MASTER

The RecLog%5E2 method for process improvement using historical data to improve work-item assignment

van Bussel, M.W.H.A.C.

Award date:
2012

Final version March 22, 2012

Master Thesis

The RecLog^2 method for process improvement:

Using historical data to improve work-item assignment

Author:
Mike van Bussel (0578052)
M.W.H.A.C.Bussel@delagelanden.com
M.W.H.A.C.v.Bussel@student.tue.nl

Supervisors:
ir. Harm Hoebergen
H.T.W.Hoebergen@delagelanden.com
dr.ir. Irene Vanderfeesten
I.T.P.Vanderfeesten@tue.nl
dr.ir. Boudewijn van Dongen
B.F.v.Dongen@tue.nl

Abstract

Management is always looking to improve operational processes. In this master thesis the RecLog^2 method is developed and implemented to provide the management with a new process improvement tool. The result of a successful implementation of the RecLog^2 method will yield recommendation logic to manage an operational service process. The method is not suitable for all operational processes. The process needs to have a service orientation. The method presented consists of two phases each containing a number of steps. The first phase in this method is the defining phase. It describes how to obtain necessary information and define the hypotheses that serve as input for the second phase. This second phase, the implementing phase consists of steps to validate the hypotheses, select the appropriate relations and define the actual recommendation logic. The RecLog^2 method allows the construction of recommendation logic for a huge number of operational processes, that have a service orientation. It implements the proven added value of recommendation systems with a new focus on the internal entities interacting in the process as opposed to the external entities. The method is set up to ensure improving the performance of the key performance indicators of the process. Besides the developed method this master thesis also provides an implementation of it within the operational process of Freo.

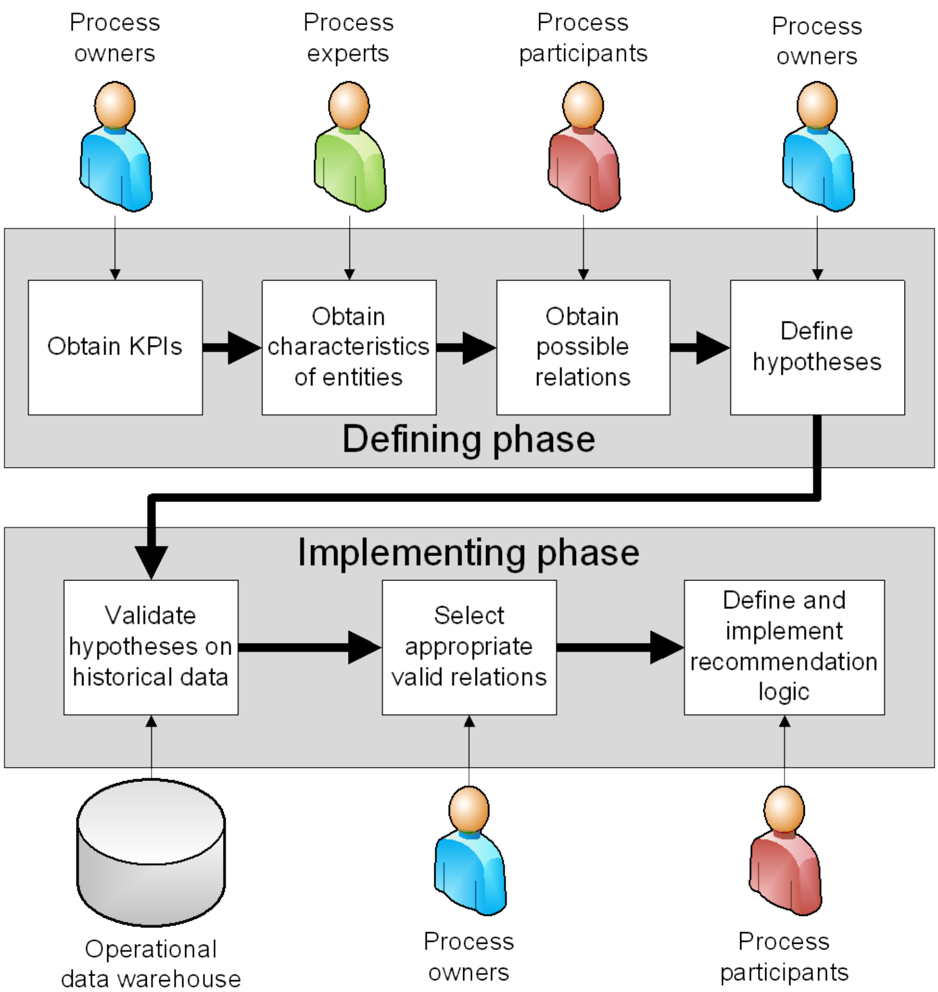


Figure 1: The RecLog^2 method

Preface

The master thesis you are now reading is the result of my graduation project, which completes my Business Information Systems study at Eindhoven University of Technology. The main goal of the project is to develop a method to set up (internal) recommendation logic. The method was implemented at De Lage Landen Eindhoven (DLL). DLL is a global provider of leasing, business and consumer finance solutions, including vendor finance and factoring. Freo is one of the brands that is used by DLL for online consumer credit. The developed method of this project focuses on the application process of Freo and especially the interaction between the call agents and the applications. This is to serve as a validation of the method. This project is the final assignment that marks not only the end of my educational career, but also the start of my professional career, in which the learning will continue.

This work is of high quality thanks to the involvement of a large number of people. I therefore would like to thank everybody who participated in any way in the creation of this master thesis. I would like to give special thanks to some of these valuable contributions. First of all, I express my sincere gratitude to my supervisors Irene Vanderfeesten, Boudewijn van Dongen and Harm Hoebergen who helped me tremendously during my graduation project, provided an enormous amount of crucial advise and exhibited unusually flexibility in dealing with my deadlines. Furthermore I would like to thank the Freo management team for their openness and willingness to test the method in practice. They created an ideal working environment for my graduation project. In particular Harold Schuurman and Pepijn Pullens are thanked for their active participation throughout the process. Also I am very grateful for the managers of the datawarehouse of DLL, who even with there already overfilled agendas still found time to help the project whenever it was stuck. Therefore, many thanks to Jef Hopmans, Gaston Spronk and Javier Hernandez Albarran. I hope I have repaid you with enough cups of coffee. And last, but certainly not least, I would like to thank Emiel van Berkum who helped me to understand the statistics that are used during the validation process. All other people are of course also thanked, their combined contribution was critical for this splendid result.

To conclude this section I would like to dedicate this master thesis to my lovely fiancée, Charlotte Strooband, who inspired me not only to do things, but to do things in an excellent fashion. Without her the level of quality would have been dramatically lower. This thesis is also dedicated to my close family for their support during my academic career and their unquestionable believe that eventually I would succeed. Their willingness to work hard, pushed me to work hard as well. Especially my sister, Mayke van Bussel, showed me what the true meaning of perseverance is. Without these people I would not have become what I am today and for that I will always be in debt to you.

Mike van Bussel
March 2012

Table of contents

List of figures

List of tables

List of appendix figures

List of appendix tables

1. Introduction

In this chapter the goal of the graduation project is described. Then the main research question is presented. The answer to this research question is provided in this thesis by developing a new method for process improvement. The last section of this chapter explains the structure of this master thesis.

1.1. Goal

The main goal of this master thesis is to develop an method for defining and implementing recommendation logic to improve an operational process. This method, which will be called the RecLog^2 method, is to provide process owners with a new tool to improve the process performance. A large number of processes these days are supported by information systems and can therefore be identified as a workflow process in which a predefined set of procedures and rules needs to be obeyed. An important precondition for the RecLog^2 method is that the operational activities of the entities within the process are logged in a datawarehouse. The guidance of the available work items can also be seen as a sort of voluntary workflow engine. It provides added value to internal resource entities by recommending which task to perform next. Another crucial precondition is the service orientation of the process. The developed method assumes that internal and external entities interact with each other. This interaction can be influenced by the recommendation logic that is designed with the RecLog^2 method. Although not all operational processes fit this precondition, a large number of them does. The operational process referred to in the remainder of the master thesis is one with this service orientation. The method lends the ideas of Product-based workflow support [Vanderfeesten et al, 2011] to look at specific entities in stead of general ones and letting the resource entity determine the next activity on it, but at the same time also helping the resource entity to select the next work item with the use of recommendation logic. The recommendation logic is created within the already established operational process and is inserted in it. This is done so that the desired outputs of the resource entities improve and at the same time the work of the resource entities becomes easier, simpler and/or faster. There already exist rerouting systems for, for instance incoming (customer) calls to the best fitted call agent, which are developed on some what similar principals, but they have an exclusively external focus. The customer has to indicate what kind of help he would like to receive. The RecLog^2 method developed in this graduation project allows this to also work with an internal focus, designing recommendation logic that provides a recommendation to any entity. Any expected and found relation, which can be implemented and can be customized to fit the characteristics of the operational process, can serve as a base for the recommendation logic. Relations or dependencies that exist within the process which have influence on process key performance indicators are good candidates for the recommendation logic. This to design recommendation logic that allows process entities to interact based on a relation or match between entity characteristics. This idea and the method to design the recommendation logic are explored in the remainder of this master thesis and lead to the development of the RecLog^2 method.

1.2. Research question

Below the main research question and its intention are presented. The main question is answered by providing a new method. The combined insights and results of the steps within the method allow correctly answering the main research question. As will be seen later, the $RecLog^2$ method is provided as the answer to the main research question. The main research question is formulated as follows:

How to improve performance of an operational service process on its KPIs, using historical data for improving work-item assignment?

The work-item assignment is treated in the remainder of this thesis as recommendation logic. Both are based on dependencies or relations between characteristics of interacting entities during the services within the operational process. The goal of the recommendation logic is to positively influence the key performance indicators (KPIs). Because the recommendation logic influences the interaction between entities, these entities must interact with each other during services within the process. These services should be between internal and external entities and need to be logged. The logging of the activities and transactions is necessary to be able to validate any dependencies between them.

1.3. Structure master thesis

The rest of this master thesis follows the following structure. The next chapter explains the preliminaries, the motivation for the research, it introduces the Freo application process in which the $RecLog^2$ method is implemented for validation and it explains some key elements that are used in the method. In chapter 3 the developed $RecLog^2$ method is explained in detail step by step. The defining phase during the Freo implementation is described in chapter 4, following with the implementing phase in chapter 5. The method is executed with the help of the process owners. The recommendation logic was set up with the purpose to improve process performance on its KPIs. The conclusions of this master thesis are provided in chapter 6. In the same chapter a reflection of the process is provided together with the future work. The chapter after that contains the appendix.

2. Preliminaries

This chapter explains the preliminaries. It contains the motivation for the research, it introduces the Freo application process in which the RecLog^2 method is implemented for validation and it explains some key elements that are used in the method.

2.1. Research motivation

In an earlier part of the graduation project the added value of process mining and the process mining tool ProM is showed to DLL by applying the principles to two operational process, one of which is described in the section 2.2. To be able to check process model conformance, first an process model was created. Also effort is spent defining desired management information and the measurement points in the process to be able to extract this management information. During the necessary activities to achieve this, more and more process insight was gained and process improvement ideas began to spawn. One of these ideas regarded the manner in which call agents selected the next application to work on, or more generally the resource entity selecting its next work entity. At that point in time, no real allocation logic was implemented in the process to guide the resource entities in their choice. The question arose what recommendation logic could be implemented in such cases. There are a lot of known strategies that can serve as logic to assign work. For instance a random assignment, first come first serve or earliest due date [Panwalkar et al, 1977] [Silver et al, 1998]. These strategies can be used, but for the given operational service process there was not much difference between the work entities. They all look alike. Therefore no logic was implemented, although sometimes the resource entity used the first come first served logic. This was however an exception as usually the selection was random. Because of this, time is wasted during the selection process. This could be improved. To determine when a proposed solution would be an improvement with regard to the current situation one would have to look for the most important and most desired results for the process. This is done by identifying the key performance indicators (KPIs). This is explained later. As the standard well known logic to provide guidance during the selection were not applied and to some degree not suitable for the given process, the main research question, as formulated earlier, came to be. In search of a starting point for the method the attention was drawn to recommendation systems. The idea behind the developed method is to use these techniques to recommend a certain entity to another one. As a recommendation system usually provides a recommendation to external entities, the method should also allow to provide an internal entity with a recommendation to help resource entities in operational service processes to receive work entities that they will probably be best fit for or that they will probably like.

2.2. Background Freo application process

The RecLog^2 method is implemented in the Freo application process of DLL. To introduce the Freo application process and explain some of the context in which the method is developed, the Freo application process is described in this section. The aim of the application process is to provide consumer credit through the Freo website. The process is modeled during an earlier project. The process model that was a deliverable of this project is added to this thesis as App. A figure 1 and App. A figure 2. Because of its unique and suitable characteristics, the Freo application process is a very interesting choice for the implementation of the method. The high volume of applications, between the 400 and 600 a week, and the rich data, which is available in the datawarehouse, are perfect to find interacting characteristics, relations of influence and extract benchmarks for key performance indicators. This rich information is also very useful when performing process mining. Which, among other activities, is performed in the first part of the graduation project. To illustrate the flow of an application the process is explained in the rest of this section.

2.2.1. Process participants

There are three groups of human resources that interact with the customers primarily. They are supported by a number of information systems. The human resources are the call agents, the administrative staff and the acceptation assessors. Their most important support system is Advisor. This information system provides functionality as presenting the work items for applications and entering in application information. The group of call agents consists of smart, higher education students who are employed by a temporary employment agency and their work at DLL is an additional job next to their studies. They are young, flexible, eager to learn and quick on their feet. The next group is the administrative staff. They handle all the mail, make sure all information of an application is brought together and ensure the process is compliant to the rules the organization has set herself and is obliged to be compliant with. They accurately redirect all internal available information to the correct application or person. The last group are the acceptance assessors. These are the people with the experience and knowledge about credit decisions. They are employed by DLL and there task is to evaluate each application which successfully reaches that point in the process. Their job is to assess the completeness of the information in the application, evaluate the risk of activating the application for the organization and discover fraud. Their work is an important contribution to the success and growth of Freo.

2.2.2. Flow of applications

To give a feeling of the process in which the above mentioned groups operate, a few common paths of applications are described in this section. A complete process model is included in App. A figure 1 and App. A figure 2. To illustrate the description a simple sequential, but incomplete, process model is created for a normal activated application in Figure 2. For the complete process model check the Appendix A. Always the first action in the process is the submitting of a credit request by the customer on the Freo website, as can be seen in Figure 2. The following step is an automated credit check by the credit decision engine in Advisor. Which, as mentioned, is the information system supporting the participants in the process. After the credit decision engine accepts the application a call agent will contact the customer to complement the application information. During this contact the desired offer is sent by the call agent to the customer. Then the process waits until the customer sends the offer and additional documents back. They are received and ordered by the administrative staff. The acceptation assessors approve the application and trigger the back office to activate the application and execute the initial payment. After this the additional documents of the customer are sent back.

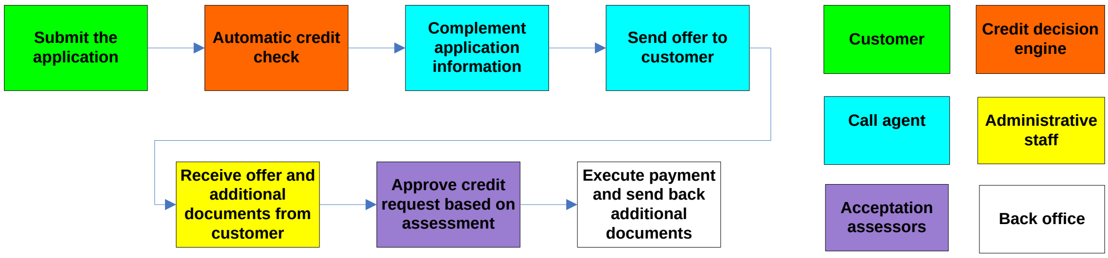


Figure 2: Simple process model of an activated application

Unfortunately not all applications follow this ideal and successful path to activation. The first automatic credit check may result in an acceptance by the credit decision engine. The application will then continue on the normal path. A decline during the automated credit check causes the system to automatically inform the customer that his application has been declined. The third possible result is a reference, this means the system is unable to uniquely identify the customer or misses vital information and one of the call agents has to asses the application before it can continue. This occurs frequently when the credit decision engine is down, or the customer has credit score synonyms, which means that multiple possible matches have been returned during the automatic credit check. It is evident that based on this information a decline is undesired, because the applicant may very well be a good customer.

2.2.3. Work item queues in the Advisor information system

The Advisor information system contains five queues from which work items can be picked. These queues are: handling leads, complement application information, call after send offer, assess application and call after incomplete application. These queues are discussed below in more detail since they are the representation of work items to the call agents.

2.2.4. Handling leads

A call agent can pick up and check the applications in this queue. The call agent assesses why the application could not be automatically checked and indicates to the system how to handle the application. When the second credit check accepts the yet unchecked application the following action is the same as when the first automated credit check accepted the application. All accepted applications will be automatically assigned to the queue for complementing the application.

2.2.5. Complementing application information

The customers in this queue need to be contacted by the call agents. In this first contact the call agents checks the information which was filled in by the customer on the website and asks for additional information that is important for the assessment of the application. In an ideal situation this is the only contact with the customer and all necessary and relevant information is gathered in one go. In a less perfect world, this is the longest contact with the customer and later shorter inquiries are needed to receive missing information. The first moment of contact is therefore very important, as it is the customers' first impression of the organization. The gathering of information during this contact may prevent later finding the application information to be incomplete and this conversation sets the expectations of the customer. This is therefore the most valuable and decisive contact moment with the customer in the application process. During this contact moment the call agent may find, based on the new and acquired information, to discontinue the application in the process by declining it. This is immediately communicated to the customer, who also receives a mail of rejection. When the information is complemented, the second credit check also accepts the application and the customer still has the desire for a consumer credit, the call agent will create a number of offers to send to the customer. The administrative staff will handle the printing, sorting and mailing of the offer and adds a list of additional documents which are necessary for the assessment by the acceptation assessors. At the same time, the application in Advisor is moved to the queue containing all the applications which have sent offers outstanding.

2.2.6. Call after sent offer

The tasks corresponding to this queue belong to the call agents. After a certain time the call agents will contact the customers and ask if they have received the offers in good order. They will also ask if and when the customer is to return the offer and requested documents. Of course this action is not performed when the customer sends back the offer and requested documents immediately.
When the postal item of the customer is received back, the administrative staff files it and sets the application to the queue belonging to the acceptation assessors.

2.2.7. Assess application

In this queue the application waits to be evaluated for approval. The acceptation assessor will pick up a work item and assesses the application. If all necessary documents are available and approved, the acceptation assessor will activate the applications, trigger the payment by the back office and let the administrative staff send a welcoming package with the original documents of the customer. The process is then finished. If new information or documents are required the application is marked as incomplete and is put in the corresponding queue.

2.2.8. Call after incomplete application

This queue is worked on by both the call agents and the acceptation assessors. The customer is contacted by either of them and is requested for the additional information or documents. The request is explained as to why this extra information is needed. After the request is fulfilled, the evaluation is completed and a decision is made. If the application is declined at this stage of the process the administrative staff will send a letter containing the reason for the decision and the process ends.

2.3. Key performance indicators

The objective of the method is to design recommendation logic that improves performance on key performance indicators (KPIs) of the operational service process. In his book [Parmenter, 2007] defines KPIs as: "KPIs represent a set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization." These KPIs can be of very different nature, whatever the organization sees as success will have influence on the definition of them. Often organizations define KPIs on four important dimensions. These dimensions are time, quality, costs and flexibility. As is explained in [Brand et al, 1995] these areas influence each other. A suggested process improvement will hopefully be an improvement in one dimension, but it is likely to decrease the value of another. Resulting in the devil's quadrangle in which the four dimensions have a certain value creating the quadrangle. They pull on each other when they are rearranged during process (re)design. The only way to improve the process as a whole is to increase the size of the quadrangle. The RecLog^2 method improves performance on KPIs and thereby focuses its improvement efforts to the dimensions that are related to the obtained KPIs. The formulation and definition of the KPIs is therefore an important part of the method. As a last remark, as mentioned previously, organizations need to identify what they regard to be success for them. Interaction with the organization of the operational process is thus necessary.

2.4. Recommendation systems

The early e-commerce recommendation systems have transformed to serious business tools to help the customers in their buying decisions [Schafer et al, 2001]. They provide significant added value to the sales efforts of the organizations that use them. These recommendation systems use hand-coded or mined knowledge about the customer to guide them to products that they will probably like. They use the inputs from customers and provide a recommendation to them in different manners. Techniques to provide these recommendations vary. In [Park et al, 2012] an overview is provided which data mining techniques are used in recommendation systems. Some examples are k-nearest neighbors, decision trees, clustering and association rules. Usually they group together somewhat similar entities with the assumption that if a certain entity is liked or fits, similar entities will also be liked or fit. This results in recommending entities that look alike to the entity that was liked. This manner of recommendation is used in sales and e-commerce and has contributed added value to its implementers. The organizations using this recommendation usually have a KPI set that focuses on sales. Recommendation systems work in those situations, in which an external entity receives a recommendation about an internal entity of the organization. The RecLog^2 method should also work on recommending an external entity to an internal entity. In specific for the Freo application process. The RecLog^2 method is to set up recommendation logic which provides the call agents (internal resource entities) with recommendations about which application to contact next (external entity). The dependencies and relations that can be used for the recommendation logic should influence the formulated KPIs. The method developed uses available operational information about previous entities and tries to discover dependencies between them that are used to define the recommendation logic in order to provide recommendations to the interacting entities.

2.5. Statistical analysis

The statistical analysis used in this master thesis is taken from [Montgomery et al, 2003]. In chapter 10-6.1 the definitions of the statistical test are provided. The tests are for inference on two population proportions. It considers the comparison of two binomial parameters. The binomial distribution, which is used for the binomial parameters, is a collection of Bernoulli experiments. The Bernoulli experiment is a simple experiment with only one observation. Either the experiment is a success (1) or the experiment is a failure (0). Thus the Bernoulli distribution relates to one specific observation. When more experiments are put together, the number of observations increases, leading to the binomial distribution. It is described by the number of observations and the chance of one observation being successful. To be able to model the KPIs as binomial parameters they must be defined in such a way that they have only two outcomes. Either the entity is regarded as a success or a failure for a certain KPI. This way of defining the KPIs allows to model them as binomial parameters. To test if there are dependencies or relations between the characteristics of the entities, subsets of the dataset are compared with each other. These two independent random samples have a sample size and a corresponding number of successes within the sample. These samples are compared when the normal approximation for the binomial parameter can be applied. This is when the sample size is at least 30. Assuming that the two parameters that are to be compared are the same, one can test the hypothesis (H_0) that they are the same (see Figure 3). When this hypothesis is then significantly rejected the conclusion must be drawn that there is a difference between the parameters. When this is the case at least one of the three alternative hypothesis (H_1, Figure 3) must be valid. This indicates that there is a difference between the two samples. When the samples are chosen based on certain entity characteristics the test can identify valid relations between the characteristics. The assumptions which are used to conduct the test are mentioned above, however to provide a clear overview Table 1 is created, that contains all the assumptions.

Nr	Assumption	Met because
1	The KPIs are binomial distributed	A collection of observations is used and an single observation has either a success of failure outcome with regards to the KPI
2	The sample size allows for a normal approximation of the parameters	All sample sizes should be equal or greater then 30, else no analysis is performed
3	The parameters are equal	This is assumed in the hypothesis (H_0) that is tested and results in a strong statement when rejected

Table 1: Overview statistical assumptions

As mentioned above the statistical test is to check inference on two population proportions. The calculus looks like this. Select two samples based on entity characteristics and denote the two binomial parameters of the samples 1 and 2, as p_1 and p_2 respectively. The sample sizes (number of observations) are given by n_1 and n_2 and let X_1 and X_2 be the observations with the desired or successful result. Two hypotheses are provided in the test of which the first is listed as the third assumption. H_0 is the hypothesis that there is no difference between the parameters. This hypothesis is assumed and tested because the search is for a valid dependency between characteristics of entities and if the H_0 hypotheses is not rejected, one must assume that there is no dependency between the characteristics of the samples. Therefore the assumption of equal parameters is contradictory to the desired outcome of the test, making the rejection of it a strong statement. This is a very strong indication that there actually exists a difference between the two parameters and there is probably a dependency between the characteristics of the samples. This indication is much stronger than being unable to reject a desired hypotheses that there exists a difference. There are three possible alternative hypotheses H_1 that are considered as can be seen in Figure 3. The first alternative hypotheses indicates that there is a difference between the parameters, but does not assume any direction for the difference. The second and third alternative hypotheses do so. The second indicates that there is a difference and that p_1 is bigger than p_2. For the third it is the other way around, there is a difference and p_1 is smaller than p_2.

$H_0 : p_1 = p_2$

$H_1 : p_1 \neq p_2 \qquad z_0 > z_{\alpha/2} \quad or \quad z_0 < -z_{\alpha/2}$

$H_1 : p_1 > p_2 \qquad z_0 > z_{\alpha}$

$H_1 : p_1 < p_2 \qquad z_0 < -z_{\alpha}$

Figure 3: H_0 and H_1 hypotheses and rejection criteria

Using the normal approximation for p_1 and p_2, which is allowed when assumption two is met, the test statistic becomes the equation in Figure 4: Test statistic for H_0: $p_1 = p_2$. This equation is extracted from [Montgomery et al, 2003] page 362 and is used for the statistical validation of the dependencies in the RecLog[2] method. The Z_0 value that is calculated with the test statistic is compared to the rejection criteria, which are depicted next to the corresponding alternative hypotheses in Figure 3. For all statistical tests in this work an α is chosen of 0.05. This means that the confidence interval used in the test is 95%. The rejection values for hypothesis H_0 then become: $z_{0.025} = 1.960$, $-z_{0.025} = -1.960$, $z_{0.5} = 1.645$ and $-z_{0.05} = -1.645$.
All comparisons in this thesis are done against all the alternative hypotheses. Of course not all the alternative hypotheses can be accepted. For the directive alternative hypotheses either of the two can be accepted. As soon as one of the three alternative hypotheses is accepted, the H_0 hypothesis is rejected. The directive alternative hypothesis will indicated what kind of dependency exists between p_1 and p_2. This allows to evaluate which sample with certain characteristics performs better then the other. When the alternative hypothesis $p_1 \neq p_2$ is also accepted there is a even more powerful difference between the two samples than when only one alternative hypothesis is valid.

$$Z_0 = \frac{\left(\frac{X_1}{n_1}\right) - \left(\frac{X_2}{n_2}\right)}{\sqrt{\left(\left(\frac{X_1 + X_2}{n_1 + n_2}\right) \cdot \left(1 - \left(\frac{X_1 + X_2}{n_1 + n_2}\right)\right) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)}}$$

Figure 4: Test statistic for H_0: $p_1 = p_2$

3. The RecLog^2 method

The content of this chapter describes the actual method that is developed. First the RecLog^2 method is presented. The name RecLog^2 is obtained from the abbreviations of elements within the method. RecLog is short for recommendation logic. The square is added because the recommendation logic is defined with the use of historical data. A representation of this data may be in to form of a log, therefore the square at the end. The phases of the RecLog^2 method are then described in two separate sections. In these sections the individual steps within the phases are presented and explained.

3.1. The RecLog^2 method

The RecLog^2 method is visualized in Figure 5. As can be seen in Figure 5 the RecLog^2 method consists of two phases: the defining and the implementation phase. This distinction is made because there are two important parts in the method. The first part in which all possible dependencies and relations between entity characteristics and the process KPIs are defined to testable hypotheses. The second part in which the hypotheses are actually tested and validated, after which the recommendation logic is selected, defined and implemented.

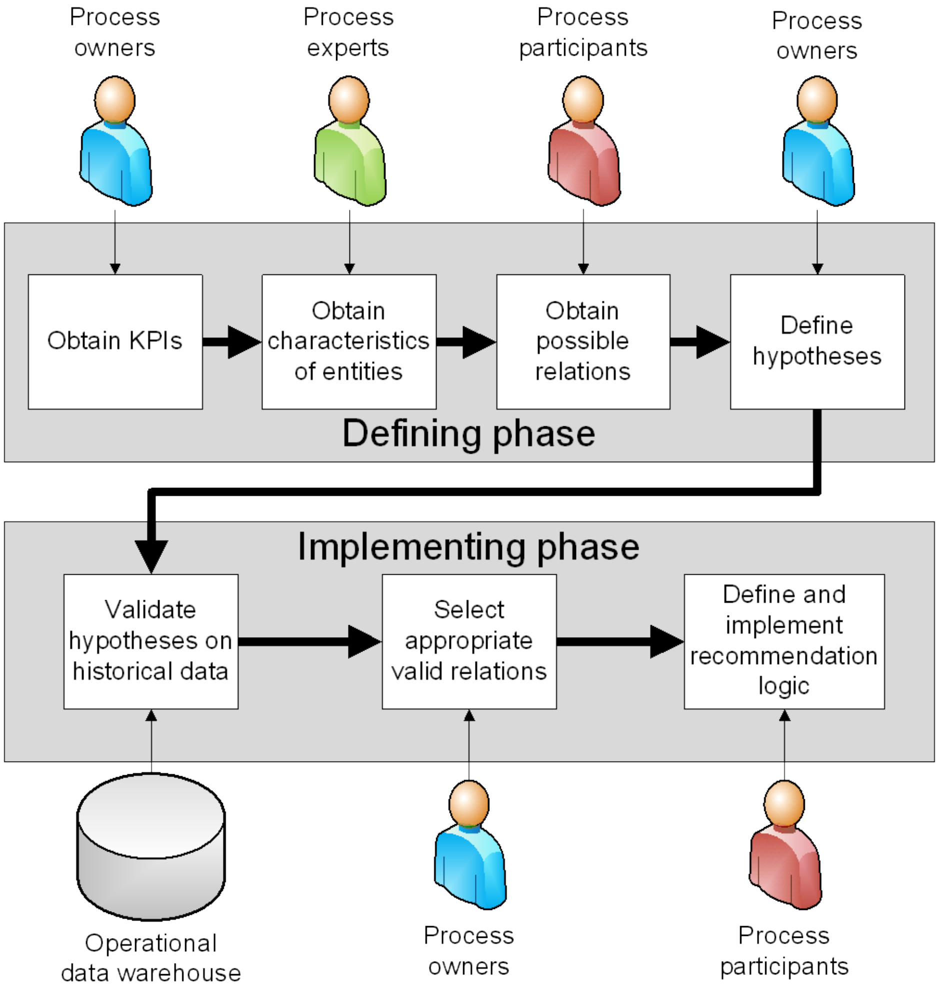


Figure 5: The RecLog^2 method

In the method model also the most important party to be involved in each step is linked to that step. This does not mean other sources or parties are not used or consulted in that step. Only the most important party is linked to the step with a thin arrow in Figure 5. This party is the most important, because of varies reasons. It may have valuable insights, have hands-on experience with entities or it may have the authority needed to formulate a decision within the process. In the remainder of this chapter the phases are explained including their internal steps necessary to produce recommendation logic for process improvement on its KPIs.

3.2. Defining phase

In this phase of the RecLog^2 method the goal is to define the hypotheses that are to be tested in the implementation phase. However, to be able to define the hypotheses clearly first three other elements need to be obtained. These are the relevant KPIs, the characteristics of the entities and the possible dependencies or relations between them. Therefore this phase is divided into four steps, as can be seen in Figure 6. The following four sections explain the activities in these steps in more detail.

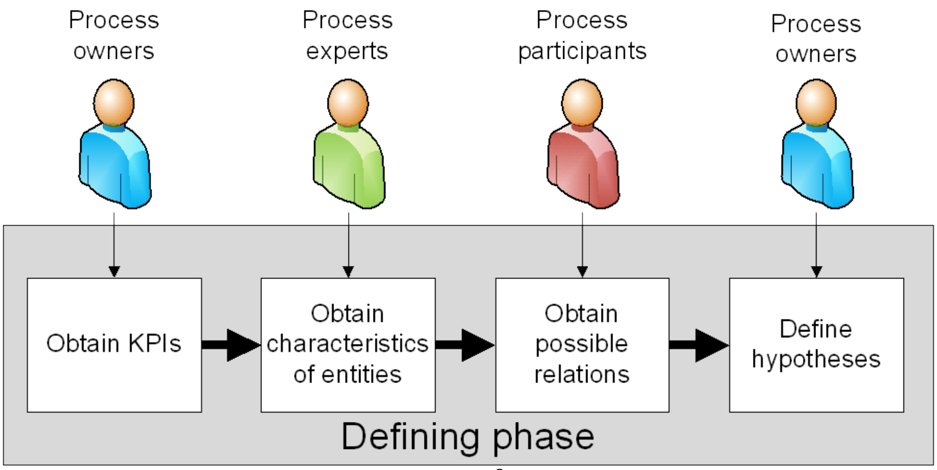


Figure 6: Defining phase of the RecLog^2 method

3.2.1. Obtain KPIs

To be able to measure the degree of performance of a designed or proposed optimization it should be clear what KPIs there are. The first question that needs to be answered is: What are the relevant KPIs of the process? This information needs to be available even before performing a zero measurement, to which the results after the implementation can be compared. Besides this necessity, insight in the KPIs provides at least a direction to find possible important and valuable elements in the process, which may point to interesting characteristics. It is also vital that the selected KPIs can be influenced by changing entities' behavior by providing recommendations. When no dependency exists, the recommendation logic is unable to improve performance. The first step in the RecLog^2 method is therefore concerned with the KPIs of the process. Usually the process owners have at least a general idea of what they think the KPIs are and what they would like to see as the KPIs. This general idea needs to be clarified and defined to the point that the KPIs are unbiased and measurable. If process owners are unable to produce any KPI, one may help them identifying important elements of the process in the four dimensions of time, quality, costs and flexibility. Together with the process owners one must reach agreement about the entities that are relevant for the KPIs and the KPIs themselves. Usually the KPIs are created by a collection of characteristics of entities. The KPIs are those characteristics the process owners manage the process on. They are the most important output of the process and these are the characteristics which the process owners would like to improve. The KPIs will provide the method with its first relevant entities. As can be seen the required information is highly dependent on the opinion of the process management. Their involvement and commitment to this method is necessary to generate any meaningful result. Appropriate tools to obtain this information are interviews and brainstorm sessions with the process owners. During the interviews individual opinions about the KPIs are investigated and a global feel for the process is obtained. When available one can specify and identify any missing information with regard to the KPIs using the process model. When this step is completed the result will be a list of the most important KPIs.

3.2.2. Obtain characteristics of entities

The KPIs from the previous step already point to one relevant entity within the process. However the recommendation logic is to recommend at least one entity to another one. So one must answer the question what the relevant characteristics of entities interacting with each other within the operational service process are. Therefore the entities that interact with the KPI entities are also considered. To be able to do so these entities need to be identified and collected. The recommendation logic is designed for a given operational service process. The entities which need to be matched within the service can therefore be identified within the process. When all relevant entities are collected their characteristics should be mapped. The characteristics of the entities are required to be able to specify hypotheses on them in the next step. These characteristics can be any kind of information about the relevant entities. There are no limitations on the possible characteristics that may be included at this point in time. The more characteristics are included, the more possible dependencies and relations can be analyzed and the bigger the chance of finding suitable recommendation logic. To extract all interesting and relevant characteristics, the involvement of process experts is helpful. Again interviews may be hold to obtain the necessary information. The process experts are persons that have a deep understanding of the process and have a feeling for the interaction between characteristics. When a certain unavailable characteristic is identified, that could be of high influence on the entities and their dependencies with the KPIs, this characteristic must be made available by implementing new logging requirements to the datawarehouse. The result when this step is taken successfully is a list of the characteristics per entity involved, together with the list of the KPIs it forms the input for the following step described in the next section. The insights until now should be put together for a brainstorm session, this to facilitate the process owners during the session and to provide as much clarity as possible about the characteristics and KPIs.

3.2.3. Obtain possible relations

The third step in the defining phase is to obtain the possible dependencies and relations between entities. At this stage, the only restrictions on the possible dependencies and relations are the lists with characteristics of the entities. This is the reason why as many characteristics as possible are included in the previous step. These dependencies can be of very different nature. They can be formulated based on beliefs of the process participants, the beliefs of one of the relevant entities under consideration, events and dependencies which occurred in specific cases in the past, general beliefs, customer suggestions and so on. These dependencies and relations again need to be gathered during interviews and brainstorm sessions. Of which the first may very well be conducted during the brainstorm session in which clarity is provided to the process owners about the characteristics and KPIs. Of course the other parties and especially the process participants should also be consulted. During these sessions all ideas and thoughts are appreciated, however keep in mind that the dependencies must relate to the characteristics of the relevant entities and that they must at least be expected to have influence on the KPIs. It is also important that the dependency can be influenced or managed to improve performance. When the list of possible dependencies and relations is long enough, the hypotheses to test them must be specified. This is done in the last step of the phase.

3.2.4. Define hypotheses

The hypotheses that are defined in this step will later be analyzed using the statistical analysis from section 2.5. So how to formulate correct hypotheses for the possible and/or expected dependencies or relations? The hypotheses themselves are the same for all comparisons and are given in Figure 3 on page 15, but the characteristics of the samples vary. Therefore the first action is defining the possible dependency or relation in terms of the interacting characteristics and identifying which KPIs may be influenced by it. It is possible that one relation is expected to influence one or more of the KPIs. In this case an individual test is created for every KPI separately. As mentioned before, the difference in characteristics of the samples that are to be compared should be under the influence of the organization. The characteristics of the external entities over which the operational service process has no influence should be kept constant to reach a valid conclusion with the statistical analysis. The most important activity in this step is thus the selection of the entity characteristics in the samples that are to be compared. The hypotheses are then specified stating in H_0 that there is no difference in the mean of the samples. As this is assumed in the test, the rejection of this hypothesis will indicate that there is a significant difference and that therefore the underlying dependency or relation is valid. The underlying dependency or relation is formed by the entity characteristics of the samples compared. In order to assess which recommendation rules are considered by the process owners, this must explicitly be asked during a number of interviews with them. They may at this stage reflect on what kind of recommendation logic can be implemented. This should not limit the size of the set, but if there are any expected dependencies tested that are deemed inappropriate or even dangerous to the organization the process owners can indicate this and extra caution is then taken to prevent undesired consequences. The list with the samples and corresponding hypotheses that need to be tested serves as input for the next phase in the RecLog^2 method.

3.3. Implementing phase

This chapter describes the steps in the second phase of the RecLog^2 method, as illustrated in Figure 7. The three steps in this phase are elaborated in the next sections.

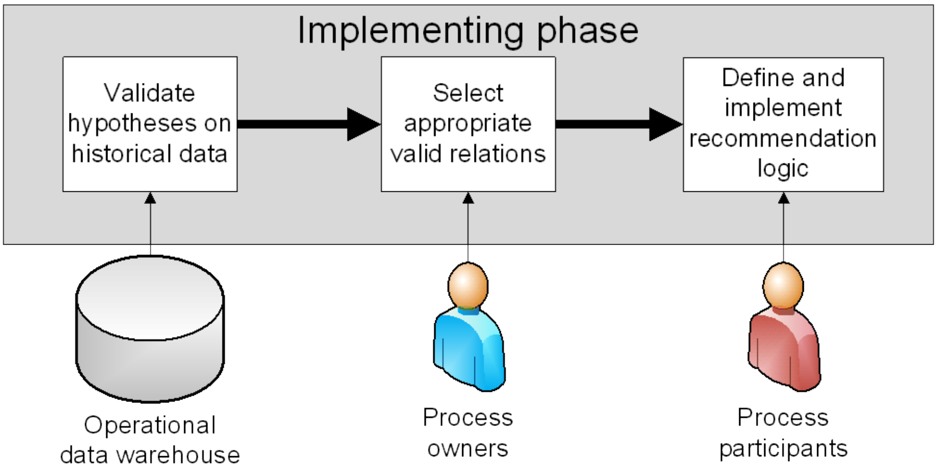


Figure 7: Implementing phase of the RecLog^2 method

3.3.1. Validate hypotheses on historical data

The list of samples and hypotheses from the previous phase results in pairs of binomial parameters, which can be validated on historical data. The pairs of binomial parameters correspond to the pairs of samples defined for the hypotheses. To be able to validate the hypotheses historical data is required to serve as observations for the statistical analysis. The extraction of the data to serve as observations is not an easy task. This is however a crucial part of the RecLog^2 method. This is also the reason why the method is created for operational processes. Nowadays most operational processes have data warehouses full of logs containing service actions or transactions of entities within the process. This rich source of information is both a requirement for the method and a curse for the implementers of it. The information contained in the datawarehouse is virtually never in the desired form, so it can easily be translated into the observations needed to test the hypotheses. When it is not possible to extract the required data yourself the help of datawarehouse and process experts may prove to be extremely valuable. However first effort must be spent to define exactly the desired data of the relevant entities in the process. Small data transformations are possible after the dataset is extracted, but any missing characteristics or other information will lead to extra iterations with the experts, which if possible should be avoided. When the dataset with all the observations of the entities' interactions is collected every hypotheses from the first phase needs to be tested as is described in section 2.5. Usually the hypotheses are not proven and only intuitive. The possible dependencies underlying the hypotheses are thought to be important in the interaction between the entities. However this is not always the case. Therefore the hypotheses formulated need to be examined and validated. The hypotheses can be investigated with the use of the historic operational data from the datawarehouse. If some characteristics are not present in the data warehouses this information needs to be stored and after a certain period the hypothesis must be assessed as either valid or invalid. This assessment is done using the statistical analysis (section 2.5) on the available data treating them as observations. When the

hypothesis is invalid, there is no evidence that the relation exists and therefore it should not be included in the recommendation logic. When a valid hypothesis is found, its influence on the KPIs must be determined. This influence can be either positive or negative and either significant or strongly significant. This last notion of strongly significant is when also the alternative hypothesis $p_1 \neq p_2$ is accepted. The significant influence is always presence when the H_0 hypothesis is rejected.
All valid dependencies and relations that are proven during the statistical analyses are listed as possible recommendation logic. In the following step of the method a refinement of these dependencies and relations is performed using only the ones that are valid.

3.3.2. Select appropriate and valid relations

The second step in this phase selects the appropriate and valid relations for the recommendation logic, but which of the valid relations need to be in this selection? The recommendation logic needs to be built up using the dependencies and relations from the valid hypotheses, because setting up recommendation logic on non existing relationships is futile. The overview of significant relations from the previous step in the method only looks at the valid individual relations. At this point a selection has to be made as to what relations will be incorporated in the recommendation logic. This again requires the involvement of the process owners. They may find that some found relations may not be used for the recommendation logic because of possible negative consequences either for the entities or for the organization. Needless to say these relations should not be included in the recommendation logic. Conflicting relations should also not be included in the recommendation logic, because of the obvious reason you want to provide a clear recommendation for the entities and not create more confusion between them. Also the degree of power of the relation may be of influence on the choice whether or not to include it. From the results of the statistical tests an indication of the degree of power of the relation can be extracted. Every relation that had its H_0 hypothesis rejected is significant and is a good candidate to use and as explained earlier when the H_1 hypothesis $p_1 \neq p_2$ may also be accepted the relation is even stronger then when only one of the other two alternative hypotheses is accepted, as the rejection criteria is more strict. The direction of the relation is extracted by checking which of the alternative hypotheses was accepted in the statistical analysis. Of course the characteristics of the better performing sample should be in the recommendation logic. Another issue are the limitations imposed on the implementation by the process itself or maybe even by the organization to which the process belongs. A carte blanche is seldom given to the implementers of the recommendation logic. Therefore the possibilities to implement recommendation logic need to be investigated. The impact of implementing individual recommendation logic rules needs to be explored in order to allow the process owners to weigh the investment against the possible return. As final activity in this step consensus needs to be reached with the process owners about what set of relations is to be used in the definition of the recommendation logic. This set is the input for the final step in the method.

3.3.3. Define and implement recommendation logic

The last step in the $RecLog^2$ method deals with how to define the actual recommendation logic and how to implement it. The set of relations from the previous step is first defined in clear and understandable recommendation rules. This is done by describing which characteristics of which entities interact in the relations and describing the direction that the recommendation rule should assign. What should be considered at this stage is how to provide the entities with the recommendation. Of course this issue was also part during the impact analysis, but now the physical implementation is considered. There are numerous ways to present entities with a recommendation. Insight must be achieved in how this is to be done. The last areas that need to be addressed are when to provide the entity with a recommendation during the process execution and how to enforce it. There are several options, as for instance on request allowing the entities more independence for the supporting system or always and mandatory, making the recommended work item the entities' next work item. All matters above must be considered and during a meeting with the process owners the best implementation within the limits of the process and the presentation of the recommendation to the entities must be agreed upon. When this is done activities are taken to implement the created recommendation logic in the operational service process. Thereby concluding the $RecLog^2$ method developed in this master thesis.

4. Defining phase

This chapter and the next chapter each describe a phase during the implementation of the RecLog^2 method in the Freo application process at DLL. The defining phase of the method is again presented in Figure 8. Each step in this phase corresponds to a section in this chapter. The sections contain the activities performed for the implementation of the RecLog^2 method.

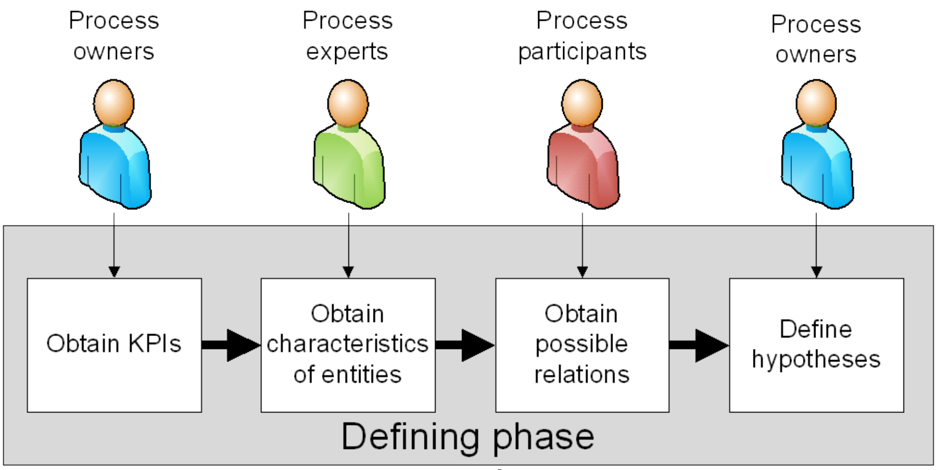


Figure 8: Defining phase of the RecLog^2 method

The RecLog^2 method is implemented with the project scope defined by the call agents, the applications, the key performance indicators and the dependencies and relations between these entities. The call agents are the agents working within the process, contacting customers to complete applications, answer questions and check with the progress of outstanding offers. The scope for the applications is limited to the number of applications that get through the first automatic (credit)check and enter the process in which the call agents actually need to perform tasks on them. For a more detailed description of the process see section 2.2 and in App. A figure 1: Freo application process model (part 1/2) the scope boundary is visualized by a red line with arrows beneath it to indicate which part of the model is in scope.

4.1. Obtain KPIs

The process owners of Freo indicated during the interviews that there are four KPIs. These are the factors on which they would like to manage and improve the process. Below the four KPIs are given in Table 2. They are explained in greater detail in the remainder of this section.

Nr	Abbreviation	Key performance indicator
1	ACT	The number of activated applications
2	GAP	The number of applications which gave a smaller gap then initially
3	OTH	The number of applications which contains existing credits of other credit providers
4	CRC	The number of applications which are in a €1,000.- range of a higher credit class and are taken there

Table 2: Overview key performance indicators

The KPIs here are defined to the number of applications with certain properties. This number is however dependent on the scope of the time period considered. The number of activated applications will only increase in time, which is a logical consequence of the continued work within the process. To limit this time dependency the KPIs used for the remainder of this master thesis are represented as percentages. Under the assumption that the total number of applications coming in is relatively constant or growing, the process performance is improved when the percentages of the KPIs are improved. Therefore the KPIs are explained and assumed to be percentages from here one.

- The percentage of activated applications

The first KPI is derived from the core business. The percentage of activated applications is an indication of the success of the brand, it increases the revenue of the portfolio and with more activated applications there is an automatic risk diversification. This is the main reason why the percentage of activated applications is more important than the height of the amount of the application. This percentage is relative to the total number of applications within the scope as it is defined earlier.

- The percentage of times the final gap between the maximum loan capacity and the principal amount is smaller than the initial gap.

The second KPI focuses on the height of the principal amount. The principal amount is the agreed amount of credit that is provided to the customer. This KPI is designed to maximize the generated business on the applications that are already coming in. When the initial gap between the maximum loan capacity and the first requested amount is larger than the gap between the maximum loan capacity and the final principal amount, effort has been spent to increase the business of that customer. This is observed as the difference between the gaps in the initial request and the final agreement. When this gap has become smaller this is interpreted as a success, otherwise it is regarded as a failed attempt. This small investment of asking if the customer would consider an acceptable higher credit, increases the portfolio growth and prevents customers from asking for a credit increase in the first period of the contract, as they have already got extra freedom within their credit. This percentage is relative to the total number of applications within the scope of the project that have an initial gap.

- The percentage of times existing credits of other credit providers are included in the new contract

The third KPI is some what similar to the second. Now the focus is not on the still available open loan capacity, but on the currently used loan capacity with other credit providers. It can be seen as a service to the customer, including credits from other credit providers, creating a clear overview of the customers outstanding credits all in the care of one credit provider. It is also very well possible that by doing so the customer reaches a higher credit class in which a lower interest rate is charged. For Freo this is also an interesting way to grow the portfolio. Besides that benefit, the extra information about the progress of the credits enables DLL to construct a more complete and reliable customer risk profile, which creates more transparency about the risk DLL is facing. This percentage is relative to the total number of applications within the scope of the project that have existing credits with other credit providers.

- The percentage of times a credit request which is only € 1,000.- or less from a higher credit class is carried to the next credit class

The fourth and last KPI is also a service to the customer and it has overlap with both the second and third KPI. When the initial requested amount for the credit is within a € 1,000.- range of a higher credit class, in which the interest rate charged is lower, the customer is advised to increase the credit to be eligible for this lower interest rate. Of course the maximum loan capacity must be sufficient to allow for this increase. The action is considered successful when the final principal amount is in the higher credit class. This KPI is also relative to the total number of applications within the scope of the project that fulfill the requirement of being in the € 1,000.- range of the higher credit class, which is allowed by the maximum loan capacity.

The KPIs need to be defined in such a way that they have two possible outcomes. Either the application is regarded as successful or as a failure with respect to a KPI. This is an important characteristic for the distribution that is used during the statistical analysis as described in section 2.5 Statistical analysis. The outcomes of the KPIs are defined in Table 3. The distribution that can be used to model these KPI variables is the binomial distribution. The number of applications which have characteristics that resemble the predefined requirements for the KPI are interpreted as observations and the success outcomes are modeled as the desired result. These observations and success occurrences are used to compare different selections of the dataset to uncover the possible relationships that interact with the applications.

Nr KPI	Success outcome (1)	Failure outcome (0)
1	When application has state activated	When application has another state as activated
2	When the gap between the maximum loan capacity and the principal amount is smaller than the gap between the maximum loan capacity and the initial requested amount	When there is no gap, the gap is the same size or the gap is greater
3	When any existing loan is included in the application	When no existing loan is included in the application
4	When the application is carried to the higher credit class	When the application is not carried to the higher credit class

Table 3: Overview outcomes KPIs

4.2. Obtain characteristics of entities

The KPIs already point to one relevant entity: the Freo applications.

4.2.1. Characteristics Freo applications

The set of available characteristics of the Freo applications is very large. A lot of data is logged in the datawarehouse. It is hidden within a large number of tables each containing some specific parts of information. Fortunately a lot of information is available right from the start of the process, because the form customers fill in on the website is a rich source of application characteristics. Also the automatic credit check provides information about the application. All these characteristics are taken into account during the implementation of the RecLog[2] method. During interviews with the process experts a complete overview of the Freo application characteristics is created. It is adopted in App. A table 2. A selection from these characteristics will be made in the next step of the method.

4.2.2. Characteristics Freo call agents

The available characteristics of the call agents were initially very limited. Only the name, employee number and registration date of the account were available. The process owners indicated that they would also like to see dependencies on the performance characteristics of the call agents. This information was not readily available. Fortunately with the suggested change from the earlier project extra information about the actual performed tasks is logged. This new table contains data from which one can calculate on which queue the call agent works and to what extent this call agent is successful in progressing the application to the subsequent queue. This extra information also can reveal which call agent has made the complement application call. This is regarded as the most important contact moment with the customer. Next to the characteristics mentioned above an extra set of characteristics was gathered by requesting for them in a letter. In the letter permission was asked to use the data from the datawarehouse. Also some extra characteristics were requested to be added in the analysis. These characteristics are:

- The months of experience as a DLL call agent
- The months of experience as a call agent (not with DLL)
- The average number of work hours per month
- The call agents nationality
- The call agents place of residence
- The preferred work item queue to work on
- The foreign languages the call agent speaks

These characteristics are included because an early possible relation brainstorm session indicated there might be a dependency with the characteristics of the applications. The complete list of call agent characteristics is presented in App. A table 1.

4.3. Obtain possible relations

The dependencies and relations between the characteristics and the KPIs are discussed with the process participants during interviews and brainstorm sessions. They indicated eight intuitive possible relations. The possible dependencies are given below, with their abbreviations between parentheses.

- Gender relation (Gender)
- Nationality relation (Nationality)
- BKR – credit score complexity relation (Comp A)
- Gap complexity relation (Comp B)
- Higher credit class complexity relation (Comp C)
- Part of the day registration relation (Part)
- Queue relations (Queue)
- Account manager relation (Account)

In the remainder of this section the different expected relations are explained in more detail, before defining the hypotheses in the last step of the defining phase.

1. Gender relation

The idea exists that there is a relation between the gender of the applicant and the gender of the call agent who handles the complement application call. Some of the process owners believe that men prefer a female call agent and female prefer a male call agent. Others have it the other way around. They believe male applications will not attribute experience and authority to the female call agents and therefore are better serviced by the male colleagues. Intuitively there is a possible relation between these characteristics, but the direction and the gravity of the relation need to be investigated in the statistical analysis.

2. Nationality relation

Although Freo focuses primarily on the Dutch consumer finance market, there are also foreign customers in the Freo portfolio. With them the second possible relation is identified. It is possible that customers with a different nationality are less proficient in the Dutch language. The process owners believe it is considered a great service to the customer to be flexible with the language in which the calls are made. This service may therefore be of influence on the KPIs. The relation between the native language of the customer and the possibility to have the call agent communicate in it, may have a positive influence on the KPIs.

3. BKR – credit score complexity relation

The next three relations are linked with each other. They all consider that the more complex cases need to be handled by a more experienced call agent in order to get the KPIs values higher. The characteristic of importance for this relation is the experience of the call agent. This is captured with the month of starting work as DLL call agent for Freo. This characteristic is compared with the characteristic of the application. For the BKR – credit score relation complexity is defined as the fact that the customer has other open credit amounts with other credit providers and when the customers' credit scores is in a certain range. Because the credit score can result in a multiple of results a binary distinction is made. The credit decision engine within the supportive information systems returns a positive integer as the credit score for the customer. To make is a binary characteristics the range for distinction is based on a meeting with one of the most experienced process participants and a separate meeting with a risk specialist. Both were asked what range of credit score results indicates a higher than average complexity. Both came up with the same range. The credit score result is a positive integer number which can be even bigger that 1.000. DLL guidelines indicate any customer with a credit score higher or equal to 704 is to be considered a customer with a sufficient and good credit score. Any credit score of 704 or higher, is therefore considered not to have extra complexity. This guideline is not enforced to the extent that any customer with a credit score below 704 is declined. Just under this threshold value of 704 there is a range of credit score results which are considered a gray area. This is the area in which the guideline indicates the customer should be declined, but with a lot of these customers there still are possibilities for a Freo credit. Because these possibilities have to be identified and explained to the customer by the call agent, these applications are regarded as more complex. The range of this gray area is given by both experts from a credit score of 600 to the desired result of 704. In this range the applicants often still have options to consider and there are possibilities to agree upon a credit. However, the options and possibilities need to be suggested to the customer. This is a task for the more experienced call agent who is able to see the options and possibilities within the boundaries of the organizational strategy and to explain these options to the customer in a convincing way, in order to be able to accept the customer in stead of loosing his business. To be complete, any application with a credit score below 600 is declined. There are almost no exceptions to this rule which results in a very simple process for these applications. The possible relation is therefore between the experience of the call agent and the degree of complexity of the application.

4. Gap complexity relation

As mentioned before, this relation is somewhat similar to the previous. The characteristics of the call agent remain the same, but now another complexity issue of the application is considered. Now the existence of a gap between the desired amount of credit and the maximum loan capacity of the customer is taken into account. The existence of such a gap provides DLL with the opportunity to increase business by informing the customers of this gap and explaining the benefits to increase the desired amount of credit at this point. This prevents customers asking for an increase soon after the credit has been approved, increases the portfolio of DLL and provides extra freedom for the customer within his credit. The idea again is that a more experience call agent is more capable of explaining the benefits and getting the customer to agree.

5. Higher credit class complexity relation

The last complexity relation looks at an application with very specific requirements. The credit amounts can be classified into different credit classes, for the Freo application process the credit classes are from € 5,000.- until € 10,000.-, from € 10,000.- until € 15,000.-, from € 15,000.- until € 25,000.- and from € 25,000.- until € 50,000.-. These classes have different interest rates and other features. For the customer it can be very interesting and even cheaper to be in a higher credit class by increasing the credit amount. The higher the credit class the lower the interest rate for the customer. This is a given for the process as the organization only charges interest and with a higher credit the fixed costs for the application are spread out over a higher amount. The relation looks specifically at the applications that are € 1,000.00 or less from such a credit class boundary and the maximum loan capacity of the customer allows to provide a credit in the higher credit class. The maximum loan capacity is the maximum amount the customer may get as a credit. The process owners are convinced that applications fitting the above mentioned description should always be carried to the higher credit class. This is regarded as a valuable service to the customer whose credit becomes cheaper with the lower interest rate and DLL benefits with the happy customer and a higher portfolio.

6. Part of the day registration relation

This relation considers the best suited moment to contact the customer for the complementing application call. It takes the characteristics of the moment of registration of the application and the moment the call agent is able to complement the application. There is the notion by the process owners that call agents are more successful in contacting the customers in the same part of the day as was the registration of the application. This notion is already implemented in the process as the management urges the call agents to take action on newly registered applications within the first hour. Therefore this will usually be in the same part of the day as the registration. However the direction of this relation needs to be investigated with the statistical analysis as it has not been proven by data before.

7. Queue relation

This relation takes a different approach. The queue in which the applications wait to be complemented is available to the call agents and information about their performance in this queue is made available. A possible relation exists between the success of the call agent in the complement application queue and the outcome of the application. It is expected that the higher the success percentage of the call agent the higher the results on the KPIs are. This also needs to be investigated during the statistical analysis.

8. Account manager relation

The last relation is based on a hunch that by having the same call agent that successfully complemented the application call the customer again when more contact moments are required. That specific call agent was part of the first conversation with the customer and is therefore probably more knowledgeable about the customer than other call agents. This can mean that less time is needed to read up on the application and to gather any missing information. Besides the increased efficiency the customer may adopt the idea that the call agent is his personal account manager, which in turn can influence his customer satisfaction.

4.4. Define hypotheses

The description of the hypotheses until now is not unambiguous and clear. In Table 4 the characteristics of the hypotheses as they were analyzed and defined are presented. For every hypothesis both the call agent and the application characteristics that are considered in the hypotheses are provided. These hypotheses are analyzed further specified in the following sections.

Nr	Hypotheses	Call agent characteristics	Application characteristics
1	Gender	Gender	Gender
2	Nationality	Nationality Foreign language	Nationality
3	BKR – credit score complexity	Number of months of experience as DLL call agent	The presence of existing credits and/or a credit score within the 'grey' area (600 – 704)
4	Gap complexity	Number of months of experience as DLL call agent	The presence of a gap between the maximum loan capacity and the initial requested amount
5	Higher credit class complexity	Number of months of experience as DLL call agent	The requested amount is within a € 1,000.- range of a higher credit class and the maximum loan capacity allows to enter the higher credit class
6	Part of the day	Part of the day that the complementing application call is performed (morning, midday, evening)	Part of the day that the application is submitted by the customer (morning, midday, evening, night)
7	Queue	Success percentage in complementing application queue	The presence of the application in the complementing application queue
8	Account manager	Which applications have been complemented	Percentage of the total number of notes created by the call agent for the application

Table 4: Overview hypotheses' characteristics

4.4.1. Gender hypotheses

The hypotheses corresponding to the gender relation state that there is a difference in performance when comparing the four different gender pairs that can be made with the call agents' and applicants' gender as depicted in Table 5. Each line in this table is analyzed against the four KPIs. This table, as well as the similar tables in the following sections is interpreted as follows. On the left side the characteristic of the application or the applicant is defined with its possible values. On top the characteristic of the call agent is presented. The table then shows all possible matches between the two characteristics and their values. The numbers in the cells correspond therefore to a possible match between the characteristics. For instance the number 1 in Table 5 corresponds to the applications in which a female applicant has received a complementing call by a female call agent. The number 2 is for a female applicant who received a complementing call by a male call agent.

Gender		Call agent	
		Female	Male
Primary applicant	Female	1	2
	Male	3	4

Table 5: Gender hypotheses possible matches

The rest of the numbers follow the same line of reasoning, just follow the column and row and you will know the possible match that goes with the number. The numbers are chosen and further have no meaning. The characteristics of the call agent were set in the columns of the table, because this is the entity characteristic on which the process owners have influence within the operational service process. The characteristics of the rows are considered given and cannot be changed. This causes that the samples to compare with each other are on the same row. For the numbers 1 and 2 the statistical analysis will reveal if female applicants should be recommended to female of male call agents, whichever performs best on the KPIs. What is also possible is that the H_0 hypothesis, no difference in performance, is not rejected. In that case there is no need to design a recommendation rule for it, as it seems, the choice for a female or male call agents does not have any influence.

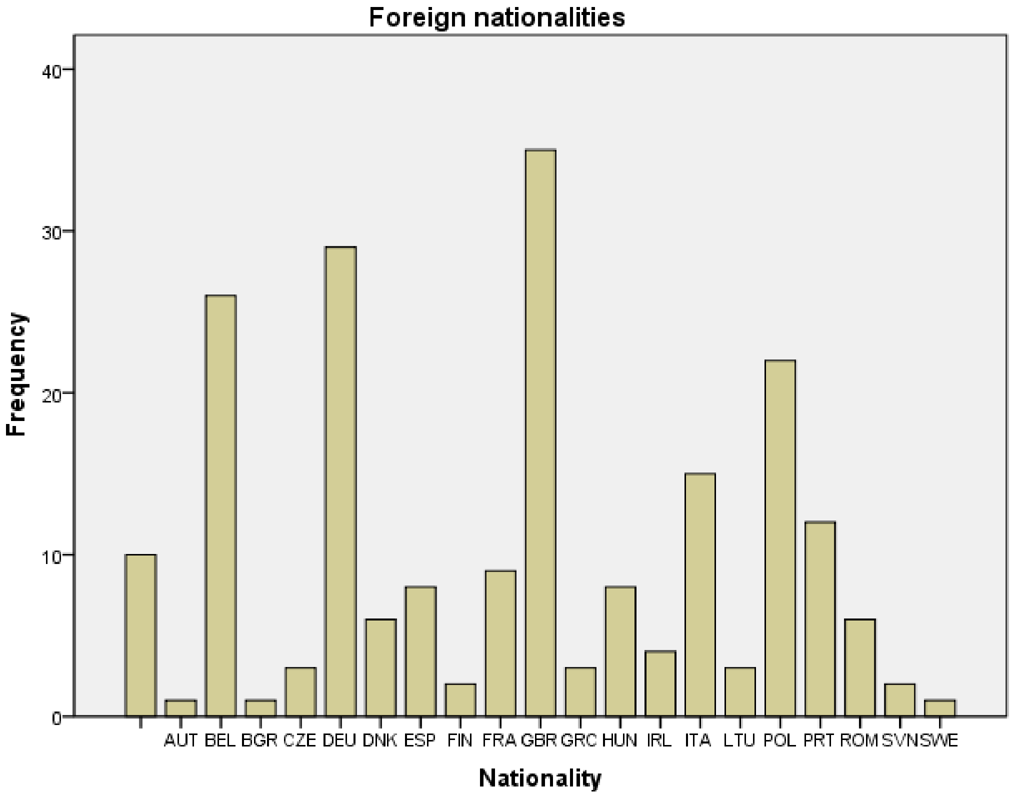


Figure 9: Foreign nationalities in the dataset containing 17.338 applications

4.4.2. Nationality hypotheses

The hypotheses for the nationality relation argue that there is a difference in performance when there is a match between applications of foreign nationalities and the call agent who is able to talk to the customer in his native language as opposed to a mismatch in this respect. This strong belief of the process owners states that customers who are spoken to in the language of their nationality are more likely to perform well on the KPI. When searching for the nationalities of the applicants the first analyses revealed however, that the number of nationalities other than the Dutch nationality is very small. In Figure 9 it can be seen that it is so small that besides the

Dutch customers, no nationality produces a sample size large enough to fulfill the requirement of at least 30 observations. The number of customers from Great Britain is the highest with 35, but as this group should be divided into a group where there was a match and one in which there was not, the samples become to small for analysis. Therefore no further analyses where performed as the sample sizes were too small. The belief of the process owners may in fact be true, but as the information of the preferred language of the customer is not available and the nationality does not provide a large enough sample space, these hypotheses cannot be tested.

4.4.3. BKR – credit score complexity

Hypotheses in this relation are that there is a difference between more experienced call agents and less experienced call agents when the complexity (A) of the application increases. The same interpretation is valid for Table 6 as it was for the gender table in section 4.4.1. The call agents characteristic of experience as a DLL call agent is divided into either less than one year or one year and more. The complexity is determined by the BKR and the credit score. The lower the row in the table, the higher the complexity and the more the difference in experience should be of influence on the KPIs.

BKR – credit score complexity		Call agent	
		<12 months experience as DLL call agent	=>12 months experience as DLL call agent
Application	No BKR or 'gray' credit score	1	2
	Either BKR or 'gray' credit score	3	4
	Both BKR and 'gray' credit score	5	6

Table 6: BKR – credit score complexity hypotheses possible matches

4.4.4. Gap complexity hypotheses

The hypotheses defined for the gap relation are that there is a difference between more experienced call agents and less experienced call agents when the complexity of the application increases. The complexity level of an application is now divided into two, either the application does not have a positive gap between the maximum loan capacity and the initial requested amount or it has a positive gap. The presence of a positive gap entails extra effort for the call agent in order to reduce the size of the gap. These applications are therefore considered more complex and the influence of the more experience called agent should be noticeable in the performance.

<table>
<tr><th colspan="2" rowspan="2">Gap complexity</th><th colspan="2">Call agent</th></tr>
<tr><th><12 months experience as DLL call agent</th><th>=>12 months experience as DLL call agent</th></tr>
<tr><td rowspan="2">Application</td><td>No positive gap between maximum loan capacity and initial requested amount</td><td>1</td><td>2</td></tr>
<tr><td>Positive gap between maximum loan capacity and initial requested amount</td><td>3</td><td>4</td></tr>
</table>

Table 7: Gap complexity hypotheses possible matches

4.4.5. Higher credit class complexity hypotheses

Again the hypotheses to test, state that there is a difference in performance between more experienced call agents and less experienced call agents when the complexity of the application increases. As can be seen in Table 8, the complexity of the application now is defined as being in a €1,000.- range of a higher credit class while the maximum loan capacity allows the higher credit class. When the application is, extra effort is expected from the call agents, interpreted as a higher complexity for the application.

<table>
<tr><th colspan="2" rowspan="2">Higher credit class complexity</th><th colspan="2">Call agent</th></tr>
<tr><th><12 months experience as DLL call agent</th><th>=>12 months experience as DLL call agent</th></tr>
<tr><td rowspan="2">Application</td><td>Not in €1,000.- range of a higher credit class while the maximum loan capacity allows the higher credit class</td><td>1</td><td>2</td></tr>
<tr><td>In €1,000.- range of a higher credit class while the maximum loan capacity allows the higher credit class</td><td>3</td><td>4</td></tr>
</table>

Table 8: Higher credit class complexity hypotheses possible matches

4.4.6. Part of the day registration hypotheses

The hypotheses for this relation define that there is a difference in performance when there is a match between the part of the day of the registration of the application and the part of the day the application is complemented as opposed to complementing it in another part of the day.

<table>
<tr><th rowspan="2">Part of the day registration</th><th colspan="2">Call agent</th></tr>
<tr><th>Complementing in another part of the day as registration</th><th>Complementing in the same part of the day as registration</th></tr>
<tr><td>Application</td><td>1</td><td>2</td></tr>
</table>

Table 9: Part of the day registration hypothesis possible match

4.4.7. Queue

The queue hypothesis claims that there is a difference in performance of more successful call agents and less successful call agents on the complementing application queue. Only this queue is analyzed, as it is the queue for the most important call during the process and call agents actually are allowed to carry the application over to the next queue. The success percentage of 34% is chosen because this is the average value of it over all call agents. Less then 34% therefore means below average success percentage and the alternative is above average.

<table>
<tr><th colspan="2" rowspan="2">Queue</th><th colspan="2">Call agent</th></tr>
<tr><th>Less than 34% successful transfers to next queue</th><th>34% or higher successful transfers to next queue</th></tr>
<tr><td>Application</td><td>Focus on the most important queue: complementing application</td><td>1</td><td>2</td></tr>
</table>

Table 10: Queue hypothesis possible match

4.4.8. Account manager

The hypothesis created for this relation state that there is a difference in performance of applications which have a higher percentages of notes in the system, made by the same call agent who performed the complementing application call. The notes in the system correspond to calls made with the customer. When this percentage is high, the customer may have the impression the call agent is his account manager and this may result in the customer being more eager to accept the offer of the organization.

<table>
<tr><th rowspan="2">Account manager</th><th colspan="2">Call agent</th></tr>
<tr><th>Less than or equal to 50% of the notes for the application are made by the complementing application call agent</th><th>More than 50% of the notes for the application are made by the complementing application call agent</th></tr>
<tr><td>Application</td><td>1</td><td>2</td></tr>
</table>

Table 11: Account manager hypothesis possible match

All hypotheses stated in this section are tested against all the KPIs with the statistical analysis that is described in section 2.5, unless stated otherwise in the next chapter.

5. Implementing phase

In this chapter the implementing phase of the RecLog^2 method is described as it was performed within the Freo application process. The three steps in this phase correspond with the sections of this chapter.

5.1. Validate hypotheses on historical data

In this section the validation of the defined hypotheses from the first phase of the method is performed. First the challenges to retrieve the operational data are listed, then the dataset that is used for the statistical analyses of the hypotheses is described and finally the results of the statistical analyses are provided in the last section.

5.1.1. Challenges of obtaining relevant operational data

As is mentioned earlier, the obtainment of the relevant operational data is by no means a trivial task. DLL by law is obliged to record and log an enormous amount of data about all products it offers. The first task is to get insight in which data may be available in the process and to identify what data is necessary for the hypotheses. With the use of Oracle Toad 9.7.2 data is extracted from the datawarehouse. During numerous iterations the SQL query in Toad is adapted to suit the requirements for the statistical analyses. When necessary the help of direct process support staff or the datawarehouse management team is called in to ensure the quality of the data. To be able to gather the data detailed process knowledge had to be obtained together with insights on the interaction between support systems and datawarehouse policies. The final SQL query, needed to gather the data, required more than ten hours to do so. This to illustrate the complexity of the task of obtaining the relevant operational data.

5.1.2. Freo dataset

The dataset for these analyses is selected in the following manner. Because the call agent are the entities to whom the recommendation is to be given, they are center to the collection of relevant applications. This is necessary as the possible relations to be analyzed have elements of both the application as of the call agent in them. When either of these elements is missing, the relation cannot be tested. Part of the call agent characteristics are gathered with a letter to all the call agents. This letter is only received from call agents who are still employed at DLL. The dataset used for the analyses is therefore limited to applications with only these call agents. The statistical analysis provides more significant differences between samples when there is a high value of observations. The choice is made to include all applications for this group of call agents in the dataset. This resulted in a dataset that consists of applications from the 5th of March 2010 until the 8th of February 2012 that either directly received the accepted state by the credit decision engine or were sent to the handle leads queue. In both cases the application is represented to the call agent as a work item in the information system Advisor, that supports the process. All automatically declined application are therefore not included in this analysis. As mentioned only applications of which the call agents characteristics are known are taken into account. From the datawarehouse a dataset is extracted containing 17.338 applications.

5.1.3. Results of the statistical analyses

The first step in evaluating the relationships is to see the number of applications that are considered for the four KPIs, without any selection of call agent characteristics. In Table 12 it can clearly be seen that the activation KPI is available for all applications. The gap and other credits KPIs have respectively 45% and 43% of valid observations in the dataset. The higher credit class KPI (CRC) has just a very small fraction (<<1%) of which the applications fulfill the requirements for the KPI. This is to be expected as there were strict requirements for this last KPI.

	ACT	GAP	OTH	CRC
Valid	17338	7850	7370	71
Missing	0	9488	9968	17267
Mean	0.47537	0.23847	0.46079	0.53521

Table 12: Maximum sample size for the KPIs

The small number of valid observations will limit the possibility to provide significant results in the tests. Although the second and third KPI also have missing values, because not for all the applications the existing outstanding credit with other credit providers is available, the much higher number of valid observations of the other KPIs will have more possibilities to find significant dependencies or relationships. Also in Table 12 the means of the KPIs are displayed. As the KPIs are modeled with the binomial distribution which only contains the values 0 and 1, the means give an indication about the number of successful observations per KPI. Therefore the mean equals the proportion of successful applications in the dataset. This entails that 47.5% of the applications within the scope of this project are activated. For the other KPIs the same reasoning applies. In the next sections the defined hypotheses are analyzed.

5.1.4. Gender hypothesis

The first hypotheses to check are the gender hypotheses. These hypotheses have four possible matches. Because these hypothesis are based on intuition, no preference is made and all combinations are considered. The four possible matches are displayed in Table 13. Because all inputs and results of tests are added to this thesis in App. A table 3: Test inputs and results, only the interesting columns are elaborated on here. For these first analyses Table 14 is explained in great detail. Later similar tables are constructed in the same way. There are eight columns in the table. The first is a number to be able to identify the test that was performed. The second shows for which expected relation the test is performed. In the third column one can see for which KPI the test is done. Columns four and five indicate which samples are compared. The first sample provides a value for p_1 and the second sample for p_2. From Table 13 the samples can be identified with the characteristics that correspond to them. The last three columns correspond to the three alternative hypotheses H_1. When a 0 appears in all three columns the alternative hypotheses are not accepted. This means the test statistic of the statistical analysis was not sufficient to significantly reject H_0. When a 1 appears in any of these columns the H_0 hypothesis is rejected and that alternative hypotheses is accepted. At most two cells can contain a 1 in each row, indicating that there is a strong dependency. When one cell in a row contains a 1, there is a significant dependency, but less strong than when two cells contain a 1. The header of the last three columns indicates to which alternative hypothesis the column belongs.

To illustrate the interpretation of the tests within this section test 1 from Table 14 is explained in great detail. The hypothesis tested, considered the gender relation as can be seen in the second column. The tested KPI is the activation KPI which is indicated by ACT in the third column. The first and second sample refer to the 1 and 2 in Table 13. In this table one can see that the first sample (1) contains observations of the KPI where the female applicants are called by a female call agent. Also one can see that the observations in the second sample (2) are female applicants who are called by male call agents. Note that the characteristic of the applicant is kept constant, as the process does not have influence on it. It does have influence on the gender of the call agent who calls the applicant. This is why the female – male difference for the call agents exist between the two samples. The two samples both have a proportion of observations that was successful. These proportions are presented in the hypotheses by p_1 and p_2. The tested hypothesis H_0 is that there is no difference between these two parameters (not listed in the table, see Figure 3 on page 15). The last three columns of test 1, all contain a 0. This means that the hypothesis H_0 can not be rejected and that therefore none of the alternative hypotheses may be accepted. In English this means, that there is no significant difference between the samples and that it does not matter if a female or male call agent calls an application of a female applicant. They both perform equally well and no dependency which can be used for the recommendation logic is found.
The second test shows a different result. This hypothesis indicates that there is a significant difference of means between the samples of matches 3 and 4, as can be seen in Table 14. The alternative hypothesis H_1: $p_1 > p_2$ is found to be valid for the activation KPI. This alternative hypothesis says that the successful proportion of the sample of p_1 is significantly larger that the proportion of p_2. This means the first sample (p_1) performs better than the second sample (p_2). This result means that when the primary applicant of an applications is of the male gender, he can best be contacted by a female call agent for the complement application call, because they perform significantly better than their male colleagues.

Gender		Call agent	
		Female	Male
Primary applicant	Female	1	2
	Male	3	4

Table 13: Gender hypotheses possible matches

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
1	Gender	ACT	1	2	0	0	0
2	Gender	ACT	3	4	0	1	0
3	Gender	GAP	1	2	0	0	0
4	Gender	GAP	3	4	0	0	0
5	Gender	OTH	1	2	0	0	0
6	Gender	OTH	3	4	0	0	0
7	Gender	CRC	1	2	0	0	0
8	Gender	CRC	3	4	0	0	0

Table 14: Gender significant relation

5.1.5. BKR – credit score complexity hypothesis

The second hypothesis is the BKR – credit score complexity hypothesis, as the nationality hypothesis could not be analyzed. These hypotheses look at the open existing credits that exists (BKR) and the possible credit score in the range of 600 to 704, the so called 'gray' area. These application characteristics are compared with the experience characteristic of the call agents. The call agents are divided into two groups, the first group has less than 12 months experience as a DLL call agent, the other group has 12 months or more experience. The possible matches are given in Table 15.

BKR – credit score complexity		Call agent	
		<12 months experience as DLL call agent	=>12 months experience as DLL call agent
Application	No BKR or 'gray' credit score	1	2
	Either BKR or 'gray' credit score	3	4
	Both BKR and 'gray' credit score	5	6

Table 15: BKR – credit score complexity hypotheses possible matches

From Table 16 one can see there are no less than six significant relations found. All the significant relations state that the more experienced call agent performs better than the less experienced call agent. The level of complexity does not really influences this as the relation is constant over the different complexity levels. Especially the no complexity level is susceptible for the more experienced call agents.

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
9	Comp A	ACT	1	2	1	0	1
10	Comp A	ACT	3	4	1	0	1
11	Comp A	ACT	5	6	1	0	1
12	Comp A	GAP	1	2	1	0	1
13	Comp A	GAP	3	4	0	0	1
14	Comp A	GAP	5	6	0	0	0
15	Comp A	OTH	1	2	0	0	1
16	Comp A	OTH	3	4	0	0	0
17	Comp A	OTH	5	6	0	0	0
18	Comp A	CRC	1	2	0	0	0
19	Comp A	CRC	3	4	0	0	0
20	Comp A	CRC	5	6	0	0	0

Table 16: BKR – credit score complexity significant relations

5.1.6. Gap complexity hypothesis

The third hypothesis focuses on the gap between the maximum loan capacity of the application and the initial principal amount. The possible matches are depicted in Table 17. Again the significant relations, depicted in Table 18, emphasize that the more experienced call agent performs better than the less experienced colleague.

Gap complexity		Call agent	
		<12 months experience as DLL call agent	=>12 months experience as DLL call agent
Application	No positive gap between maximum loan capacity and initial requested amount	1	2
	Positive gap between maximum loan capacity and initial requested amount	3	4

Table 17: Gap complexity hypotheses possible matches

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
21	Comp B	ACT	1	2	1	0	1
22	Comp B	ACT	3	4	1	0	1
23	Comp B	GAP	1	2	0	0	0
24	Comp B	GAP	3	4	1	0	1
25	Comp B	OTH	1	2	0	0	0
26	Comp B	OTH	3	4	0	0	0
27	Comp B	CRC	1	2	0	0	0
28	Comp B	CRC	3	4	0	0	0

Table 18: Gap complexity significant relations

5.1.7. Higher credit class complexity hypothesis

Again the call agents experience is set against an applications' complexity level, as defined in Table 19. The very selective nature of the higher credit class complexity hypothesis causes that only significant relations in the sample size without this selection are found. To no ones surprise the more experienced call agents perform better as can be seen in Table 20 on the next page.

Higher credit class complexity		Call agent	
		<12 months experience as DLL call agent	=>12 months experience as DLL call agent
Application	Not in €1,000.- range of a higher credit class while the maximum loan capacity allows the higher credit class	1	2
	In €1,000.- range of a higher credit class while the maximum loan capacity allows the higher credit class	3	4

Table 19: Higher credit class complexity hypotheses possible matches

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
29	Comp C	ACT	1	2	1	0	1
30	Comp C	ACT	3	4	0	0	0
31	Comp C	GAP	1	2	1	0	1
32	Comp C	GAP	3	4	0	0	0
33	Comp C	OTH	1	2	0	0	0
34	Comp C	OTH	3	4	0	0	0
35	Comp C	CRC	1	2	0	0	0
36	Comp C	CRC	3	4	0	0	0

Table 20: Higher credit class complexity significant relations

5.1.8. Part of day registration hypothesis

Part of the day registration	Call agent	
	Complementing in another part of the day as registration	Complementing in the same part of the day as registration
Application	1	2

Table 21: Part of the day registration hypothesis possible match 1

This fifth hypothesis looks at the part of the day the call agent should (try to) complement the application. The part of the day registration can be in the morning, midday, evening or night. The call agents do not work at night, so they can not complement at night. The first analysis, based on the matches in Table 21, showed it was better to complement the application in a different part of the day then the registration has been. This significant relation is shown in App. A table 3: Test inputs and results, test number 37. However this significant relation conflicts with the belief of the process owners it is best to undertake action on a new application within the first hour of its registration. Extra analyses are therefore performed to explain this. Zooming in on the KPI activated, that has this valid relation, a new set of possible matches is created in Table 22.

Part of the day registration		Call agent	
		Complementing in another part of the day as registration	Complementing in the same part of the day as registration
Application	Morning	3	4
	Midday	5	6
	Evening	7	8
	Night	-	Not possible

Table 22: Part of the day registration hypotheses possible matches 2

These new possible matches revealed two very interesting and strong relations. The significant relations showed in Table 23 mean that the morning applications perform best when they are complemented in the same part of the day. However, for the evening applications this is not the case. There the relation indicates that an application which is registered in the evening, should not be complemented in the evening.

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
41	Part	ACT	3	4	1	0	1
42	Part	ACT	5	6	0	0	0
43	Part	ACT	7	8	1	1	0

Table 23: Part of the day registration significant relations

The relation that call agents should not complement applications that were registered in the evening still conflicts with the management opinion that all applications registered during working hours should be called within one hour, because this would mean at least in some cases to complement in the evening. The process owners tried to explain this by assuming that the applications registered in the evening should be of less quality than the other applications. Also the question arises if there is a 'best' part of the day to have applications registered. Although this is a given for this process, it could prove the process owners are right. The tests 44 until 46 show however, that evening registered application are better complemented in the morning, next best in the midday and least best in the evening. Extra tests 47 until 52 show that there is indeed a least performing part of the day for the registration of applications. This is however not the evening, but the night. Remember, registrations can happen in any part of the day, but the call agents do not perform calls at night. Therefore one must conclude that although the applications registered in the evening are equal of quality compared to the morning and midday, when the evening applications are also complemented in the evening, they lead to significantly lower performance on the KPIs.

5.1.9. Queue hypothesis

This hypothesis matches the queue in which the application is, with the historic success percentage of the call agent on that queue. As call agent characteristic the average success percentage is used as the boundary. When a call agent has a success percentage of 34% or higher he is labeled as successful. The possible matches are given in Table 24 for the complementing application queue. Only this queue is considered as it is the most important queue of the process and most suitable for a recommendation to the call agent.

<table>
<tr><th colspan="2" rowspan="2">Queue</th><th colspan="2">Call agent</th></tr>
<tr><th>Less than 34% successful transfers to next queue</th><th>34% or higher successful transfers to next queue</th></tr>
<tr><td>Application</td><td>Focus on the most important queue: complementing application</td><td>1</td><td>2</td></tr>
</table>

Table 24: Queue hypothesis possible match

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
53	Queue	ACT	1	2	1	0	1
54	Queue	GAP	1	2	0	0	0
55	Queue	OTH	1	2	0	0	0
56	Queue	CRC	1	2	0	0	0

Table 25: Queue significant relation

Test 53 presented in Table 25 shows that a more successful call agent performs better than a less successful call agent. The ability to carry the application to the next queue (successful completing calls) is a indicator for better performance on the activation KPI.

5.1.10. Account manager hypothesis

This last hypothesis is somewhat different from the rest. As this information only came available, after the first analyses were performed. Therefore this hypothesis is tested on a special created dataset with 1.641 applications. For this dataset only the activation KPI was available, so this was the only KPI tested. The possible match for this hypothesis is given in Table 26. The term note comes from Advisor. Whenever a call agent performs actions on an application he is expected to note what actions were performed and the note in this hypothesis refers to a call with the customer.

Account manager	Call agent	
	Less than or equal to 50% of the notes for the application are made by the complementing application call agent	More than 50% of the notes for the application are made by the complementing application call agent
Application	1	2

Table 26: Account manager hypothesis possible match

The last test in App. A table 3: Test inputs and results, which is presented in Table 27, provides no significant difference between applications which have more than 50% notes of the same call agent that complemented the application and the applications who have not. There is therefore no need to assign call agents to the same application to mimic an account manager.

Test	Relation	KPI	First sample	Second sample	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
57	Account	ACT	1	2	0	0	0

Table 27: Account manager no significant relation

5.2. Select appropriate valid relations

The previous section displayed the results obtained from the statistical analyses on the different hypotheses. In this section the valid relations for the recommendation logic are selected.

5.2.1. Overview Freo significant relations

In total 22 significant relations have been found. Some of them are very similar to each other, like the more experienced call agents performing better, others are more unique like the significant gender relation between male applicants and female call agents. An overview of all the found significant relations is provided in Table 28.

Hypothesis	KPI	Significant relation
Gender	ACT	Applications with a primary male applicant can better be complemented by a female call agent
Comp A	ACT	A more experienced call agents perform better than less experienced call agents no matter what the complexity level of BKR – credit score is
Comp A	GAP	A more experienced call agents perform better than less experienced call agents no matter what the complexity level of BKR – credit score is (to small sample size for highest complexity level)
Comp A	OTH	A more experienced call agents perform better than less experienced call agents on the lowest complexity level of BKR – credit score
Comp B	ACT	A more experienced call agents perform better than less experienced call agents no matter what the gap complexity level is
Comp B	GAP	A more experienced call agents perform better than less experienced call agents no matter what the gap complexity level is
Comp C	ACT	A more experienced call agents perform better than less experienced call agents on the lowest complexity level of higher credit class
Comp C	GAP	A more experienced call agents perform better than less experienced call agents on the lowest complexity level of higher credit class
Part	ACT	Applications complemented on a different part of the day as the registration part of the day perform better
Part	ACT	Applications registered and complemented in the morning perform better than applications complemented in another part of the day
Part	ACT	Applications registered in the evening and complemented in another part of the day perform better than the applications complemented in the same part of the day
Part	ACT	Applications registered in the evening are better complemented in the morning than in the evening
Part	ACT	Applications registered in the evening are better complemented in the midday than in the evening
Part	ACT	Applications registered in the evening are better complemented in the morning than in the midday
Queue	ACT	Applications complemented by call agents with a higher success percentage on the complementing queue perform better then application complemented by call agents with a lower success percentage on the complementing queue

Table 28: Overview Freo significant relations

The found valid relations are discussed with the process owners in order to exclude any undesired relation. Fortunately no relations were dismissed during this process. The gender relation is significant and therefore a good candidate relation for the recommendation logic, but it is less strong than some other relations because the H_1 hypothesis $p_1 \neq p_2$ could not be accepted, attributing less power to it.
There were no conflicting relations, but there were some, less useful for the recommendation logic. The complexity relations all indicated that the more experienced call agent is preferred above the less experienced one. This was the case for all levels of complexity thereby limiting the recommendation possibilities. It is expected that with the low levels of complexity the less experienced call agents would not be significantly underperforming in comparison to the more experienced call agents. This result would have lead to assigning low level complexity applications to less experienced call agents and the higher level complexity applications to the more experienced call agents. Because this distinction cannot be made, the recommendation power is reduced and these relations are not included in the recommendation logic.
Unfortunately the process supporting information system Advisor is to be replaced shortly. No investments in or changes to the system are to be made. Therefore it was not possible to implement the recommendation logic to work automatically within the system. It also limited the possibilities to present the recommendation to the call agents. For instance the success percentages of the call agents could not be shown to them, eliminating the queue relation from the recommendation logic.
The remaining relations are the gender relation and the part of the day relations. The part of the day relations can be divided into two recommendation rules. The first is that the applications that are registered in the morning should be complemented in the morning. When this is not successful in the same morning, the process owners decided it would be to long to reschedule to the next morning. Therefore the relation is not included in the recommendation logic, because the current business rule of contacting the customer within one hour of registration is very similar to what the recommendation rule would look like. The other relation of the applications registered during the evening is included in the recommendation logic. The process management, somewhat reluctantly, allowed for the recommendation rule to not contact applications registered in the evening within the hour, but to schedule them in for the next morning. This follows three of the part of the day relations, because if the attempts in the next morning are unsuccessful, attempts in the midday are made and if this still is unsuccessful the customer is contacted in the evening.

5.3. Define and implement recommendation logic

After the selection step two relations are left. The gender relation and the not complement in the evening relation. The definition of these two relations results in the recommendation logic with two recommendation rules:

- Applications with a male primary applicant should be called by a female call agent for the complement application call.
- Applications which are registered or submitted in the evening (after 18:00 hour) must not be contacted for the complementing application call that evening. The first call attempt has to be the following morning.

Because no changes may be made to the supporting information system, the recommendation logic is implemented by communicating it to all the call agents via a mail from their manager and the recommendation rules are prominently included in the cheat sheets of the call agents used during their work. To make the recommendation rules more visible to the call agents the registration time in the queues, which already was available, was highlighted and the gender of the primary applicant is made visible by the call agents themselves by setting the scheduled contact time to 1 minute past the whole hour for females and 2 minutes past the whole hour for male applicants. The call agent team is young and flexible and enforcing the recommendation logic by making it mandatory does not suit the situation. Process management committed to the rules for a test period and helped to ensure the call agents respect the recommendation logic as much as possible.

6. Test of the recommendation logic

6.1. Performance predictions

A very helpful tool in getting support for the implementation of the recommendation logic, once it is created with the developed RecLog^2 method, would be a prediction of its influence. This step is not included in the method, because its validity is not been proven. For this further research is necessary. To provide some insight in the possible influence a prediction of the improvement per recommendation rule is provided in this section. Note that this predication does not hold any guaranties and it is probably more the upper level of the improvement than the expected improvement. However a prediction of the influence of the recommendation logic may be valuable for the process owners during their decision making to implement the logic. As this is only to show how a prediction could be formulated, the next sections show the prediction for the recommendation logic that is implemented in the Freo application process on only the activation KPI.

6.1.1. Gender relation performance improvement prediction

The recommendation rule states that male applicants must be contacted by female call agents for the complement application call. The dataset can again be divided into samples. One sample containing male applicants called by any call agent and one sample containing male applicants called by female call agents. The average values of the samples on the activation KPI provide a notion of the degree of performance. Under the assumption that the recommendation rule is followed in a strict manner and that the extra workload is handled by the female call agents in exactly the same way as before. A prediction for the performance may be derived by subtracting the average performance of the any call agent sample from the average performance of the sample in which the female call agents call the male applicant. For this recommendation rule this would result in the following computation: 0.495063 – 0.487085 = 0.007978. This is an increase in performance of 0.80%. With around 324 to 486 applications with male applicants a week, this could on average result in 3 extra activated applications a week and 168 extra activated application per year. However keep in mind that this is not statistically justified. The statistical analysis fails as these samples are no longer independent from each other, because they overlap. To be more precise, one sample is a subset of the other.

6.1.2. Part of the day relation performance improvement prediction

The same approach for the prediction can be included for the part of the day relation. Although the same predicaments hold as before. Now the selection from the dataset containing all the applications registered in the evening (between 18:00 and 0:00 hour) is compared to the sample containing the applications also registered in this part of the day, but not complemented in it. This is what the second recommendation rule prescribes. Again the averages of the samples are computed and the performance of the bigger sample is subtracted from the performance of the smaller sample. This leads to: 0.473723 – 0.391175 = 0.082548. This is an increase in performance of 8.25%. With around 123 – 184 applications registered in the evening each week, under the assumption that all of these are complemented during the morning or midday, the number of extra activated applications per week on average is 12. Which could translate to 658 extra activate applications per year. Again, these predictions are to be investigated further, but they do give an idea of the possible gain.

6.2. Test results

The testing of the implemented recommendation logic was unfortunately cut short due to circumstances beyond the control of this project. Because of time limitations the initial test period was already short and the missed testing time eliminated the possibility to statistically justify the influence of the recommendation logic on the process KPIs. Nevertheless effort is spent to give at least an indication of the performance of the process with the recommendation logic in place. The dataset consists of the applications registered in the week from 16th of February until the 23rd of February 2012. To provide the indication of the performance two graphs are created. The graphs are presented as Figure 10 and Figure 11. As explained the test did not run for a long enough period to be able to determine the total influence of the recommendation logic on the performance. This is due to the fact that only 221 applications within scope were registered and 15% of them are still in progress at the time of this analysis. However, the other applications did reach their end state, completing the process. This group may be used to illustrate their performance against similar historical applications. To be able to do so one must consider time within the process. The longer the applications are in the process, the greater the chance they have reached their end state. In both Figure 10 and Figure 11 the horizontal axis represents the number of working days the application has been in the system. The vertical axes of both figures have two scales assigned to them. The first is the number of applications that reached the end state of the figure on that number of working days in the system. This is visualized with the blue line with diamonds on it. The peak of this line is at the 12th working day in the system with 391 applications being activated in the historical dataset. This line is then transformed to a cumulative percentage line for all the applications in the dataset. The yellow line with triangles is drawn to visualize this against the secondary vertical scale, which represent the percentage (right side of the graph). Against this same scale the last line is set. The pink line with squares represents the cumulative percentages of activated applications within the test dataset. The same has been done for the negative end states, declined and cancelled. Both graphs are constructed in the same way.

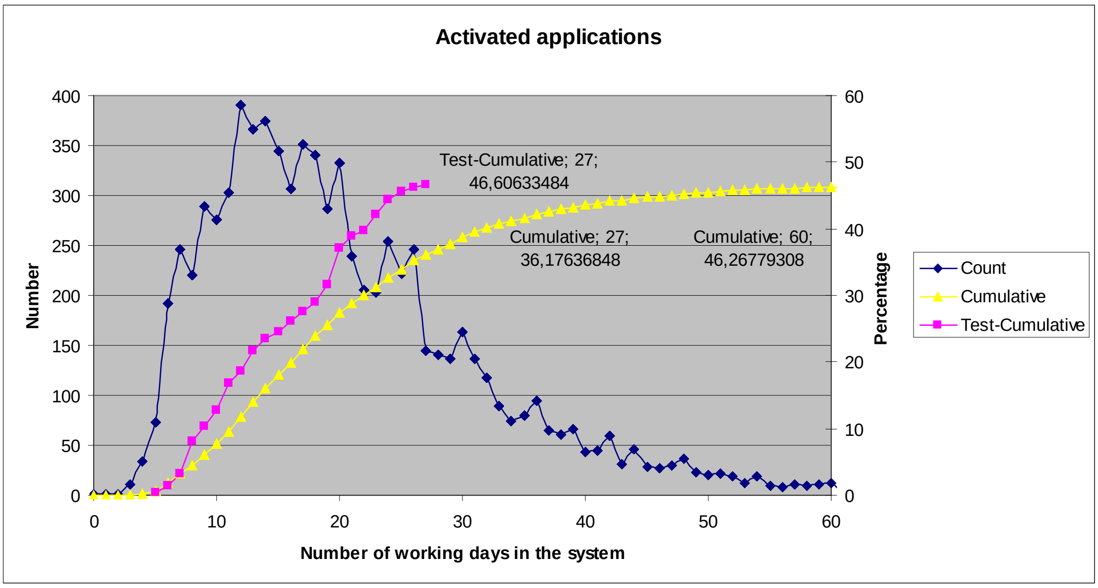


Figure 10: Test results – positive end state of application

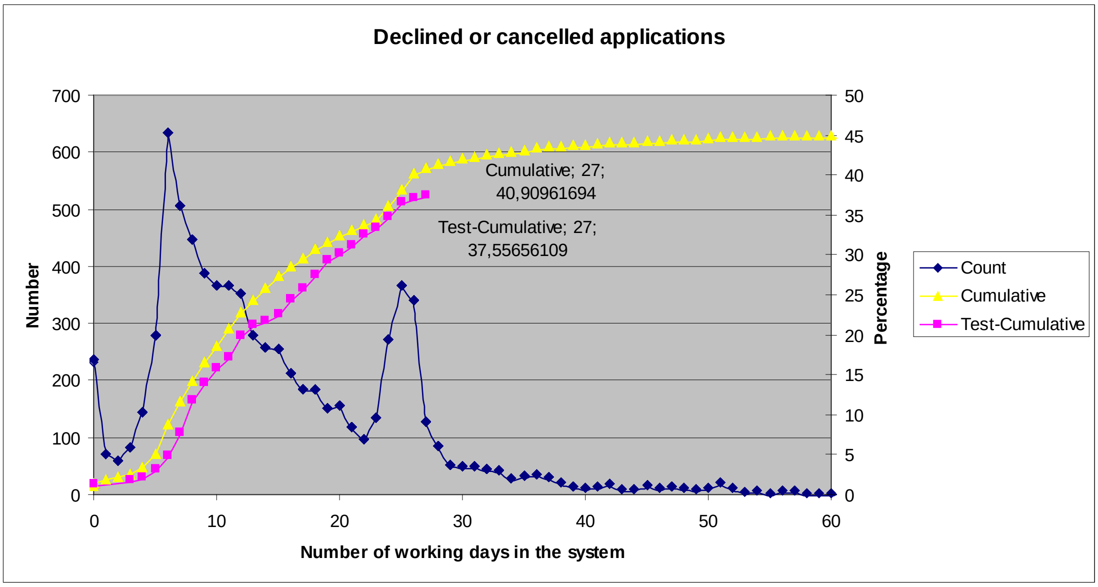


Figure 11: Test results – negative end state of application

The results in both figures looks very promising. In Figure 10 the test cumulative percentage of activated applications (pink) is quite a bit higher than the cumulative percentage of the historical dataset (yellow). As mentioned earlier, it is to soon to conclude that the process with the recommendation logic performs significant better, because not all applications have reached their end state. In the first 27 days however the added value of the recommendation logic is there. One can even see that the test cumulative percentage at the 27th working day is higher than the cumulative percentage at the 60th working day. A very promising result.

To switch to the negative end states in Figure 11, in this figure the pink line is preferably under the yellow line. Indicating less applications are declined or cancelled. This is the case, but with a smaller difference than can be seen in Figure 10.

Together the results of the test look very promising, but further research must be conducted to prove the influence of the RegLog^2 method on KPIs.

7. Conclusions

In this chapter the conclusions regarding the developed RecLog^2 method are given. First a summary is provided about the results of the method, then a reflection is provided about the process of its development and the last section describes future work regarding the RecLog^2 method that is identified during the project.

7.1. Summary

In this master thesis the RecLog^2 method is developed to create recommendation logic in order to improve the performance of an operational service process on its KPIs. The model for this RecLog^2 method is described in detail in chapter 3 and is provide in Figure 1. To test the method, it is implemented in the Freo application process at DLL. The chapters 4 and 5 provide all the activities performed to successfully implement the RecLog^2 method. Successful in the sense that all steps of the method are followed, all activities are performed and the result of the implementation efforts is recommendation logic, that is implemented in the current process. With this successful implementation an answer is provided to the main research question, which was: "How to improve performance of an operational service process on its KPIs, using historical data for improving work-item assignment?" The developed RecLog^2 method is in fact an appropriate method to set up the recommendation logic for a given operational service process of which the activities or transactions of entities interacting within the process are logged. The only part of the research question that is not fully proven, is the influence of the recommendation logic on the KPIs. The testing of the recommendation logic is not completed and therefore the actual influence of the recommendation logic is still unknown. However the first results of the test are hopeful and look very promising. The set up of the RecLog^2 method is developed in such a way that all valid relations that are used in the recommendation logic at least have had influence on the KPIs in the past. Therefore the chance that this relation is still present and can be used to improve the process performance definitely exists. Together with the positive first test results this argues in favor of the developed RecLog^2 method. Besides the recommendation logic that is designed in the method, an incredible amount of process insight is also gained during the performance of the activities. This insight alone is already a source of added value to the organization. Furthermore the insights may also include other kinds of valuable information about the process. Take for instance the more experienced call agents from the Freo application process. They performed significantly better than their less experienced colleagues, which could be a reason for the process owners to invest more in call agents who are more likely to stay committed to the organization for a longer period of time. This focus will prevent the organization from investing in only the period in which the call agent is outperformed by his more experienced colleagues and ensures that the benefits are gained when the call agent accumulates enough experience.

7.2. Reflection

Although the RecLog^2 method in this master thesis is represented by a sequential model, the actual order of steps taken during the implementation of the method in the Freo application process was different. This is not because the RecLog^2 method model is wrong, but because the method is developed while implementing it. Also the earlier work within the operational process provided a lot of insight, which of course was not ignored during the implementation. When reflecting back on the process, the sequential order of the method facilitate the most natural and logical way to begin the process of recommendation logic development. It is also possible that the inexperience with the process before analyzing it, revealed more improvement opportunities than that may have been found by someone with already a lot of experience with the process and who may have developed a blind spot for these opportunities. The improvement opportunities that were found, were also tested and used in this method. Directly linking intuition to a method for validating this intuition. In that sense the process has been very successful in testing ideas. This was largely thanks to the open and informal nature of the work environment in which both the process owners and participants were open for suggestions and their willingness to get involved. The direct communication between the involved parties certainly contributed in fast and clear information sharing. The delivered added value during the process kept everybody enthusiastic and motivated to keep helping and producing. The combined insights, intuitions, knowledge and expertise of all people involved lifted the result of the process to a higher level and that is certainly worth mentioning. Of course there were also points for improvement. Although communication was direct, in some instances a miscommunication occurred which lead to wasted hours spent on finding information that simply was not there. What definitely can be improved is the time spent in creating the SQL query to gather the information for the dataset. The master thesis mentions that this crucial step is time consuming, but that may be an understatement. More involvement of the datawarehouse experts can reduce the creation time significantly, but usually these people already have more than enough tasks to fill up their day. An even more proactive involvement of these experts could have shortened the throughput time of the project. Nevertheless the datawarehouse managers of Freo always found time to help, when the project was really stuck and that is very valuable.

7.3. Future work

The process of developing the RecLog^2 method presented in this master thesis has been done very thoroughly. The method is generalized from the implementation in the Freo application process. It would be very interesting to research another implementation of the RecLog^2 method in a different operational service process. The new insights gained from a second implementation may be reason to change small elements of the method. When successful the extra implementation would also contribute to the method by providing evidence that it is general and appropriate for more operational service processes.

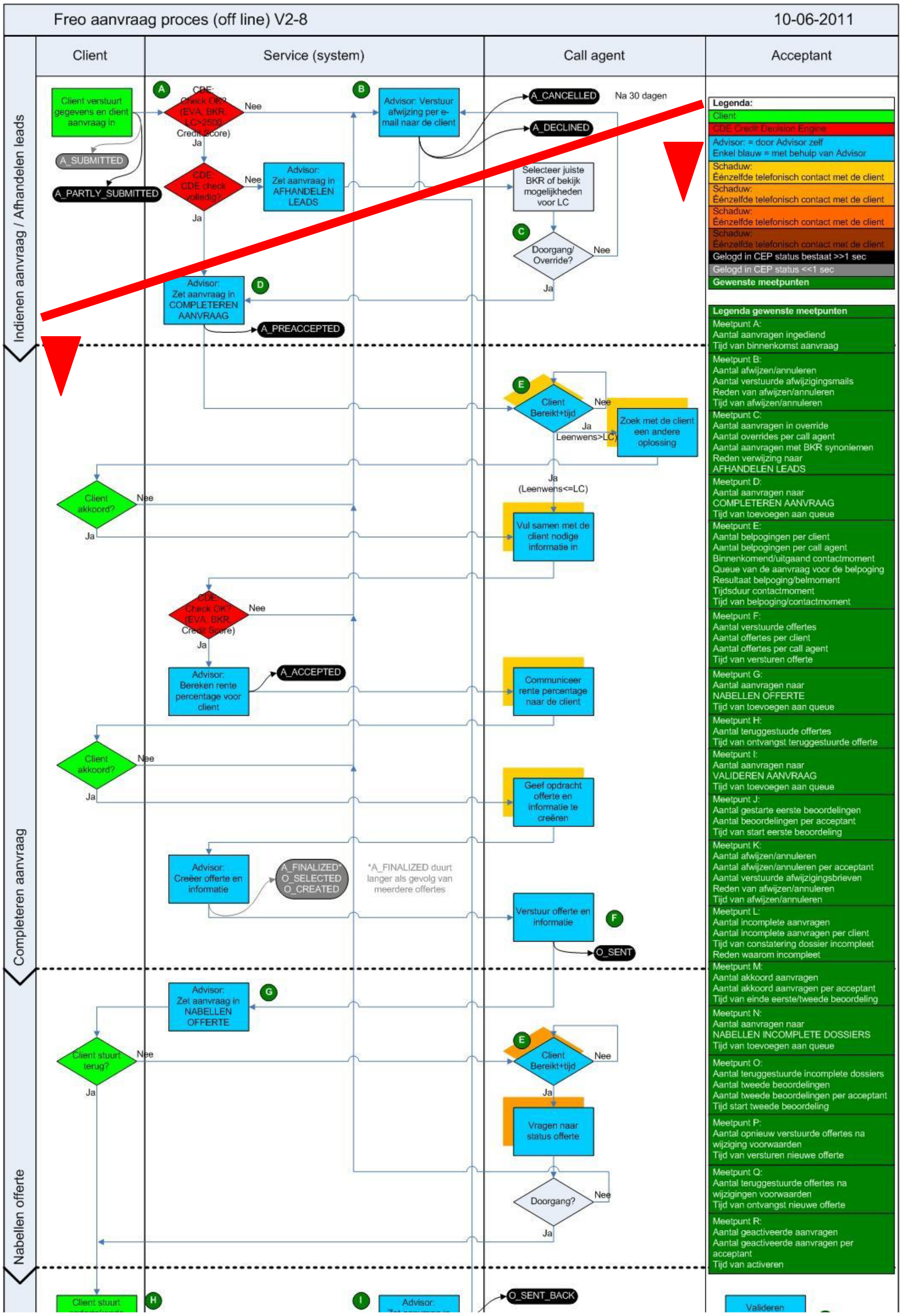
Another point of view can focus on the influence of different sample sizes on the statistical analysis. The incomplete test of the implementation as DLL indicated that for smaller sample sizes the difference between the two means has to be greater. This is a logical consequence of the formula for the test statistic that is used. It would be very interesting to know how different sample size relate to rejecting the hypothesis or not. This research is especially valuable for the validation of the proposed recommendation logic. Insight in this influence may contribute in designing a more appropriate test period for the recommendation logic, to actually prove it outperforms the process without the recommendation logic.

A definite asset to the RecLog^2 method, in getting support for the implementation of the recommendation logic, would be an extra step that, more accurately, predicts the its influence on the process KPIs. Because the statistical analysis uses averages one could argue that implementing the recommendation logic would increase the total performance of the process with a certain degree dependent on the difference of the two samples and the number of occurrences within the actual process. This is only

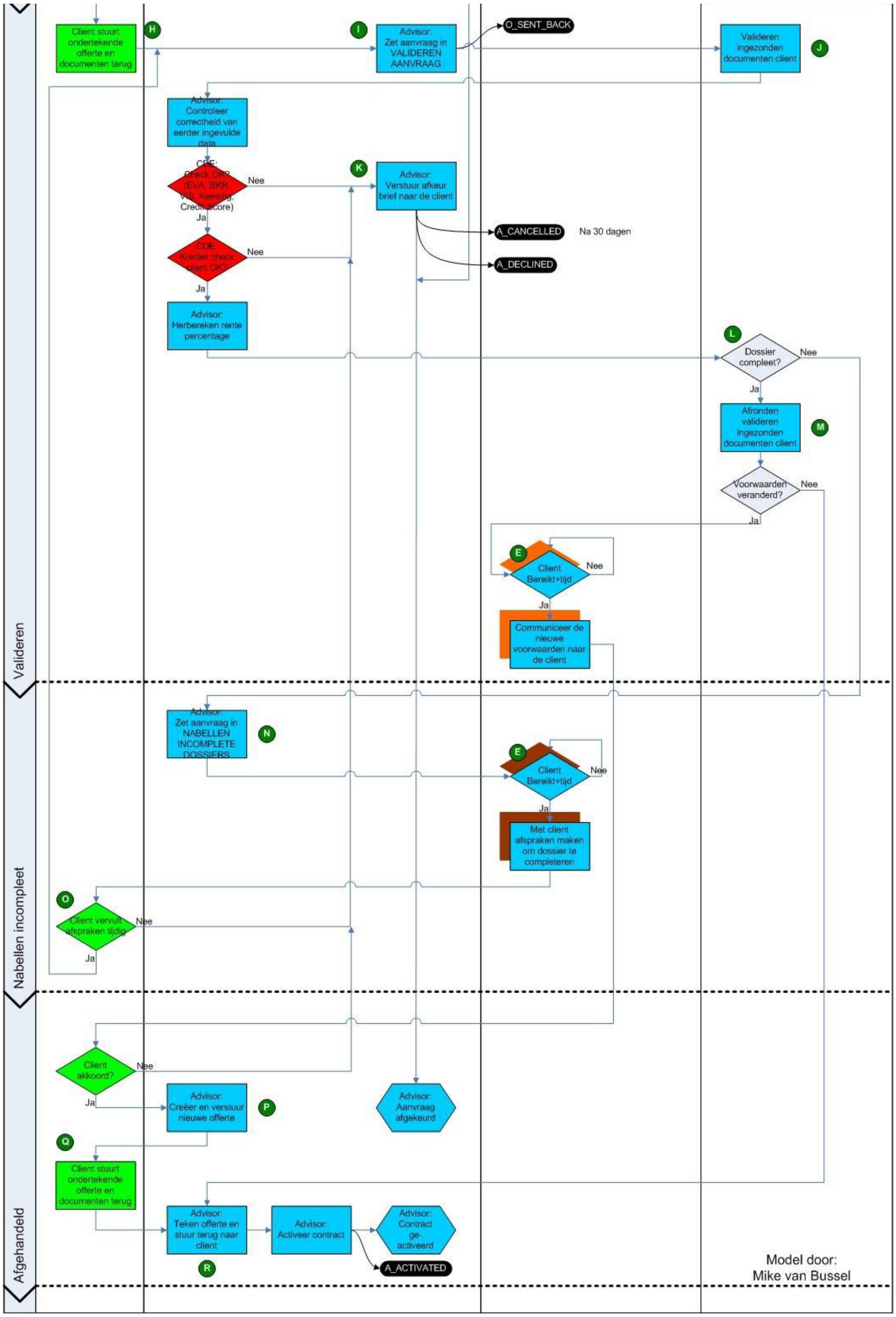
touched upon in this master thesis. Further research with longer tests and within other operational service processes are needed to make these predictions better.
A last area of research may be in defining the minimal requirements for an operational service process to be suited for implementing the developed RecLog^2 method. This could eliminate, futile efforts of implementing the method in processes that are not well fitted for it.
Practical further investigation can of course be done by running another test at Freo to prove the influence of the recommendation logic on the KPIs. A new test can be designed and carried out that will produce results more suitable for interpretation. A positive outcome of such a test would be beneficial for the method.
The vulnerability of expected dependencies analyzed in the RecLog^2 method to dependencies that go even deeper then expected may also qualify for extra investigation. This may answer possible questions as to if the better performance of a certain entity is not caused by another characteristic as was assumed. Such hidden dependencies may undermine implementations of the method.
To conclude a question about the time dependency of the data in the dataset may also prove very insightful for the RecLog^2 method. Are relations found in a dataset also present is a subset of the dataset. What kind of conclusions may be drawn if not and is the relation then still a good candidate for the recommendation logic.
There still are a lot of good questions to be asked concerning the method and only the future will tell if the proposed model for the RecLog^2 method is viable.

8. Appendix

8.1. Appendix A: Figures and tables



App. A figure 1: Freo application process model (part 1/2)



App. A figure 2: Freo application process model (part 2/2)

Nr.	Characteristic	Description
1	Employee number	The number of the employee
2	User reference text	The string which is associate with the employee in the datawarehouse
3	First name	The call agents first name
4	Last name	The call agents last name
5	Experience DLL	The number of months the call agent has experience as a DLL call agent
6	Experience non DLL	The number of months the call agent has experience as a non DLL call agent
7	Working hours	The number of working hours per month
8	Nationality	The call agents nationality
9	Place of residence	The call agents place of residence
10	Preferred work queue	The queue in which the call agent preferably works
11	Foreign language	The foreign languages the call agent speaks
12	Total actions	The total number of actions by the call agent
13	Total success	The total number of actions that resulted in transferring the application to the next queue (successful call)
14	Total percentage	Total success / total actions * 100
15	Leads actions	The number of actions by the call agent in the handling leads queue
16	Leads success	The number of actions by the call agent that resulted in transferring the application to the next queue (successful call) from the handling leads queue
17	Leads percentage	Leads success / leads actions * 100
18	Complementing actions	The number of actions by the call agent in the complementing application queue
19	Complementing success	The number of actions by the call agent that resulted in transferring the application to the next queue (successful call) from the complementing application queue
20	Complementing percentage	Complementing success / complementing actions * 100
21	Offer actions	The number of actions by the call agent in the call after offer queue
22	Offer success	The number of actions by the call agent that resulted in transferring the application to the next queue (successful call) from the call after offer queue
23	Offer percentage	Offer success / offer actions * 100
24	Validate actions	The number of actions by the call agent in the validate application queue
25	Validate success	The number of actions by the call agent that resulted in transferring the application to the next queue (successful call) from the validate application queue
26	Validate percentage	Validate success / validate actions * 100
27	Incomplete actions	The number of actions by the call agent in the call after incomplete applications
28	Incomplete success	The number of actions by the call agent that resulted in transferring the application to the next queue (successful call) from the call after incomplete applications
29	Incomplete percentage	Incomplete success / incomplete actions * 100

App. A table 1: Overview call agent characteristics

Nr.	Characteristic	Description
1	Contract number	The number of the contract with the customer (if any)
2	Application number	The number of the application in Advisor
3	Application ID	The number identifying the application in the DWH
4	Main contractor ID	The number identifying the main contractor in the DWH
5	Registration date	The date and time of registration of the application
6	Registration part of the day	The part of the day of registration of the application
7	Complementing part of the day	The part of the day the application was successfully complemented
8	Status	The current state of the application as in the DWH
9	Initial requested amount	The amount of credit initially requested
10	Requested amount	The amount of credit requested after complementing the application
11	Days in the system	The number of days the applications is in the system, from registration to last change
12	Housing situation	The housing situation of the main contractor
13	Marital status	The Marital status of the main contractor
14	Date of birth	The date of birth of the main contractor
15	Nationality	The nationality of the main contractor
16	Postal code	The postal code of the main contractor
17	House number	The house number of the main contractor
18	Gender	The gender of the main contractor
19	The number of children	The number of children of the main contractor
20	Employment contract	The employment contract of the main contractor
21	Maximum loan capacity	The maximum of credit that the main contractor has
22	Credit score	The credit score given by the credit decision engine of the main contractor
23	Outstanding loans before	The monthly credit cost before the activation of the application
24	Outstanding loans after	The monthly credit cost after the activation of the application
25	Outstanding loans took over	The outstanding loans from other credit providers that has been taken over
26	Principal amount	The credit amount fixed in the contract (if any)
27	User ID complement application	The employee ID of the employee performing the complementing application call
28	Last action user ID	The employee ID of the employee performing the last action on the application
29	Last queue name	The name of the queue in which the last action is performed
30	Reached handling leads	Binary indication is it reached queue handling leads
31	Reached complementing application	Binary indication is it reached queue complementing application
32	Reached calling after offer	Binary indication is it reached queue calling after offer
33	Reached assessing application	Binary indication is it reached queue assessing application
34	Reached incomplete application	Binary indication is it reached queue incomplete application

App. A table 2: Overview application characteristics

Test	Relation	KPI	First sample	Second sample	X1	X2	N1	N2	P1	P2	Paverage	Z0	$p_1 \neq p_2$	$p_1 > p_2$	$p_1 < p_2$
1	Gender	ACT	1	2	655	742	1512	1773	0.433	0.418	0.4253	0.8495	0	0	0
2	Gender	ACT	3	4	3309	3536	6684	7369	0.495	0.48	0.4871	1.8021	0	1	0
3	Gender	GAP	1	2	176	171	634	704	0.278	0.243	0.2593	1.4463	0	0	0
4	Gender	GAP	3	4	739	786	3155	3357	0.234	0.234	0.2342	0.0089	0	0	0
5	Gender	OTH	1	2	296	329	598	674	0.495	0.488	0.4914	0.244	0	0	0
6	Gender	OTH	3	4	1346	1425	2959	3139	0.455	0.454	0.4544	0.0719	0	0	0
7	Gender	CRC	1	2	4	5	8	9	0.5	0.556	0.5294	0	0	0	0
8	Gender	CRC	3	4	12	17	26	28	0.462	0.607	0.537	0	0	0	0
9	Comp A	ACT	1	2	246	1621	661	3328	0.372	0.487	0.468	-5.408	1	0	1
10	Comp A	ACT	3	4	931	5324	2166	10535	0.43	0.505	0.4925	-6.404	1	0	1
11	Comp A	ACT	5	6	8	112	93	555	0.086	0.202	0.1852	-2.66	1	0	1
12	Comp A	GAP	1	2	24	286	177	1296	0.136	0.221	0.2105	-2.605	1	0	1
13	Comp A	GAP	3	4	207	1322	931	5326	0.222	0.248	0.2444	-1.695	0	0	1
14	Comp A	GAP	5	6	2	31	8	112	0.25	0.277	0.275	0	0	0	0
15	Comp A	OTH	1	2	2	37	156	841	0.013	0.044	0.0391	-1.845	0	0	1
16	Comp A	OTH	3	4	481	2787	931	5322	0.517	0.524	0.5226	-0.396	0	0	0
17	Comp A	OTH	5	6	8	81	8	112	1	0.723	0.7417	0	0	0	0
18	Comp A	CRC	1	2	1	2	2	6	0.5	0.333	0.375	0	0	0	0
19	Comp A	CRC	3	4	9	26	17	45	0.529	0.578	0.5645	0	0	0	0
20	Comp A	CRC	5	6	0	0	0	1	0	0	0	0	0	0	0
21	Comp B	ACT	1	2	28	196	367	1564	0.076	0.125	0.116	-2.64	1	0	1
22	Comp B	ACT	3	4	1088	6536	2366	12164	0.46	0.537	0.5247	-6.905	1	0	1
23	Comp B	GAP	1	2	0	0	28	197	0	0	0	0	0	0	0
24	Comp B	GAP	3	4	233	1639	1088	6537	0.214	0.251	0.2455	-2.595	1	0	1
25	Comp B	OTH	1	2	10	86	26	190	0.385	0.453	0.4444	0	0	0	0
26	Comp B	OTH	3	4	481	2819	1069	6085	0.45	0.463	0.4613	-0.806	0	0	0
27	Comp B	CRC	1	2	0	0	0	0	0	0	0	0	0	0	0
28	Comp B	CRC	3	4	10	28	19	52	0.526	0.538	0.5352	0	0	0	0
29	Comp C	ACT	1	2	1166	7005	2888	14310	0.404	0.49	0.4751	-8.42	1	0	1
30	Comp C	ACT	3	4	19	52	32	108	0.594	0.481	0.5071	1.1157	0	0	0
31	Comp C	GAP	1	2	223	1609	1097	6682	0.203	0.241	0.2355	-2.714	1	0	1
32	Comp C	GAP	3	4	10	30	19	52	0.526	0.577	0.5634	0	0	0	0
33	Comp C	OTH	1	2	481	2875	1076	6228	0.447	0.462	0.4595	-0.887	0	0	0
34	Comp C	OTH	3	4	10	30	19	47	0.526	0.638	0.6061	0	0	0	0
35	Comp C	CRC	1	2	0	0	0	0	0	0	0	0	0	0	0
36	Comp C	CRC	3	4	10	28	19	52	0.526	0.538	0.5352	0	0	0	0
37	Part	ACT	1	2	1348	637	3224	1719	0.418	0.371	0.4016	3.2479	1	1	0
38	Part	GAP	1	2	246	138	1229	617	0.2	0.224	0.208	-1.173	0	0	0
39	Part	OTH	1	2	585	299	1199	605	0.488	0.494	0.49	-0.253	0	0	0
40	Part	CRC	1	2	8	3	13	7	0.615	0.429	0.55	0	0	0	0
41	Part	ACT	3	4	248	191	705	368	0.352	0.519	0.4091	-5.289	1	0	1

42	Part	ACT	5	6	417	430	1040	1025	0.401	0.42	0.4102	-0.857	0	0	0
43	Part	ACT	7	8	640	16	1351	128	0.474	0.125	0.4435	7.5901	1	1	0
44	Part	ACT	Mo|Ev	Ev|Ev	304	16	574	326	0.53	0.049	0.3556	14.475	1	1	0
45	Part	ACT	Mi|Ev	Ev|Ev	336	16	777	326	0.432	0.049	0.3191	12.463	1	1	0
46	Part	ACT	Mo|Ev	Mi|Ev	304	336	574	777	0.53	0.432	0.4737	3.5364	1	1	0
47	Part	ACT	MoReg	MiReg	2210	3323	4587	6998	0.482	0.475	0.4776	0.732	0	0	0
48	Part	ACT	MoReg	EvReg	2210	2573	4587	5344	0.482	0.481	0.4816	0.032	0	0	0
49	Part	ACT	MoReg	NiReg	2210	136	4587	409	0.482	0.333	0.4696	5.7962	1	1	0
50	Part	ACT	MiReg	EvReg	3323	2573	6998	5344	0.475	0.481	0.4777	-0.73	0	0	0
51	Part	ACT	MiReg	NiReg	3323	136	6998	409	0.475	0.333	0.467	5.608	1	1	0
52	Part	ACT	EvReg	NiReg	2573	136	5344	409	0.481	0.333	0.4709	5.8167	1	1	0
53	Queue	ACT	1	2	3011	4765	6500	9894	0.463	0.482	0.4743	-2.305	1	0	1
54	Queue	GAP	1	2	667	1112	2850	4556	0.234	0.244	0.2402	-0.984	0	0	0
55	Queue	OTH	1	2	1197	1979	2688	4254	0.445	0.465	0.4575	-1.621	0	0	0
56	Queue	CRC	1	2	16	21	35	34	0.457	0.618	0.5362	-1.337	0	0	0
57	Account	ACT	1	2	393	198	1053	588	0.373	0.337	0.3601	1.4763	0	0	0

App. A table 3: Test inputs and results

9. Bibliography

[Vanderfeesten et al, 2011]
I.T.P. Vanderfeesten, H.A. Reijers and W.M.P. van der Aalst. Product-based workflow support. In *Information systems*, volume 36, pages 517-535. Elsevier. 2011.

[Panwalkar et al, 1977]
S.S. Panwalkar and W. Iskander. A Survey of Scheduling Rules. In *Operations Research*, volume 25, pages 45–61. 1977.

[Silver et al, 1998]
E.A. Silver, D.F. Pyke, and R. Peterson. Inventory Management and Production Planning and Scheduling. John Wiley and Sons, Hoboken, NJ, 1998.

[Parmenter, 2007]
D. Parmenter. Key performance indicators. Developing, implementing and using winning KPIs. John Wiley & Sons, Inc. 2007.

[Schafer et al, 2001]
J.B. Schafer, J.A. Konstan and J. Riedl. E-Commerce Recommendation Applications. In *Data mining and Knowledge discovery*, volume 5, pages 115-153. Kluwer Academic Publishers. 2001

[Brand et al, 1995]
N. Brand and H. van der Kolk. Workflow analysis and design. Kluwer Bedrijfswetenschappen, Deventer. 1995 [in Dutch].

[Park et al, 2012]
D.H. Park, H.K. Kim, I.Y. Choi and J.K. Kim. A literature review and classification of recommender systems research. In *Expert Systems with Applications*. In press. Elsevier. 2012.

[Montgomery et al, 2003]
D.C. Montgomery, G.C. Runger. Applied statistics and probability for engineers, third edition. John Wiley & Sons, Inc. 2003.