by having the video as a reminder. It should be noted that this method will greatly increase the length of the test [16]. Another common method used in evaluation studies are interviews. Interviews can range from structured to unstructured. In a structured interview the interviewer only asks a list of predefined questions whereas in an unstructured interview there are no pre-made questions and the interviewer guides the interview as it progresses. In between is the semi-structured interview which is based on a set of predefined questions but as the conversation evolves the interviewer can chose to ask questions to probe on arising subjects [23]. Interviews with less structure often have greater depth and breadth as they can explore unforeseen topics and are very useful when exploring an unfamiliar problem domain[23]. However less structure also requires more skill from the interviewer who must be able to ask the right questions. A common drawback of the interview and retrospective think aloud is the problem of recall. Only what the participants remember from the test will be reported [23].

2.5.2 Qualitative Coding

Qualitative coding (coding in short) is a method which can be used to analyse qualitative data. It can be used to understand the data and can potentially be used to construct a theory about the data which is know as grounded theory[24]. The method has a very open framework which means that the method provides a set of tools which the researcher can use as they want[25]. Coding can therefore be done in different ways as the process is highly influenced by factors such as the data and the research question[25]. It is a tool to help researchers reach their goal rather than it is an strictly predefined path that she must follow. Coding is an inductive method where a bottom-up approach is usually taken for developing an understanding of the data. It can therefore also be seen as an exploratory method which let findings arise from the data The data used for coding can come in numerous forms and examples are: text transcribed from an interview, video, photos, websites, e-mail correspondence, journals, documents, field notes[26]. Coding is done though a number of steps. And can generally be divided into four steps; coding, sorting, theme-ing, theorising[25].

In the first step pieces of data are assigned codes which can be words or sentences that captures the essence of the given data. This step is known as the first coding cycle[26]. Throughout a dataset the same code will often be applied many times to different data. This is one of the goals of coding as it condenses the data and reveals the frequency of different topics[25]. Coding is therefore the process of linking the data and getting to the underlying concepts. There are many methods of coding and Saldana presents 29 methods in the book *The Coding Manual for Qualitative Researchers* for first cycle coding[26]. It should be noted that you are not limited to choosing one coding method and that you can encode your data in several different ways to view your data through different lenses. A commonly used method for first coding cycle is initial coding. This method looks at the data openly and breaks it down into discrete parts. The goal is to remain open to all possible directions which the data could lead to[26]. After the first coding cycle comes sorting. In this step you group codes together and see how all the codes you have generated relate to each other. The third step is the second coding cycle. Similar to the first cycle, there also exists different methods for coding the second cycle. Saldana presents 6 methods [26]. The methods are however similar since they aim to develop a sense of categorical, thematic and conceptual understanding from the first cycle codes[26]. Axial coding is a commonly implemented which divides the codes into categories by the codes properties and dimensions[26]. In the final step a theory of the data can be developed. This step is what defines grounded theory.

3 Research Method

In this section the approach taken for conducting experiment will be explained. It will start with discussing the overall method of an exploratory study. Then the more concrete aspects of the study will be explained. The task for the participants will be presented along with the decisions made for the design Then the methods for collecting data from the experiment will be presented along with the approaches.

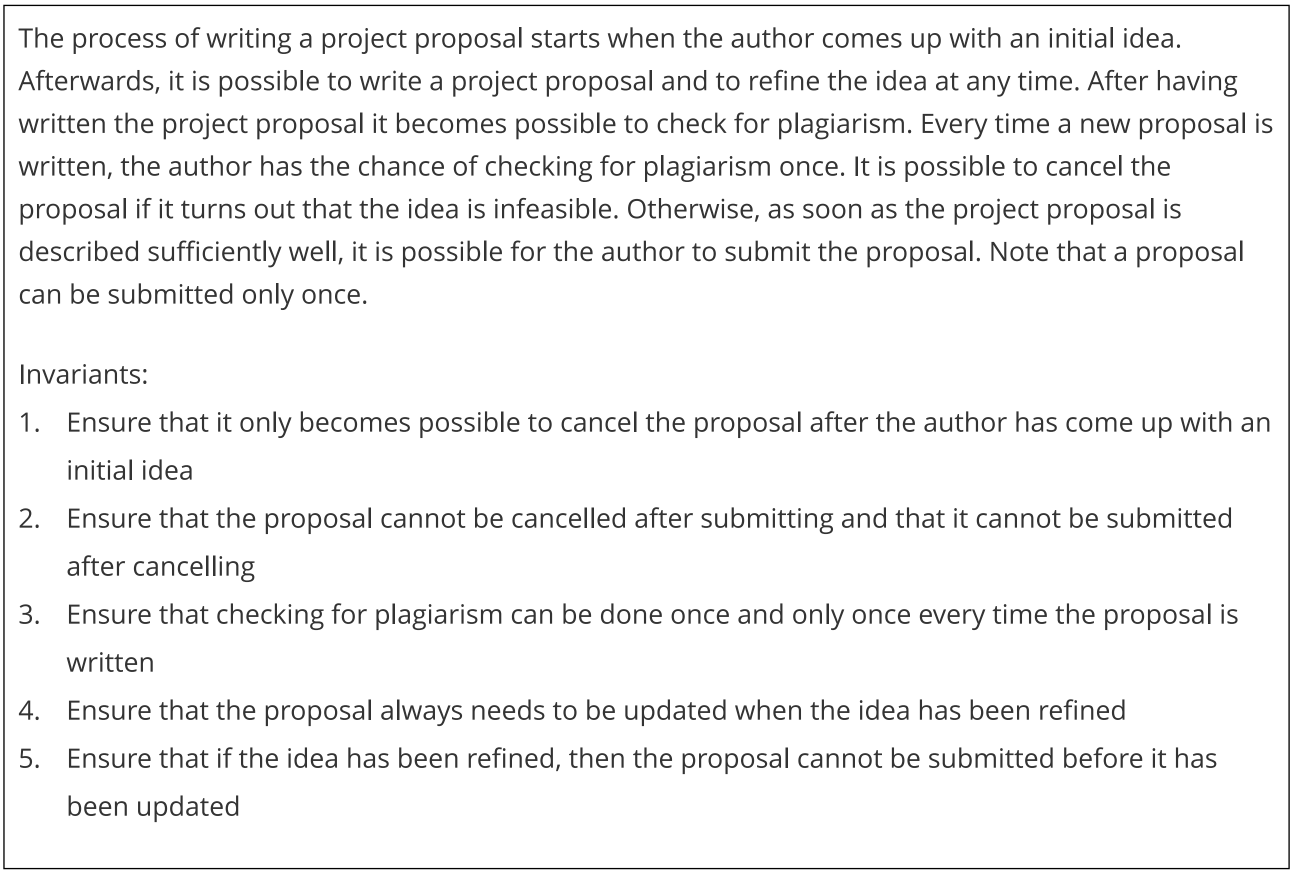
3.1 Exploratory Research

To investigate Monocle an exploratory approach was chosen. In an exploratory study a subject is examined to hopefully gain new insights. This means there are no pre made hypotheses which are to be confirmed or rejected. Instead a broad research question is defined which allows interesting aspects or issue to arise from the data. This indicate a bottom-up approach where the research should try to avoid having pre conceived ideas of what to look for and instead be open minded. Findings from an exploratory study are often not final answers but rather indications of interesting aspects in the data and about the subject. If the study finds interesting aspects then these findings can be taken further in following studies to make more evidential studies. The goal of exploratory research is therefore to find hypotheses rather than testing them[27]. This study will therefore not be searching for statistical significances in the data. Instead it will try and look at the data from many different angles to see if there is anything interesting arising. These findings will then be reported and discussed. It is then the possibility that future research might find ideas or inspiration from this work to use for new studies on the subject. Unlike a descriptive study where precision in the research- design and questions is important to eliminate biases and noise an exploratory study should be more open and flexible to be able to consider many different aspects of the phenomenon[27].

3.2 Task Design

In order to learn about Monocle it was necessary to design a task in which the participants would be using Monocle. Since Monocle is made to help both for model creation and model maintenance it was also necessary to figure out what should be the focus, as this would highly influence the design of the task. This led to a discussion with the developers of Monocle about what would be most valuable to investigate. It was decided to keep model creation as the focus since it is the most important aspect of the tool and what the tool is originally intended for.

For creating a model it was necessary to have a textual description which the participant could model and highlight from. Since DCR graphs is designed for declarative model creation it was decided to design a circumstantial description for the task. In previous work regarding declarative process modelling a circumstantial model, based on constraints, had been proposed[28]. This process model description was used and adapted to fit in the context of DCR Graphs. The process description defined how to submit a project proposal. In order to have a suitable text, it was necessary to change the formulation of a few sentences in the description. This ensured that activities and relations were explicitly mentioned and could be directly linked to words in the text. Additionally one activity were changed from *language check* to *check for plagiarism*. This was done to make a model that could support all the 5 different types of relations, in order to have a comprehensive evaluation of all the DCR constraints. Furthermore it was decided to add invariants which the participant would have to check off after building the model. The invariants were additional constraints which could be in the model but had not been explicitly mentioned in the textual description. These were added to include constraints which should not be added with Monocle. They could also be used to see whether the participants would add invariant from their own knowledge of the process domain.



The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. Every time a new proposal is written, the author has the chance of checking for plagiarism once. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

Invariants:

1. Ensure that it only becomes possible to cancel the proposal after the author has come up with an initial idea
2. Ensure that the proposal cannot be cancelled after submitting and that it cannot be submitted after cancelling
3. Ensure that checking for plagiarism can be done once and only once every time the proposal is written
4. Ensure that the proposal always needs to be updated when the idea has been refined
5. Ensure that if the idea has been refined, then the proposal cannot be submitted before it has been updated

Table 3.1: *The textual process description created for the experiments.*

Table 3.1 shows the textual description and the 5 invariants which the participants had to implement during the test. It was also decided that the test would let the participant freely chose how to use the modeller rather than oblige the participants to use it. The drawback of this was that participants might choose to ignore Monocle completely. However in light of RQ1 the investigation wanted to explore how they engage with Monocle and if the approach would be forced upon them the question could not really explore how the participants would naturally engage with Monocle.

3.3 Data Collection Methods and Design

The overall methodology for capturing and analysing data was defined at the beginning of the project. However it was still need to define how specifically these methods would be used and implemented and if there were any additional data which could be gathered to assist the analysis. This section will go over each of the methods chosen to investigate the research questions. It also explains how it was implemented to the experiment.

3.3.1 Models

Reviewing a process model in DCR graphs can reveal information about the model builders approach to creating it. It can be seen how they created the model; which activities, relations and roles did they create. It can also be seen which parts were connected to Monocle and which parts were created directly in the modeller. Since RQ1 is about investigating how they engaged with Monocle, this information can complement the eye tracking data for investigating this question. The models can also potentially be used to extract other information how they made the model and how consistent it is with the textual description.For each participant the model will be saved with their participant number so it can be used to examine their model during the data analysis.

3.3.2 Eye tracking

As mentioned process maps based on eye tracking AOIs will be used to capture the viewing patterns of the participants. The interface will be divided into AOIs and the data will be used to identify AOI- dwells and transitions. This information will then be used to discover process maps using Disco. These maps will then serve as the main driving force force for investigating RQ1.

3.3.3 Interviews

For gathering verbal data and investigating RQ2 interviews were chosen over retrospective think aloud as it was expected that the test would take a considerable amount of time (more than 20 minutes) and that doubling this time would be infeasible. To keep some reference for the participants the model and highlights they had created would be present during the interview. Following the exploratory nature of the experiment it was also chosen to perform semi-structured interviews which allowed for unforeseen topics to emerge.

6 overall questions were created for the interview These questions served as the main structure for the interview and would be asked to all participants. Each question then had a list of subquestions which would serve to either probe in on a given answer or to encourage the participant to continue talking. The 6 questions and subquestions can be seen in table 3.2.

1. Did you use Monocle to drive the modelling?
   - Was it the main tool used for creating the model?
2. Could you create all activities and relations using Monocle or did you have to use the modeller for certain aspects?
   - If no;
     - why was using Monocle not enough? Briefly go over your approach and explain what happened and what you did
   - If yes;
     - So you felt you were able to create an adequate model with only Monocle?
     - How many of you did and felt the same? (For case workers)
   - Follow up for both
     - Were there any DCR relations you could not easily identify from the text?
     - Were you in a situation where you had to extend the model with activities or relations that were not explicitly mentioned in the text?
     - For what purpose have you used the modeller?
     - Have you used and for what purpose have you used the simulator?
     - In what order have you used the tools?
3. How did Monocle help you in the modelling task?
   - Can you mention the features that were useful in monocle
   - How does the Monocle help you to identify important information from the text.
4. What challenges did you face using Monocle?
   - Were there anything you were not able to do, using the tool?
   - Could you think of features that could be improved?
5. Was there anything you especially liked or disliked about using the Monocle tool?
   - Something that worked well or not so well
6. Any Further comments?
   - Any points you wanted to make but did not feel you could express from my other questions?

Table 3.2: *The interview questions used for gathering verbal data.*

In the first question the participants were asked if Monocle was the main driving force for building the model. This was asked to get their initial view on what they used to create the model. In the second question they would be asked about how they created relations and activities and whether they used Monocle or the modeller. This first part of the question was mostly related to RQ1 but was necessary to answer in order to understand the individual participant to better understand their coming answers and to ask the right follow up questions. By explaining how they modelled it would also lead up to naturally explaining why they did so (the second part of the question). This would force the participants to reflect on their reasoning behind their modelling strategy. Having the participants explain their approach also gave some data which could be triangulated with the observations from the eye tracking data. Question 3, 4 and 5 would directly ask about the the positive and negative aspects about using Monocle. In between each question the sub questions that suited their answers would be asked and also new questions would be asked if the participants would mention something interesting that had not been expected.

3.3.4 Questionnaire

When performing an eye tracking experiment it is necessary to know whether the participant has any eye conditions as these might cause issues which can contaminate the data collected[16]. For this reason a questionnaire was added to the experiment. The questionnaire would both serve to expound any eye related conditions but also to gather basic demographic information about the participants. Thirdly a set of questions were added about the participants experience with DCR graphs. These additional information also had potential to be used in the analysis as it could potentially distinguish participants into different groups of behaviour or experience. The full questionnaire can be found in Appendix 11.4.

3.3.5 Galvanic Skin Response

Galvanic Skin Response (GSR) is the measurement of the electrical conductance of skin using electrodes. The electrodes are usually put on two fingers. Variation caused by sweat glands are then measured. These variations have a correlation with the arousal of the person being measured[16]. It was decided to measure GSR of the participant since supplementing this with eye tracking can give insight to the cognitive load and emotional response of the users[16]. This was therefore tracked as an extra data source that could be investigated in future work. However it was not used for the analysis of this work and to keep conciseness it has not been mentioned in the background chapter.

4 Experiment One

Experiment one can be considered the main experiment. The experiment was conducted individually in a controlled environment where the users would first be introduced to DCR, then created a model and finally participate in an interview. This chapter will go over the details of the experiment.

4.1 Participants

The participants which participated in the experiment were recruited through convenience sampling. As the test would require the participants to create a process model the recruitment was focussed on obtaining participants who had some previous exposure to process modelling or at least some experience with programming. A total of 10 participants were recruited for the experiment with the majority being computer science students from DTU.

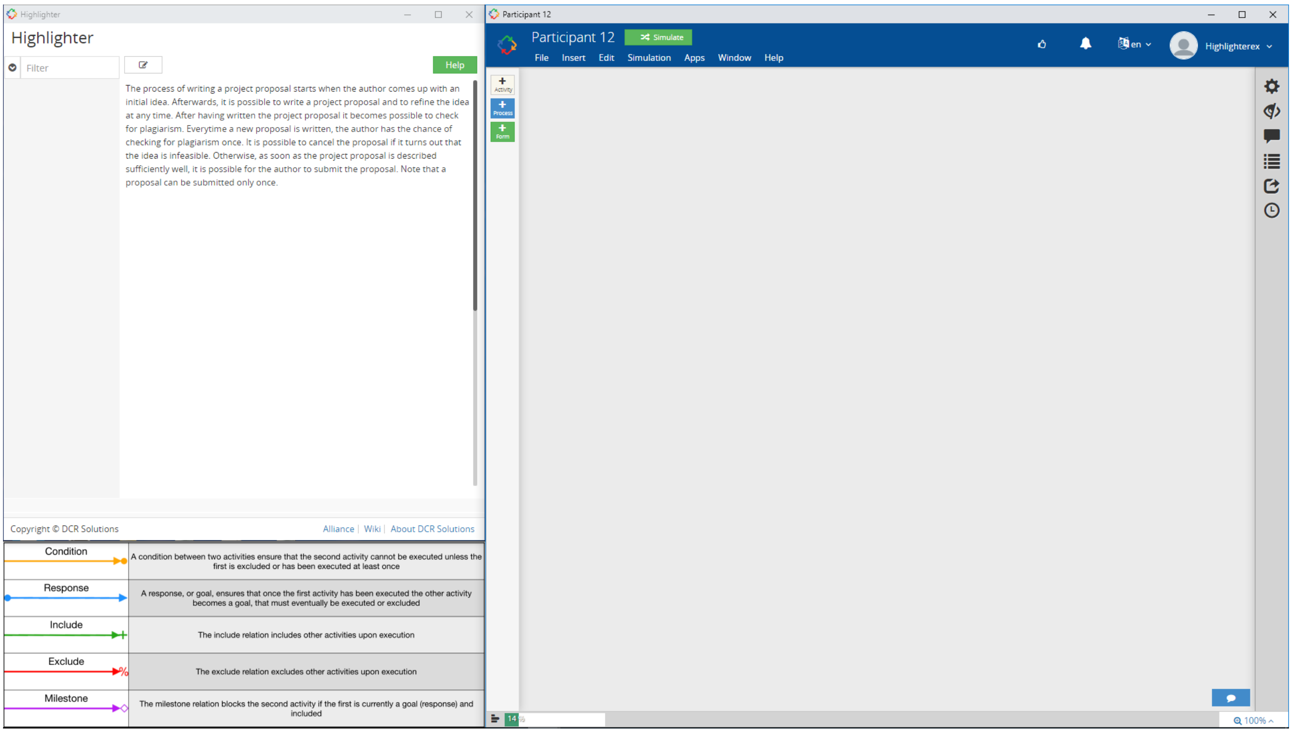
4.2 Task Description

The description of the main task can be seen in table 3.1 in research method. However as the participants had only slight- or no previous experience with using DCR graphs, it was necessary to familiarise them with the tool before throwing them into the main task. To do so a familiarisation session, in which the participants would learn the program, was designed. The familiarisation started with a slideshow in which the basics of DCR was presented. Afterwards the 5 relations were explained to the participant using the modeller. This was done by creating the relation, then explaining it and finally simulating it to show its behaviour. After this the participants were taken through the final part of the familiarisation in which they would build a model. This task was designed to resemble the actual task and contained a process description text which the participants would model. The process was based on children's show called Postman Pat[2] in which a mailman delivers letters (See appendix 11.3 for the process). This was chosen as the process as it would be easy to comprehend the process. Thereby the participants could focus their attention on learning to build the model rather than understanding the process and its semantics. The task was divided into two parts where each part contained a short process description and one invariant. In the first part the facilitator would take the lead and create the first two roles, activities and relations while discussing it with the participant. After this the facilitator would hand the mouse to the participant and let them continue while still assisting and discussing the task. After completing the first part the facilitator would ask the participant to continue on to the second part. Throughout this part the facilitator would act more passive and only intervene if the participants asked for help or were struggling. The second part was

[2] https://en.wikipedia.org/wiki/Postman_Pat

deliberately designed to be similar to the first part. Throughout the session participants were taught and encouraged to model with Monocle and shown to add invariants using the modeller.

During the main task the participant were given a help window which would display the relations and a description of them. This was to help them in case they forgot how a relation behaved. The placement of the help window along with the rest of the interface setup can be seen in figure 4.1.



The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. Everytime a new proposal is written, the author has the chance of checking for plagiarism once. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

Condition	A condition between two activities ensure that the second activity cannot be executed unless the first is excluded or has been executed at least once
Response	A response, or goal, ensures that once the first activity has been executed the other activity becomes a goal, that must eventually be executed or excluded
Include	The include relation includes other activities upon execution
Exclude	The exclude relation excludes other activities upon execution
Milestone	The milestone relation blocks the second activity if the first is currently a goal (response) and included

Figure 4.1: *Interface setup for participants.*

4.3 Procedure

The experiment took place in a designated eye tracking lab at The Technical University of Denmark between October 29 and November 6, 2018. The experiment had three steps: Familiarisation session, the test and an interview session. In the first step the participants would be introduced to DCR graphs as explained above. Afterwards the participant would be asked to read and sign a consent form allowing for usage of the data collected during the experiment. The consent form can be found in appendix 11.4. Then the participants had to fill out a questionnaire.

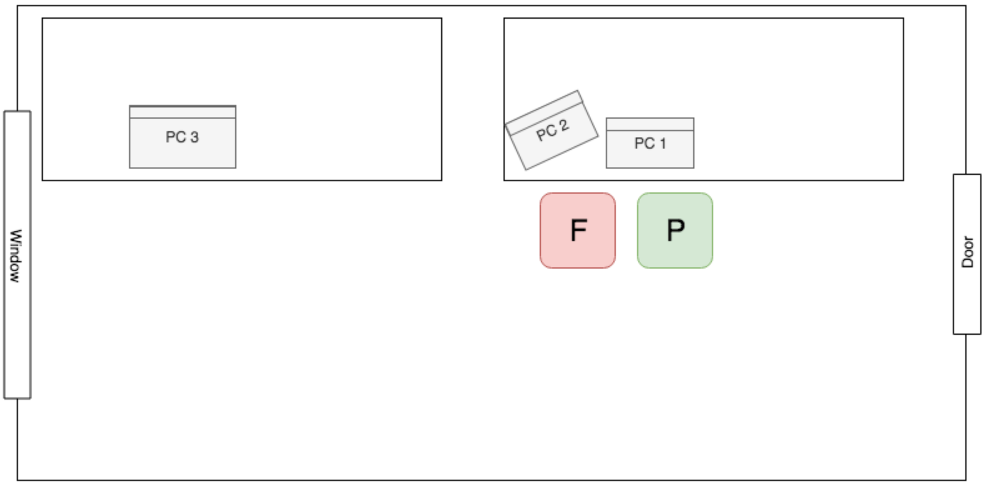


Figure 4.2: *Setup during the first step. F indicate position of the facilitator and P indicate the position of the participant. The familiarisation process model was created on PC 1 while the slideshow which contained info about the 5 relations were on PC 2.*

After finishing the questionnaire the participant were asked to move over to the eye tracking computer for the second step. Here they were first equipped with the GSR tracker and then the eye tracker was calibrated. After calibrating the eye tracker the test would start. Once the participants felt they had finished building the model they would notify the facilitator who would then provide the participant with the invariants. When the participants felt they had checked of all invariants they would again notify the facilitator. The facilitator would then turn of the eye tracker and GSR device.

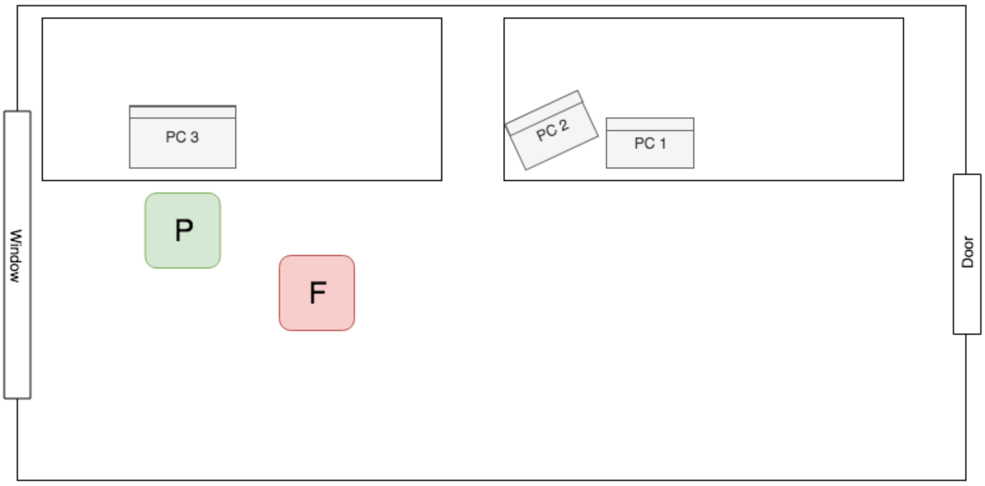


Figure 4.3: *Setup during the second step. F indicate position of the facilitator and P indicate the position of the participant. The facilitator placed himself out of sight of the participant but where he was able to overview the participant and PC 3 in case of any issues.*

In the third step the facilitator would sit next to the participant while still having their model visible on PC 3 and conduct the interview. The interview was recorded on PC 1 After the interview the experiment would be over.

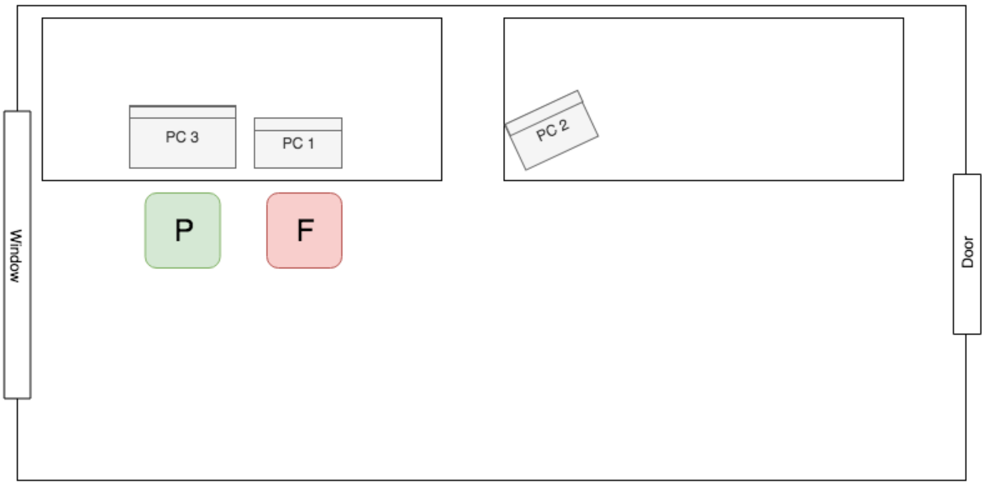


Figure 4.4: *Setup during the third step. F indicate position of the facilitator and P indicate the position of the participant. The facilitator sat next to the participant to ask questions while using PC 1 to audio record. The model was still visible on PC 3 for the participant or facilitator to comment on.*

4.4 Instruments

For tracking the participants gaze the Tobii Pro x3-120 eye tracker[29] was used. The eye tracker uses PCCR with an infrared camera and illumination to capture the eye. The eye tracker comes with an external processing unit in which the eye tracking processing of the raw data can be calculated.

The eye tracker was used in combination with Tobii Studio (now named Tobii Pro Studio). This software allows for setting up and managing experiments. It also allows for various analysis of the data such as creating and analysing AOIs, segmenting the video, attention maps and other various statistics[30].

To ensure that the interface had same dimensions for all participants the program Sizer[3] was used to adjust the size of the Modeller, Monocle and the help window.

Galvanic skin response was tracked using the Shimmer3 GSR+ Unit using their Consensys software[31].

[3] http://www.brianapps.net/sizer4/

5 Experiment Two

A second experiment was conducted in connection with a workshop concerning Monocle tool. The workshop was an introduction to Monocle for caseworkers who had previous experience with DCR graphs. 7 caseworkers participated in the experiment. The experiment was designed to be similar to experiment one, but some changes were necessary due to the setting and timeframe of the workshop. Since all participants modelled at the same time eye tracking was not possible and instead screen recordings were captured.

5.1 Participants

The participants of experiment two consisted of 7 caseworkers working at Syddjurs municipality. The caseworkers had previous experience with DCR graphs as they had attended previous DCR workshops. The majority of the case works were still novice users however their level varied.

5.2 Task Description

The same textual process description was used for experiment two as had been used for experiment one however as the workshop was in danish the text had been translated (the danish description can be seen in appendix 11.2). Furthermore the number of invariants were reduced to two as it was not assumed that they could finish all five in the given timeframe. The invariants kept were invariant 1 and 4.

5.3 Procedure

The experiment took place October 30, 2018 at a conference room in Syddjurs Municipality buildings. The experiment consisted of the process building exercise followed by an interview. The participants were given 25 minutes to model the process. The participants were asked to download, install and run a screen capturing software before the exercise. After the exercise the videos were collected with an external hard-drive. Additionally the participants were given a pre-made DCR account which ensured that the models could be analysed. After the model building exercise a semi-structured group interview took place. The interview was based on the same questions which had been used for the experiment one. Few changes were made to fit a group e.g. in the first question the participants were asked to raise their hand if they agreed to have used Monocle to drive the modelling.

Figure 5.1: *Photo taken during the group interview at Syddjurs.*

5.4 Instruments

The participants had all brought their work computers which they used for the modelling task.To record the participants screen and interactions the program Open Broadcaster Software[4] was used. To record the interview a omnidirectional conference microphone was used to ensure capturing all the participant.

[4] https://obsproject.com/

6 Data Analysis Procedure

This chapter will explain how the data collected from experiment one and two was analysed. Firstly the preparation of the raw data will be presented. Then the methods used for analysing will be mentioned and the analysis procedure will be explained. This will be done for each of the 3 main data sources used for investigating the research questions; Models, Eye tracking and interviews. When differentiating between participants from experiment one and two the participants will be referred to as student participants and caseworker participants. It should be noted that the eye tracking analysis only contains the student participants whereas the models and interviews contains both groups of participants.

6.1 Data Preparation

After realising the experiments the raw data had to be prepared and cleaned for further analysis. The models did not require any work before analysing and will therefore not be mentioned. The eye tracking data and interviews required substantial work for being prepared. This section will present what had to be done to each data source.

6.1.1 Eye Tracking to Process maps

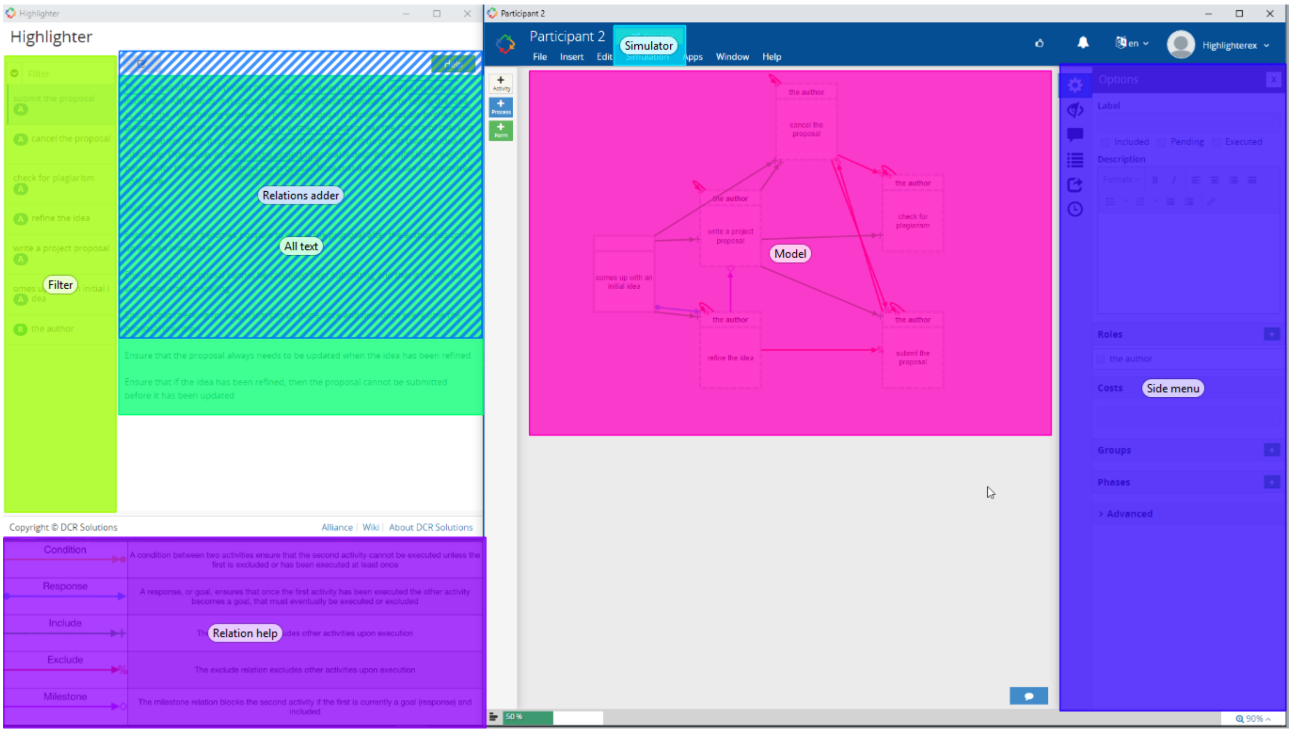
The first step for preparing the eye tracking videos was to segment them into parts. In order to compare the modelling task and the invariant task the videos had to be segmented into two parts. A segmentation feature is implemented in Tobii Studio and could be done with little effort.

During the experiments there had been some issues regarding the simulator. It became apparent that the render function of the simulator was not fully compatible with Monocle. The render therefore only worked sometimes[5]. This resulted in uncontrolled behaviour during simulation as some people would use render, some would not use it because of the issue and some would retry until it worked. Since this added inconsistency and would have been extremely cumbersome to map with AOIs it was chosen to segment out the parts where the participants used the simulator. It should also be noted that the simulator AOI therefore only covers the button that would start the simulation tool. By looking through the eye tracking videos it was confirmed that most often they would only look at the button when also running the simulator. For this reason the simulator button AOI will be taken as an indication of their simulator usage in the analysis.

Next the eye tracking data needed to be divide into AOIs. Since the interface was dynamic it was also necessary to make the AOIs dynamic. In total 6 AOIs were added. Initially more overlapping

[5] Video of simulator issue: https://youtu.be/ZZaSpMmdGrU

AOIs were added to have a more detailed division, however the overlap proved to be an obstacle for using it in Disco. The 6 AOIs created were; The text named *All text*, the simulator called *Simulator*, the Model called *Model*, the menu to the right of the model called *Side menu*, the image explaining the 5 relations called *Relation help* and finally the screen that would appear when adding a relation from Monocle called *Relations adder*. As mentioned in section 2.2.1 the text would disappear when the relation adder appears. This means that transitions between these two AOIs are not possible (except for second when the relation adder appears or disappears), this should be noted when analysing the process models.



Ensure that the proposal always needs to be updated when the idea has been refined

Ensure that if the idea has been refined, then the proposal cannot be submitted before it has been updated

A condition between two activities ensure that the second activity cannot be executed unless the first is excluded or has been executed at least once

A response, or goal, ensures that once the first activity has been executed the other activity becomes a goal, that must eventually be executed or excluded

The exclude relation excludes other activities upon execution

The milestone relation blocks the second activity if the first is currently a goal (response) and included

Figure 6.1: *AOI divisions of the interface.*

Once all the AOIs had been mapped the data could be exported to an AOI event log as an .tsv file. This file was then cleaned and formatted to .xes using a previously made script created for same cause[6]. This made it compatible to load into Disco.

6.1.2 Transcribing Interviews

A total of 1 hour and 12 minutes of interview had been recorded from experiment one and experiment two. Since the interviews were audio files it was decided to transcribe them to make it easier to analyse. This was done manually by replaying the recordings and writing down while listening. The full transcriptions of all interviews can be found in appendix 11.4.

[6] https://github.com/DTU-SPE/tsv2xes

6.2 Analysis Procedure

This section explains the procedure of how each of the 3 data sources were analysed.

6.2.1 Models

To identify what the participants had used Monocle for the models were opened and Monocle was investigated to check for highlights. After checking all models the participants were divided into three groups:

- Participants using Monocle only for activities and roles
- Participants using Monocle for activities, roles and some relations
- Participants using Monocle for activities, roles and all relations

After creating the three groups, the videos of them were review to ensure that they had in fact used Monocle to directly create the markup or whether they had created it from the modeller and then afterwards connected it to Monocle. It was observed that non of the participant had used Monocle to retrospectively add markup but that it had only been used actively.

6.2.2 Process Maps

It is crucial to understand what the model represents in order to analyse the discovered process maps. First thing to understand is how the eye tracking measurements translates over to the elements of the process map. Each participant represents a case. The activities in the maps are representing the different AOIs. Each activity has a frequency, this number is representing the total number of dwells in the given AOI. The edges between the activities represent transition between AOIs and the frequency of the edges is the total number of transitions between the two given AOIs.

After the process models has been discovered the different attribute filters could be used to explore the data from different angles. First the full map of all participants was examined. This was done by simply comparing the frequency of different AOIs and observing the transitions between the AOIs to see the importance and centrality of the different AOIs. After this different attributes were filtered in order to compare different maps. The first comparison was how the modelling task and the invariant task compared to see if there were in difference in their approach to the two tasks. From looking at the models it had been observed that the participants could be divided depending on their usage of Monocle. To see if there was a difference whether using Monocle or modeller for relations two groups were compared. The first group contained the participants only using the modeller for relations and the other group contained the participants using only Monocle for relations.

The next aspect which was decided to look into was the confidence of the participants. This was to see if participants who were more secure in what they were doing had a different approach than participants who were less secure. To define confidence their usage of assistance from the simulator and the relation help was used. Each participant were reviewed in the Disco statistics for their usage of either the simulator or relation help. They were then divided between participant spending more or less than 10% gazing towards the relation help or the simulator.

6.2.3 Interviews

To analyse the data initial coding was used for the first coding cycle. For the second coding cycle axial coding was used. A bottom-up approach was taken for the coding. Therefore no preconceived codes or groups were defined for the data. Instead the code was read through and codes defined directly from the data. To really understand the data the first cycle was done through 2 iterations. In the first iteration very detailed codes were added to all interesting parts of the interviews. In the second cycle the interviews were reviewed again but this time it was done while having the list of codes from the first iteration at hand. The codes were hereby reviewed and codes containing same connotation were merged. It should be noted that the count of the codes was the number of participants who mentioned it one or more times rather than counting each time it was mentioned. This was chosen to be more representative as some participants would mention the same code several times while probed in the interview. This could have created a skew about the relevance of a code. After the coding a total of 51 codes had been created. Next step was to identify how they could be sorted. This was also done over several iterations where different combinations were attempted. The sorting ended up having seven themes:

1. Usage of the tools
2. Strong in Monocle
3. Strong in modeller
4. Tool issues
5. Usage of the simulator
6. Implementing invariants
7. Suggestions for improvements

After having the 7 themes each group were carefully reviewed again to see what picture the combination of the codes gave.

7 Results

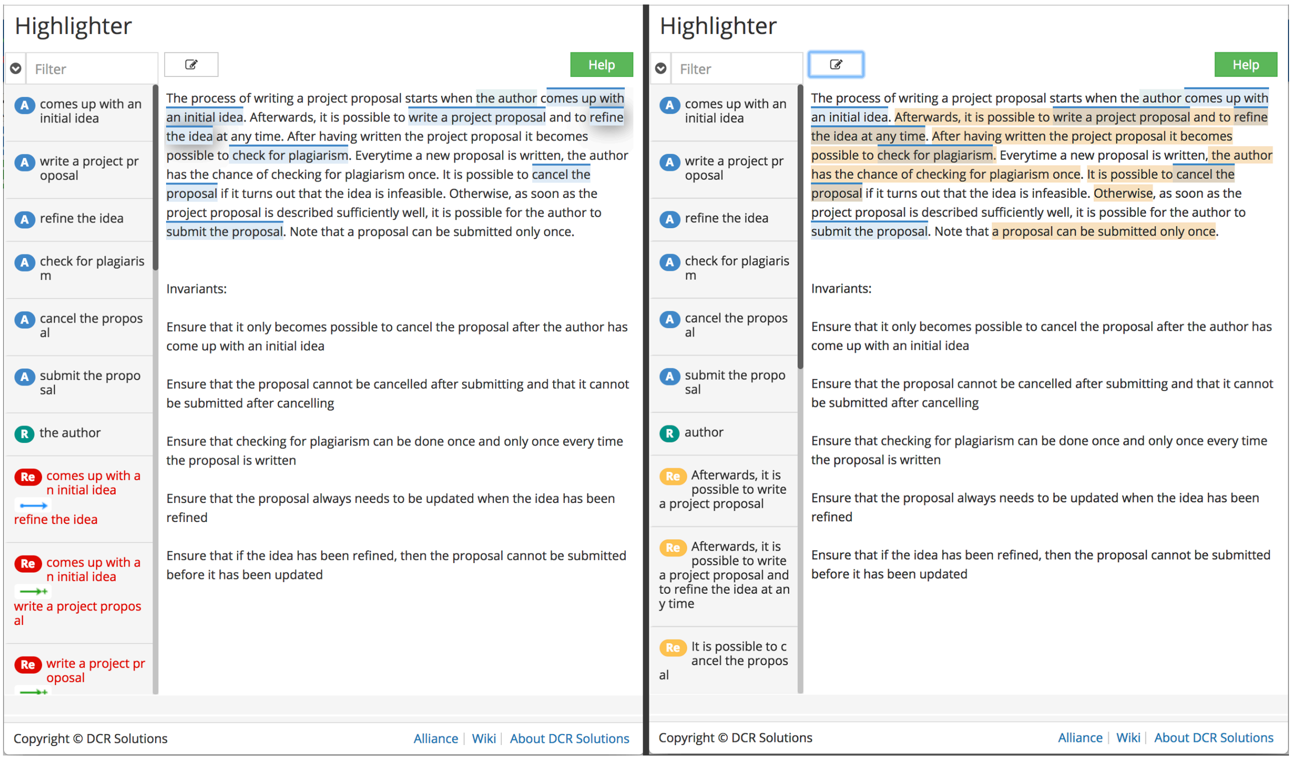
This chapter will present the results extracted from the data together with reflections on what the results are showing and what might be the reasons for the observed behaviours.

7.1 RQ1

To investigate RQ1 both the models and the process maps were used. Therefore this section is divided between the two, showing the results of each data source.

7.1.1 Models

All the participants both students and case workers had used Monocle to create the activities and the role. For relations 5 of the 10 student participant did not create any relations using Monocle, 3 student participants created a couple of the relations with Monocle but then moved to the modeller and 2 student participants used Monocle extensively for nearly all relations.


Highlighter
Filter
Help
Copyright © DCR Solutions
Alliance | Wiki | About DCR Solutions
Highlighter
Filter
Help
Copyright © DCR Solutions
Alliance | Wiki | About DCR Solutions

A comes up with an initial idea

A write a project pr oposal

A refine the idea

A check for plagiaris m

A cancel the propos al

A submit the propo sal

R the author

Re comes up with a n initial idea → refine the idea

Re comes up with a n initial idea → write a project propos al

Re write a project pr oposal →

The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. Everytime a new proposal is written, the author has the chance of checking for plagiarism once. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

Invariants:

Ensure that it only becomes possible to cancel the proposal after the author has come up with an initial idea

Ensure that the proposal cannot be cancelled after submitting and that it cannot be submitted after cancelling

Ensure that checking for plagiarism can be done once and only once every time the proposal is written

Ensure that the proposal always needs to be updated when the idea has been refined

Ensure that if the idea has been refined, then the proposal cannot be submitted before it has been updated

A comes up with an initial idea

A write a project pr oposal

A refine the idea

A check for plagiaris m

A cancel the propos al

A submit the propo sal

R author

Re Afterwards, it is possible to write a project proposal

Re Afterwards, it is possible to write a project proposal and to refine the idea at an y time

Re It is possible to c ancel the propos al

The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. Everytime a new proposal is written, the author has the chance of checking for plagiarism once. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

Invariants:

Ensure that it only becomes possible to cancel the proposal after the author has come up with an initial idea

Ensure that the proposal cannot be cancelled after submitting and that it cannot be submitted after cancelling

Ensure that checking for plagiarism can be done once and only once every time the proposal is written

Ensure that the proposal always needs to be updated when the idea has been refined

Ensure that if the idea has been refined, then the proposal cannot be submitted before it has been updated

Figure 7.1: *The highlights in Monocle created by two participants. The participant on the left only marked roles and activities while the participant on the right also highlighted relations.*

Of the case worker participants 4 did not use Monocle for relations. 2 case workers used it to add 3 relations and 1 case worker used it for adding 9 of a 13 total of relations. The results indicate that the initial attractiveness of using Monocle is rather low. It also shows that of the 8 participants who started using Monocle 5 switched to the modeller. This could be because they did not like using Monocle and that the benefits of it was not apparent to them. It could also be because they could not identify words to highlight. Since this process description has been designed to have the constraints explicitly mentioned other processes might be even harder for users to model with Monocle.

7.1.2 Process Maps

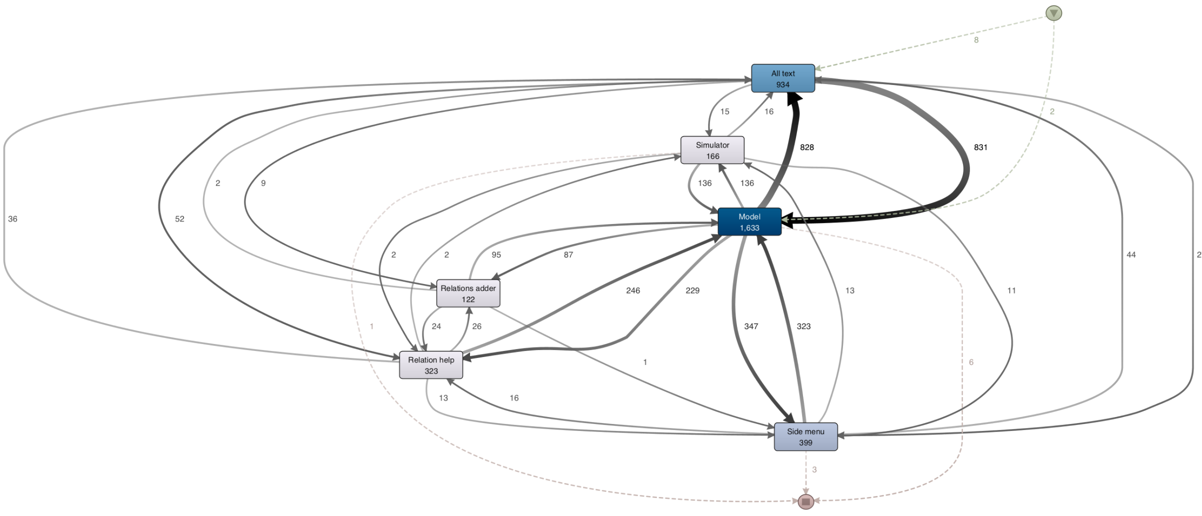


Figure 7.2: *Process map depicting the full task and all participants*[7].

Figure 7.2 shows the model with all participants for both tasks. The model shows that the model was looked to the most with a frequency of 1633 whereas the text in Monocle had a frequency of 934. It can be seen that these two AOIs also were the parts of the interface which the participants transitioned the most between with approximately 830 transitions each way. The map also indicates that almost every time the text was looked at, the previous AOI looked at was the model. This is also confirmed by looking at the other edges going to the text which all have fairly low frequencies with the highest one being 44 from the side menu. When looking at the Model it can be seen that it is the central activity with the most weight on its outgoing and incoming edges. This indicates that the model was the base of their attention an the place participants would return to after looking elsewhere. We can see that the side menu was looked at quite a significant amount of times (399) and thereby the third most looked at AOI. This is not expected as the side

[7] High resolution images of all process maps can be found in appendix 11.4

menu had very low relevance for the modelling task. There is a likelihood that a part of these cases are due to the model being next to the side menu and transitions to the side menu can be due to imprecise saccades aimed at the model (or eye tracking inaccuracies). However it seems unlikely that this explains all the dwells. Inspecting the statistics in Disco it can be seen that the median duration of looking at the side menu was 0.325 seconds. It can therefore indicate that the side menu is a distraction drawing unnecessary attention from the users.

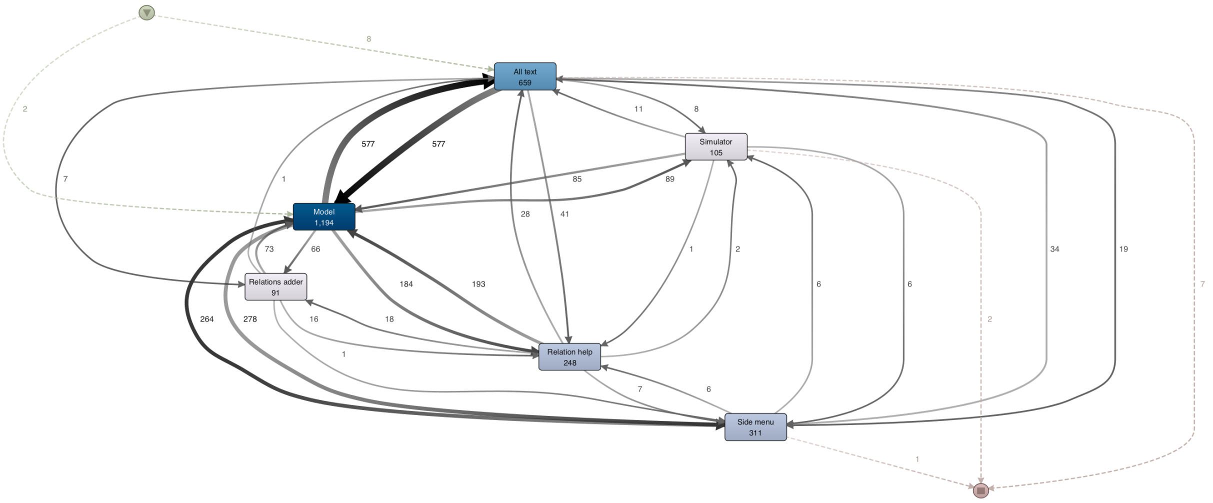


Figure 7.3: *Process map depicting modelling task.*

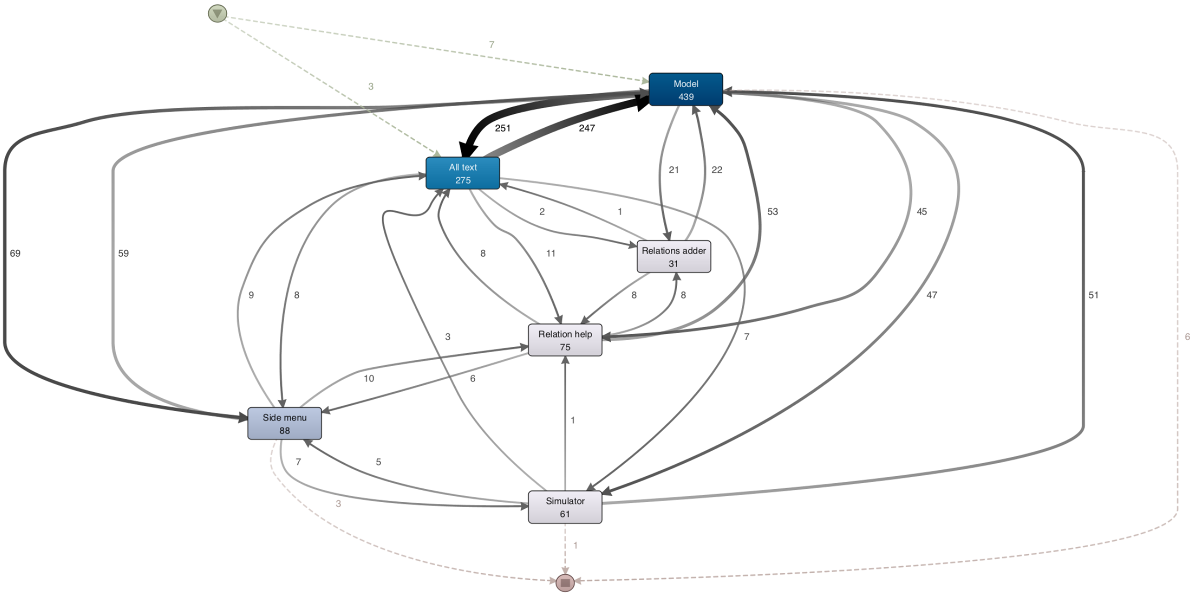


Figure 7.4: *Process map depicting invariant task.*

From figure 7.3 and 7.4 we compare the viewing patterns of the participants in the modelling task and the invariant task. In figure 7.3 we see the map from the modelling task and in figure 7.4 the invariant task. Overall the two models show quite the same tendencies. There is slightly more equal weight between model and text in the invariant task but nothing substantially. It can be seen that the simulator was used relatively more in the invariant task . This could indicate that the participants found this task more challenging. The relation helper was used more in the modelling task which indicate they used it in the beginning to better understand the relations.

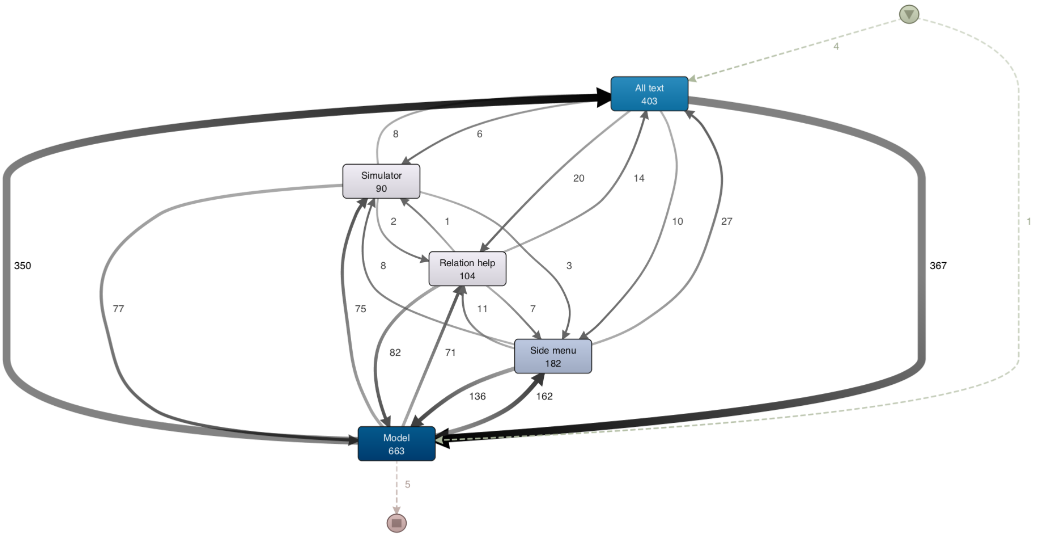


Figure 7.5: *Process map depicting modelling relations with modeller.*

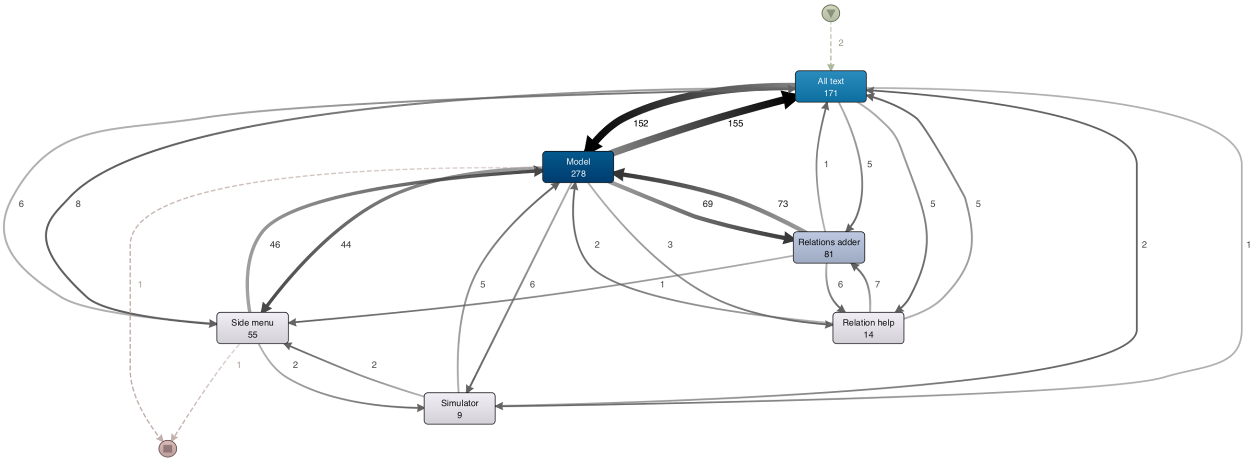


Figure 7.6: *Process map depicting modelling relations with Monocle.*

In figure 7.5 and 7.6 the participants who use the modeller for adding relations (figure 7.5) are compared with the participants who used Monocle (figure 7.6). Starting by comparing their use of text and model we see that it is quite the same between the two groups. It could have been expected that the participants using Monocle for relations would have looked more times in the text, but that is not the case. It can be seen that the participants who added relations using Monocle looked fewer times to both the help and the simulator. The participants who used Monocle were therefore more certain in their modelling which might have been the reason they used Monocle which could again indicate that inexperience might have influenced why the majority did not use Monocle for relations.

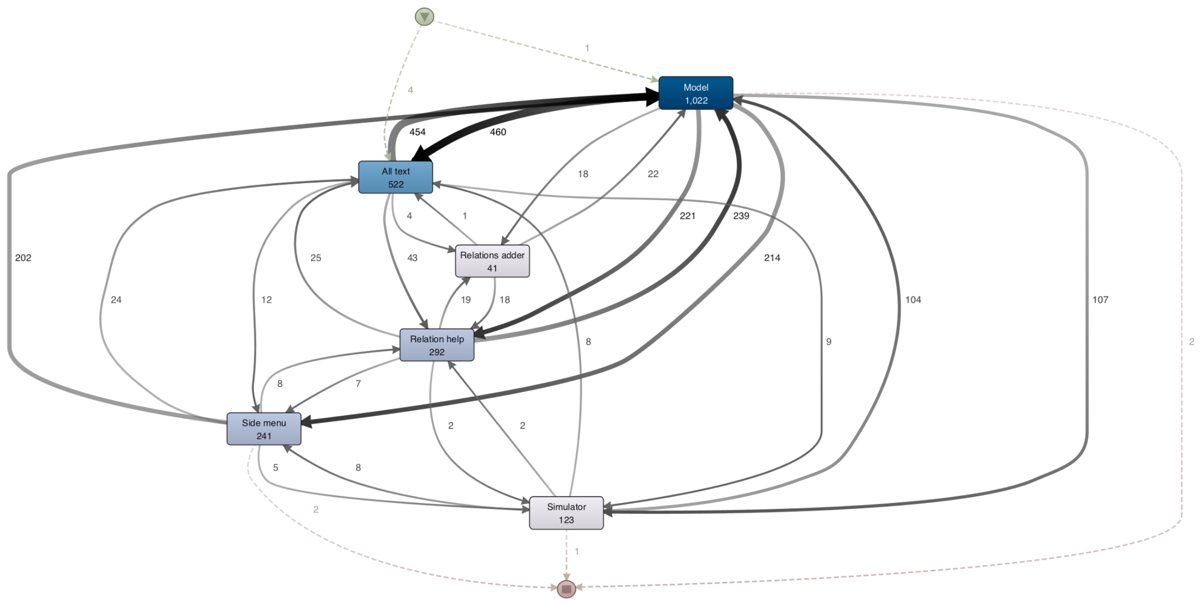


Figure 7.7: *Process map depicting participants with low confidence.*

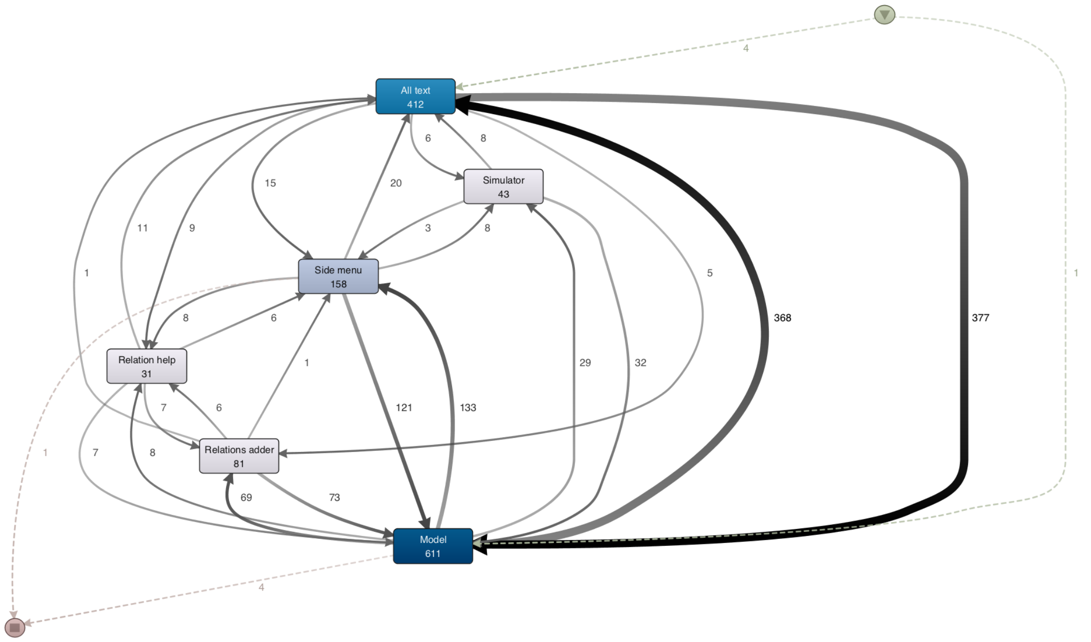


Figure 7.8: *Process map depicting participants with high confidence.*

Figure 7.7 shows the model of participants with low modelling confidence and figure 7.8 shows participants with high modelling confidence. It is quite clear that there is a difference between the two groups in the distribution between the model and the text. By inspecting statistics in Disco it can be seen that the relative frequency for looking at the model for both groups is 45% however for looking at the text the participants with high confidence had a relative frequency of 31% where the low confidence group spent 23%. It therefore seems that participants with lower confidence are less focussed on the text.

Inspecting further in disco we can see the median dwell duration in these AOIs between our two groups. The median time of looking at the text were approximately the same about 1.4 seconds however the median duration of looking at the model was 1.37 for participants with low confidence and 0.816 for participants with high confidence. This means that in terms of duration high confidence spend less time looking at model. Another thing that can be seen is a higher amount of transitions between the relations adder and the model for participants with higher confidence. This again indicate that the participants who were more secure modelling were more likely to use Monocle for relations.

7.2 RQ2

During the analysis and coding 7 categories of the codes emerged. In this part each of the categories will be presented along with what can be derived from the content. For each category a table will be presented containing the codes and number of participants who attested the code.

Usage of the tools

Code	No. Of Participants
I identified relations in the text and added them directly in the modeller	4
After the initial steps of making activities I mainly went between modeller and simulator	3
I start with Monocle for activities and roles, but then I quickly jump to the modeller to model	2
After making activities, I set it up visually in modeller - Then it is intuitive to keep working in the modeller with relations	2
Used Monocle for relations but would prefer the modeller	1
After creating the activities I set them up visually in the modeller	1
I made relations from modeller but planned to assign it to text afterwards -But mainly because of this exercise	1
I was able to find all relations with Monocle except a few I added with the modeller	1

Table 7.1: *The table show the codes belonging to the category 'Usage of the tools'.*

In table 7.1 we see the Usage of the tool codes. The codes show a rather similar picture of how the participants created the model. In several it is indicated that activities were created first using Monocle and that the relations were then created afterwards with the modeller. A few participants also comment on their logic behind: First you arrange the activities in the modeller and then it is intuitive to continue modelling there. There are also a few which explains how Monocle was used, but even here 1 of the 2 explains how he would have preferred to use the modeller.

Strong in Monocle	
Code	**No. Of Participants**
Helps to easily and intuitively find roles and activities	6
Monocle is good for overview and structure	5
Moncole add traceability for seeing what has been covered	3
Monocle is a great help for getting startet and makes the task less confusing	2
Monocle can quickly make activities	1

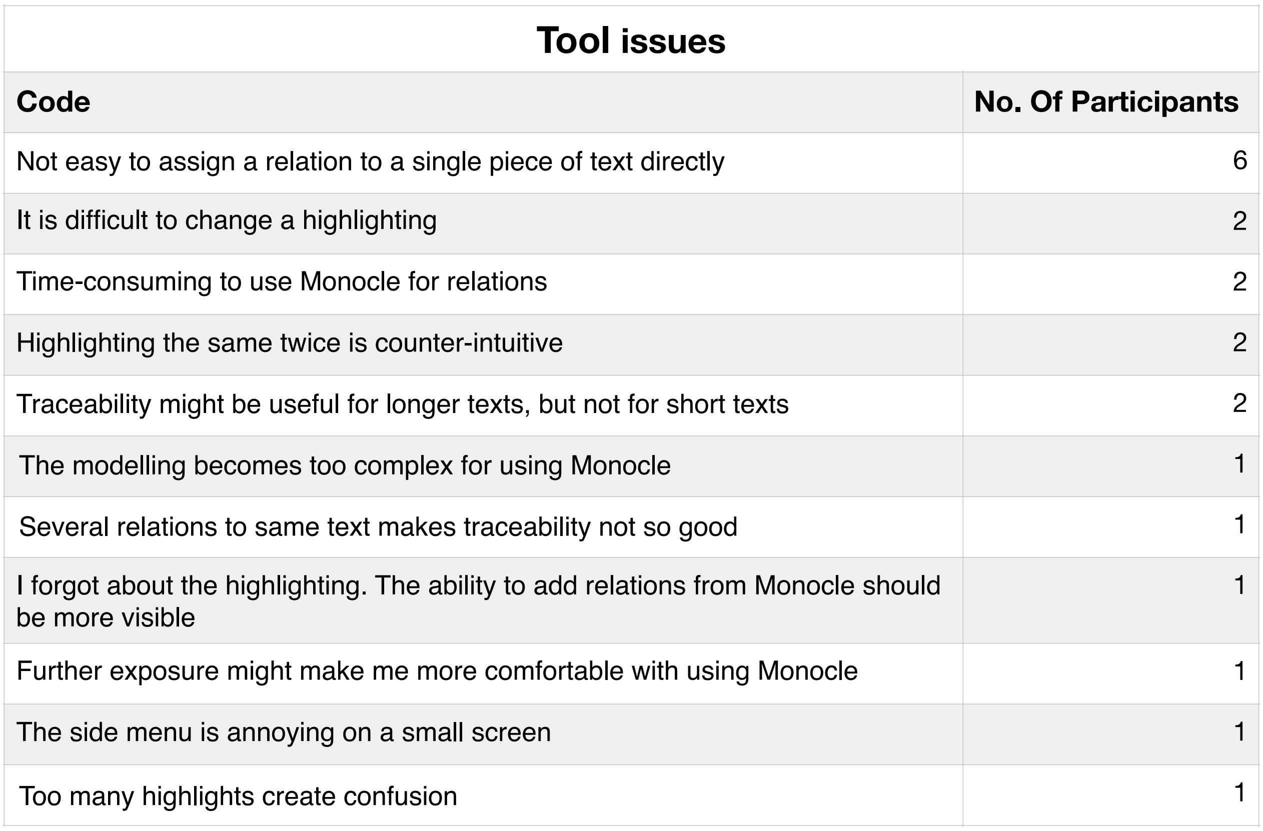
Table 7.2: *The table show the codes belonging to the category 'Strong in Monocle'.*

Table 7.2 show what the participants saw as positive with using Monocle. The two main aspects mentioned was to use Monocle for activities and roles and for getting a clear overview. Only 3 participants mentioned traceability as a benefit they saw from the tool.

Strong in modeller	
Code	**No. Of Participants**
It is easier and faster to use modeller for relations	3
Intuitive to directly connect them in model	2
Modeller is more interactive	1
Modeller gives a better visualisation of what you are doing	1

Table 7.3: *The table show the codes belonging to the category 'Strong in modeller'.*

In table 7.3 we see the positive aspects about the modeller which were mentioned. The participants indicated that it was faster and more intuitive to add relations with the modeller.



Tool issues

Code	No. Of Participants
Not easy to assign a relation to a single piece of text directly	6
It is difficult to change a highlighting	2
Time-consuming to use Monocle for relations	2
Highlighting the same twice is counter-intuitive	2
Traceability might be useful for longer texts, but not for short texts	2
The modelling becomes too complex for using Monocle	1
Several relations to same text makes traceability not so good	1
I forgot about the highlighting. The ability to add relations from Monocle should be more visible	1
Further exposure might make me more comfortable with using Monocle	1
The side menu is annoying on a small screen	1
Too many highlights create confusion	1

Table 7.4: *The table show the codes belonging to the category 'Tool issues'.*

Table 7.4 shows the issues that were mentioned and associated with the tool. We see that the majority of issues mentioned lies with using Monocle for relations. The issues faced by most participants was to find the right text to highlight for the relations which was mentioned by 6 participants. This was said even though the text had been designed to have all relations explicitly mentioned in the text. It indicates that to most users it is not a simple task to identify the words to highlight when highlighting relations. Besides this the participants mentioned that Monocle was too slow. Two participants also stated that traceability was not useful since the text was not so long.

Usage of the simulator	
Code	**No. Of Participants**
I used simulator to verify what I had built really worked	8
I used the simulator when having modelled most, to see if everything was working as intended	2
I used simulator to find out how some relations worked	1
I went back and forth between building and the simulator	1
It gives a nice testing flow	1
I used the simulator everytime I created a new relation	1

Table 7.5: *The table show the codes belonging to the category 'Usage of the simulator'.*

Table 7.5 show the comments they had about the simulator. The simulator was mainly used to verify the model. It also seemed that some participant used the simulator to learn how to model.

Implementing invariants	
Code	**No. Of Participants**
I had created some of the invariants before I saw them	6
I created my own invariants that were not mentioned	2
I only created the relations I were specifically told to add	1
I added some relations that were not clear from the text	1

Table 7.6: *The table show the codes belonging to the category 'Implementing invariants'.*

Table 7.6 show comments about adding invariants to the model. It can be seen that the majority of the participants had added invariants by themselves with only one indicating that they did not add any by themselves. This could be explained by either (a) participants add the relations from the text and consciously decide to also include the invariants, (b) when the participants are mentally translating the textual description to model they are unconsciously adding additional relations. With the data collected this cannot be further investigated however it potential that using the modeller for modelling can make users unconsciously add additional invariants not specifically mentioned.

Suggestions for improvements	
Code	**No. Of Participants**
Ability to exclude activities from Monocle	1
Add all instances of a word as role automatically	1
Auto suggest of verbs as relations	1
Guides for common ways of setting the arrows	1
Ability to format text in Monocle	1
Clearer colors in Monocle	1

Table 7.7: *The table show the codes belonging to the category 'Suggestions for improvements'.*

During the interviews the participant came with a number of suggestions that they thought could improve DCR graphs. These suggestions have been collected in table 7.7 under the category 'Suggestions for improvements'. There were no suggestions that were mentioned more than once and the complexity of the suggestions ranged from changing the colour scheme to automatic verb identification.

7.3 Summary

This section will sum up the main findings which were identified in the results of the data analysis. The section only includes aspects that were indicated or observed in a substantial number of participants.

- The participants used Monocle for adding activities and many found it to be easy and intuitive. They also found it to be helpful for getting an overview of the task when getting started.
- 8 of the 17 participants used Monocle for relation however 5 only added a few relations and instead moved over to using the modeller.
- The participants found it easier to create relations using the modeller. Many participant had troubles finding the right words to highlight which added extra work and time needed.
- Low experience might influence users to opt out from using Monocle as it initially requires higher workload.
- The side menu might be a distraction to users

- The model is the most central AOI in the interface which most participants have their gaze based on. This is the case for both participants modelling relations with Monocle and with the modeller. However, participants with a higher confidence in their modelling spend relatively less time looking at the model and relatively more time looking in the text.
- Many participants added invariants before they were asked to. This could indicate an unconscious process taking place when translating from natural language to the model that changes the nature of the process. Using Monocle might prevent users from doing this.

8 Discussion

This project set out to investigate and explore Monocle tool using eye tracking, process mining and qualitative research. This section will discuss the results and the methods used for obtaining them. It will highlight the positive and negative outcomes and discuss what could have been done differently. The chapter will end with a future works section which will discuss the steps that could be done for taking the project further.

8.1 Results and Methods

The Monocle tool has been created to help users create and maintain models. This project has especially investigated the tool for it's model creation capabilities. The results indicates that it is not attractive to model relations using Monocle. Of the 17 participants 9 did not add a single relation using it and only 3 participants chose to continue using Monocle for adding relations. This shows us that the initial attractiveness of Monocle is rather low. Users disregards it as it requires more work and is less intuitive. However some of the benefits of Monocle might not be apparent from the first impression of using it. The traceability benefit was identified by 3 participants however this benefit might be more apparent for modelling longer processes or for maintaining models. This aspect is still interesting and has only been touched upon lightly. Even if Monocle is cumbersome in the early stages it might lower the cognitive load when creating more complex models. As mentioned users might take an opportunistic approach for longer processes which require re-comprehension of their own work. Monocle can therefore give a clearer view of what has been modelled and what has not. For better testing the aspect of traceability the experiment could have been designed differently. Adding a longer textual description would not have been an option for this experiment as the timeframe of the experiment was already in the high end of what you can expect from volunteer participants. Instead an approach that could have been taken would have been to create a test that mixed model creation with model comprehension. Such a test could contain a longer text with an unfinished model which the participants would have to complete or update. Such a test could perhaps better demonstrate the traceability aspect of Monocle since the participants would need to understand the model in order to finish building it. It could then be investigated if the negative aspects of the tool, such as the issue of identifying text, becomes outweighed by the positive aspect of traceability.

For experiment one it was not possible to recruit participants with experience using DCR. Instead most of the participants had never used DCR graphs before. This aspect has most likely influenced the results. The results can be defined as applicable to new users but it is not certain the same holds true for more experienced users. It could therefore have been interesting to also have tested on more experienced users to see if their approach is different. Users with more experience might also have been more inclined to using Monocle. The reason so few participants chose to use Monocle for relations might be related to the fact that they had to learn so much new for the test and choosing the modeller for relations was one way to immediately reduce the effort needed. More experienced users might therefore also have provided more direct insight into the usage of Monocle as many of the participants only tried adding relations from Monocle in the familiarisation session.

8.2 Future Work

For future work there are several aspects which could be focussed on. Since the GSR data has not been analysed this could be a starting point for moving further with the project. As indicated Monocle might have an influence on the cognitive workload of the user. The GSR data might therefore be able to shed more light onto this aspect. This aspect can also be investigated with the eye tracking data as pupil dilation is influenced by cognitive workload.

Another thing that could be done is investigating the models the users created further. It could be analysed with test-cases how accurate the models they had created were. This could again help understand which users were better at modelling to see how this influenced how they modelled.

The data could also be explored further by cross checking between the different datasources. An example of what could be explored is whether the participants who abandon Monocle are the same participants who state that it is difficult to identify relations in the text.

Finally the results from this work could be used to dig deeper into interesting aspects. For once it could be interesting to understand if Monocle is more useful for adding relations in longer or more complex processes or if the issue of identifying the right words is still too big an issue. It could also be interesting to dig deeper into why users add invariants, which are not mentioned, to the model and if it has something to do with the modeller and whether Monocle can prevent this.

9 Conclusion

Throughout this project the Monocle tool of DCR has been investigated. Two research questions were conceived to drive the investigation of the tool. To gather data about the usage two experiments have been carried out in which participants have been creating process models using the tool. Their usage and view of the the tool has been collected with eye tracking, interviews and the models they have created. This data has then been used to investigate the research questions. Using the eye tracking data the interface was divided into AOIs. This was exported as an event log which was used to discover process maps using the tool Disco. The interviews were transcribed and qualitative coding was applied. The data was then analysed taking an exploratory approach.

Monocle was created to facilitate model creation and maintenance. This project investigated how users engage with the tool for creating models. The results show that users find the tool helpful for getting an initial overview and for adding activities and roles. However for adding relations most users prefer the modeller. It is indicated that Monocle requires more initial work for creating relations and that the traceability aspect is not outweighing this issue. It is however indicated that experience might influence this perception. The results also showed that users might be unconsciously adding invariants to the model and that using the modeller might be implicated in this.

10 Bibliography

1. Mukkamala RR, Hildebrandt T, Slaats T. Towards Trustworthy Adaptive Case Management with Dynamic Condition Response Graphs. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference. IEEE; 2013:127-136. doi:10.1109/EDOC. 2013.22.

2. Debois S, Hildebrandt T, Marquard M, Slaats T. The DCR Graphs Process Portal. http:// www.itu.dk http//www.exformatics.com http://www.diku.dk/. Accessed December 30, 2018.

3. López HA, Debois S, Hildebrandt TT, Marquard M. The Process Highlighter: From Texts to Declarative Processes and Back. https://bpmai.org/foswiki/pub/BPMAcademicInitiative/. Accessed December 29, 2018.

4. Vaisman A. An Introduction to Business Process Modeling. In: Springer, Berlin, Heidelberg; 2013:29-61. doi:10.1007/978-3-642-36318-4_2.

5. van der Aalst W. Process Mining. Berlin, Heidelberg: Springer Berlin Heidelberg; 2016. doi: 10.1007/978-3-662-49851-4.

6. Haisjackl C, Zugal S, Soffer P, et al. Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls. In: Springer, Berlin, Heidelberg; 2013:2-17. doi: 10.1007/978-3-642-38484-4_2.

7. Pichler P, Weber B, Zugal S, Pinggera J, Mendling J, Reijers HA. Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In: Springer, Berlin, Heidelberg; 2012:383-394. doi:10.1007/978-3-642-28108-2_37.

8. Fahland D, Lübke D, Mendling J, et al. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: Springer, Berlin, Heidelberg; 2009:353-366. doi:10.1007/978-3-642-01862-6_29.

9. Andaloussi A, Slaats A;, Burattin T;, Hildebrandt A;, Weber TT. Understandability of Hybrid Process Models Using DCR Graphs. APA; 2018. https://ecoknow.org. Accessed December 30, 2018.

10. DCR. https://wiki.dcrgraphs.net/wiki/68/dcr. Accessed December 30, 2018.

11. Dynamic Condition Response (DCR) - Welcome to DCR Solutions. https:// dcrsolutions.net/. Accessed December 30, 2018.

12. About – Effective co-created & compliant adaptive case management for Knowledge workers. https://ecoknow.org/about/. Accessed December 30, 2018.

13. Debois S, Hildebrandt T, Slaats T, Marquard M. A Case for Declarative Process Modelling: Agile Development of a Grant Application System. In: 2014 IEEE 18th International

Enterprise Distributed Object Computing Conference Workshops and Demonstrations. IEEE; 2014:126-133. doi:10.1109/EDOCW.2014.27.

14. Debois S, Hildebrandt TT, Laursen PH, Ulrik KR. Declarative process mining for DCR graphs. In: Proceedings of the Symposium on Applied Computing - SAC '17. New York, New York, USA: ACM Press; 2017:759-764. doi:10.1145/3019612.3019622.

15. Jacob RJK, Karn KS. Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. Mind's Eye. January 2003:573-605. doi:10.1016/B978-044451020-4/50031-1.

16. Holmqvist K, Nyström M, Andersson R, Dewhurst R, Halszka J, Weijer J van de. Eye Tracking : A Comprehensive Guide to Methods and Measures. Oxford University Press; 2011. http://portal.research.lu.se/portal/en/publications/eye-tracking--a-comprehensive-guide-to-methods-and-measures(03ff6e58-7456-448f-9f45-1c3660bc05f4).html. Accessed December 30, 2018.

17. Hyönä J. The use of eye movements in the study of multimedia learning. Learn Instr. 2010;20(2):172-176. doi:10.1016/J.LEARNINSTRUC.2009.02.013.

18. 7 Most Used Eye Tracking Metrics and Terms - iMotions. https://imotions.com/blog/7-terms-metrics-eye-tracking/. Accessed December 30, 2018.

19. Günther CW, Rozinat A. Disco: Discover Your Processes. https://pdfs.semanticscholar.org/56b5/96c0762224d612a2cbf9b873423a12ff5b6b.pdf. Accessed December 29, 2018.

20. Günther CW, Van Der Aalst WMP. Fuzzy Mining-Adaptive Process Simplification Based on Multi-Perspective Metrics. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.1207&rep=rep1&type=pdf. Accessed December 29, 2018.

21. Lewis S. Qualitative Inquiry and Research Design: Choosing Among Five Approaches. Health Promot Pract. 2015;16(4):473-475. doi:10.1177/1524839915580941.

22. Jaspers MWM, Steen T, Bos C van den, Geenen M. The think aloud method: a guide to user interface design. Int J Med Inform. 2004;73(11-12):781-795. doi:10.1016/J.IJMEDINF.2004.08.003.

23. Lazar J, Feng JH, Hochheiser H. Research Methods in Human-Computer Interaction. Wiley; 2010. https://www.goodreads.com/book/show/6206730-research-methods-in-human-computer-interaction. Accessed December 30, 2018.

24. Charmaz K, Belgrave LL. Grounded Theory. In: The Blackwell Encyclopedia of Sociology. Oxford, UK: John Wiley & Sons, Ltd; 2015. doi:10.1002/9781405165518.wbeosg070.pub2.

25. Bryman A. Social Research Methods. Oxford University Press; 2004.

26. Saldaña J. The Coding Manual for Qualitative Researchers. Sage; 2009.

27. Kothari CR. Research Methodology : Methods and Techniques. New Age International; 2004.

28. Reichert M, Weber B. Enabling Flexibility in Process-Aware Information Systems : Challenges, Methods, Technologies. Springer; 2012.

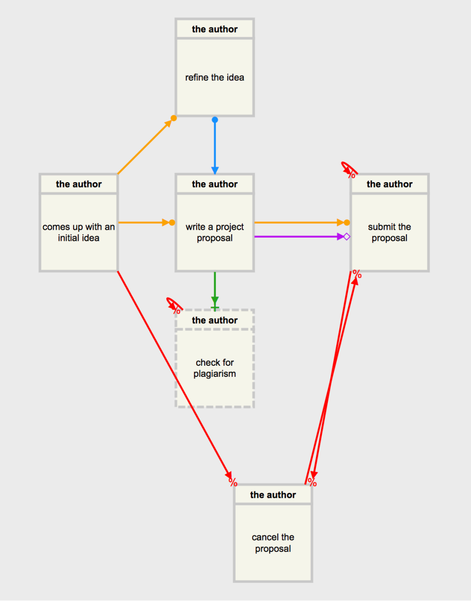
29. Tobii Pro X3-120 Eye Tracker User Manual User Manual Tobii Pro X3-120 Eye Tracker.; 2017. www.tobiipro.com. Accessed December 30, 2018.

30. Tobii Studio User's Manual User's Manual Tobii Studio. www.tobiipro.com. Accessed December 30, 2018.

31. Wearable GSR Sensor | Shimmer Galvanic Skin Response Sensor | EDA sensor. http://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor. Accessed December 30, 2018.

11 Appendix

11.1 Process Description for Test



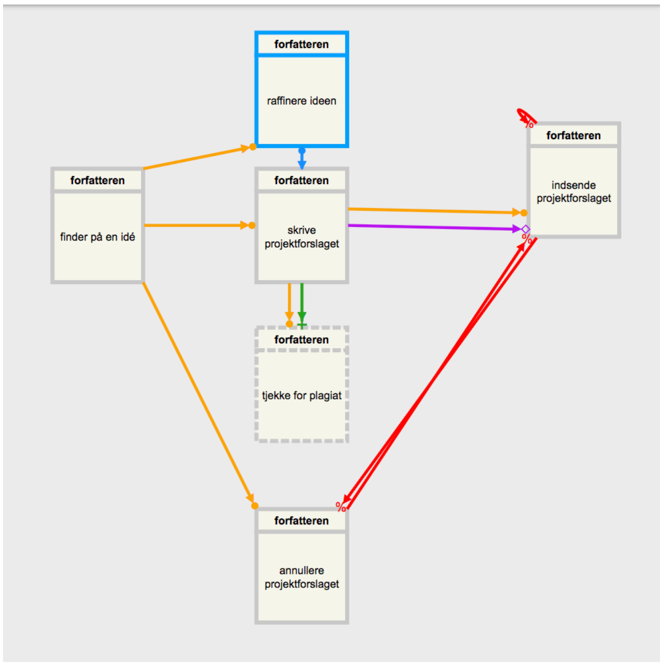
The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. Every time a new proposal is written, the author has the chance of checking for plagiarism once. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

Invariants:

1. Ensure that it only becomes possible to cancel the proposal after the author has come up with an initial idea
2. Ensure that the proposal cannot be cancelled after submitting and that it cannot be submitted after cancelling
3. Ensure that checking for plagiarism can be done once and only once every time the proposal is written
4. Ensure that the proposal always needs to be updated when the idea has been refined
5. Ensure that if the idea has been refined, then the proposal cannot be submitted before it has been updated

The process can be viewed on DCR graphs with the following link:
https://www.dcrgraphs.net/Tool?id=8844#

11.2 Danish Translation of Process Description for Test



Lav en process model over følgende tekst:

Processen for at skrive et projektforslag starter med at forfatteren finder på en idé. Bagefter er det muligt at skrive projektforslaget og raffinere ideen på hvilket som helst tidspunkt. Efter at have skrevet projektforslaget er det muligt at tjekke for plagiat. Hver gang projektforslaget er blevet genskrevet er det muligt at tjekke for plagiat én gang. Det er muligt at annullere projektforslaget hvis det viser sig at være en uholdbar idé. Ellers er det muligt at indsende projektforslaget når det er færdigskrevet. Noter at et projektforslag kun kan blive indsendt én gang.

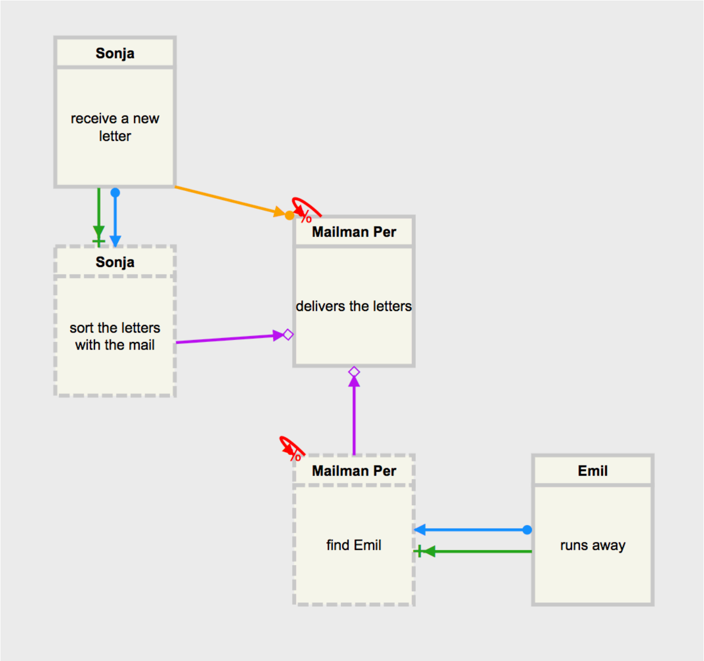
Efter at have lavet modellen, garanter da at følgende udsagn bliver opfyldt af modellen:

1. Projektforslag kan ikke blive annulleret efter det er blevet indsendt, og ligeledes kan det ikke indsendes hvis det er blevet annulleret.

2. Projektforslaget skal blive opdateret efter ideen er blevet raffineret.

The process can be viewed on DCR graphs with the following link:
https://www.dcrgraphs.net/Tool?id=8392

11.3 Familiarisation Process description



Part 1
A day for Sonja and Mailman Per:
In the mail office Sonja is responsible for the mail. At any time Sonja can receive a letter. When receiving mail it is necessary that Sonja sorts new letters with the rest of the mail. Once a day Sonja's husband Mailman Per delivers the letters. Mailman Per only delivers if atleast one new letter has been received and if all received letters have been sorted.

Invariant:
Ensure that Sonja can only sort once after receiving new letters

Part 2
However problems can occur! If his cat Emil runs away then Mailman Per must find Emil before he can deliver the letters. Emil can sometimes run away several times.

Invariant:
Ensure that Mailman Per is only able to find Emil once after he runs away.

The process can be viewed on DCR graphs with the following link:
https://www.dcrgraphs.net/Tool?id=8367

11.4 Online Appendix

https://drive.google.com/open?id=1vBx2s9PrVAAF8Etq2I4OYOZyPbmwLXL3

The above link contains the online appendix of the thesis. The online appendix includes:

- **Images of the process maps used for the data analysis**
- **The questionnaire and participants answers**
- **The AOI event log**
- **Transcription file for the interviews**
- **Consent form**
- **Slideshow for familiarisation session**

To see the raw data or have any other inquiries please contact the author: jonbuchl@gmail.com