MASTER

Characterization of splits and joins of less structured processes

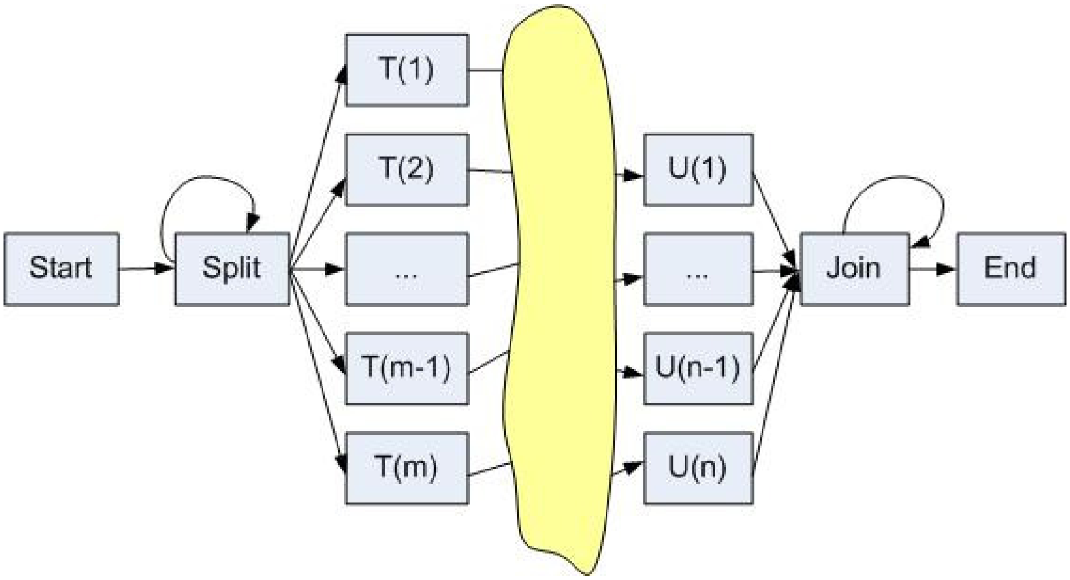
Theunissen, J.G.A.

Award date:
2007

Master Thesis

‘’Characterization of Splits and Joins of less structured Processes’’

by ing. Jo Theunissen M.Ed.



TU/e technische universiteit eindhoven

Master thesis:
“Characterization of the Splits and Joins of less structured Processes”

Eindhoven, August 2007.

Evening Student Industrial Engineering and Management Science:
J.G.A. Theunissen
Generation: 2002.

Supervisors:
dr. Ton Weijters, TU/e, Information Systems
prof.dr. Rob Kusters, TU/e, Information Systems

For the practical examples, data from the following hospitals is used:
** Hospital “Catharina” at Eindhoven*
** AMC at Amsterdam*
** Università degli Studi Pavia, Italy*
** IRCCS C. Mondino, Pavia, Italy*
** IRCCS Humanitas, Rozanno, Italy.*

I want to thank these organizations for the availability of data.

TU/e technische universiteit eindhoven

Dedicated to my dear mother

Riet Hommen
(8th April 1944 – 19th July 2001, Kerkrade)

Preface

This report is about my master thesis, which is a completion of my study Industrial Engineering and Management Science at the Eindhoven University of Technology. The master thesis project took place from October 2006 until August 2007 at the department Technology Management at the same university. The project is about the managing of flexible structure of process into the process mining tool *ProM*, which is developed at home.

The master thesis is the final part of the study. In the last 5 years I had experienced with great pleasure the academic skills and philosophy in many courses. The university teachers and researchers stimulated me in realizing good results. This study was for me the last gap of my education career over the last 20 years. The last part of the study was also an additional value of my practical experience in the field of Informatics.

The project, which included the master thesis, has been interesting and challenging all the time for me. Several types of problems had to be lightened and solved. Especially during the implementation of the new algorithm the researchers of the sub department *Information Systems (IS)* were a great support for me. Their support was a complementary value of a successful finish of this project.

Another nice matter of minor importance is the return to Eindhoven. Since the summer of 2001 I'm living not far from the place I grew up. Now I have also finished the study at the same University as my brother. The last part of the research study also had some relation with our family history. During this research phase I analyzed the data set from the hospital "Catharina", where my mother gave birth to my brother in 1970. In Dutch we say: "*De cirkel is rond*".

From my personal point of view I first would like to thank my deceased mother Riet Hommen. Her energy and appearance in the past made it all possible to finish my academic study. She will always remain in my mind.

Leefste Mam,

Bedankt vuur alles wats doe vuur miech jedoa has. Durch diech han iech dat kanne volbringe. Dienne wónsj is oeskómme. Doe has miech noeëts in sjitich jelosse. Dieng wöad: "Jef noeëts d'r mód óp, óch al is 't aaf en tsouw nit jemekkelieg.", hant noen zieng plaatsj kréje in dit werk.

Danke.

Also thanks to my old study advisor of the study Industrial Engineering and Management Science, Piet van Drunen. He was surprised in positive way that I picked up the study in 2002, after almost 10 years interruption of this study. I started the study on my brother's advice during my military service time at the Head Quarters Oirschot and Seedorf (Germany). Except playing tennis and spinning after the lunch time, we had a lot of interesting talks about the development in education world.

The last academic year, I supported university teacher Simme Douwe Flapper with his masterclass program for students from the highschool. It was a nice opportunity and experience for me as teacher thanks Simme Douwe. From Leon Osinski, working at the library of Technology Management, I learned to work with the software tool Reference Manager, which makes easy a reference list in an assignment. Thanks Leon I had also the possibility to play with his indoor soccer team, which I enjoyed during the lunch time. The last year, I joined the badminton group, Nataliya Mulyar, Lusine Hakobyan, Minseok Song and Ronny Mans. Thanks for the nice time.

Many thanks also to the secretaries of the subdepartment Information Systems, Ineke Withagen and Ada Rijnberg, for their helpfulness and also for the nice discussions that we had while pickung up a cup of coffee. Also thanks to university teacher Jan Goossenaerts. The discussions with him about still open research domains of Petri Nets, was always quite interesting for me.

A special thanks to my mother's aunt Trie Bakker, her daughter Ria Bakker, my cousin Mark Wolter and their relatives, my neighbours Hélène and John Benita, Joost den Hollander and Helen Aangenend and my best (study)friend Jochen Schellekens, who supported me in their own way during the last part of the study.

Finally, I would like to thank my university supervisors Ton Weijters and Rob Kusters for their (philosophic) support and objective advice. Especially the many conversations about the research project with Ton brought interesting and learning points for me. Also the talks during the walks in the afternoon around the campus were nice. I want also to thank Ton for the facilities at the TU/e, which I had during the research project. Without these things, it would for me almost impossible to fulfill successfully the study.

FINIS CORONAT OPUS

Jo Theunissen jr.
Eindhoven, 24th August 2007.

Abstract

The master project is about the characterization of less structured processes. In real life some processes are less structured, e.g. the medical treatments in a hospital are characterized by highly complex and extremely flexible patient care processes. This type of processes has its interest of the research group Business Process Management within Information Systems. This subdepartment is part of TU/e's Technology Management. Less structured processes have also its interests in the business world.

The research project belongs to the research domain of process mining. Process mining is a very young research domain. It started in the mid 90ties. The goal of this research domain is to extract information from event logs, which are stored by some information systems. These files illustrate the characteristics of the processes in an organization. However, it is difficult to analyze the processes by the files. Therefore, we use a process mining tool for analyzing of the processes. The process mining tool, called *ProM*, contains various process mining techniques. Some of these techniques show a process model as a result, which is based on the information in the logfile.

Less structured processes are characterized by the forks in the process model. The splits and joins of less structured processes have the characteristics of a "vague" OR. This is type is vague, because the behaviour of the processes are less clear and more complex comparing to the behaviour of structured processes. The structured processes are characterized by two types, i.e. XOR and AND. For instance, the XOR-split starts up only activity and the AND-split enables all activities. The OR-split has both and all other remaining combinations.

Some traditional process modeling languages have the OR at home. Process modeling languages are characterized by their semantics. The semantics can be distinguished in two types, i.e. local and non-local. But the semantics of these languages have problems with the representation of less structured processes. The process modeling languages with local semantics like BPEL are very complex. Non local semantics like EPC and YAWL has unclear definitions for the OR. This all leads to an adjustment of the original research objective. The research objective is the development of a knowledge discovery tool instead of a process modeling language. This new tool uses as starting point a general process model, called dependency graph. The dependency graph is part of the Heuristics Miner algorithm, which is a general process mining technique. This algorithm is based on a frequency based metric.

The framework of this research project is characterized by the forks including loops and its directly connected activities in the process model. For the analysis of the splits and joins in the process model, we use four methods, i.e. frequencies of individual activities, pattern frequencies, binary relation between activities, and grouping of activities. The four methods are based on the two dimensional binary matrix, which will be set up after the mining of the dependency graph. The columns of this matrix are the activities which are directly connected to the fork and the rows are the "baskets". The "baskets" are originated from the market basket analysis. Each basket represents the presence of an activity in the traces of the logfile by a Boolean expression.

The four methods have their own point of view to the forks. The first method gives insight in the behaviour of the individual activities. It gives also the first impression of the type of the split/join, i.e. XOR, OR, and AND. The second method determines the pattern frequencies. It illustrates the combinations of activities and their frequency. The pattern frequency supports the findings of the type of fork, but it gives also insight in the complexity of the split/join. The number of combinations of activities illustrates the heterogeneity of the fork. The third method shows the degree of dependency between two activities in relation to their split/join. Two

measures are selected for the analysis of the dependency between activities, i.e. correlation and IS. The last method is based on the apriori algorithm, which is based on the mining of association rules. This mining technique tries to find sets of activities that frequently took place together, so that from the presence of certain activities can infer that certain other presence activities will also be present.

The four methods are implemented as (statistical) tables in the new tool, called *SplitJoinIndicator*. The tables are added in a submenu in the Heuristics Miner. Also part of the knowledge discovery tool is the representation of the process model. The representation of the process model is the dependency graph including the probabilities of the individual activities. This way of representation gives already the first impression of the behaviour of the forks in the process model.

SplitJoinIndicator is validated with two types of data, i.e. internal and external. The internal data are the artificial logfiles, which are generated by the process modeling language CPN Tools. Two artificial logfiles are used for extensively testing the new tool. One logfile has the characteristics of complete structured processes and the other file has the characteristics of less structured. It is necessary to know that the new tool works correctly for both types of processes. Almost everything was a priori known about the two internal logfiles. Subsequently starts the validation with external data. The external data are two logfiles from the healthcare. The files from the hospital have typical medical characteristics, which are recorded in the files. In this situation, only the typical medical characteristics were already known.

SplitJoinIndicator has been successfully validated for all four files. The new tool discovered some more interesting points especially from the hospital data, which were unknown in advance. The knowledge from the four tables can be applied for further (statistical) analysis. Another strong point of this tool is the less sensibility of some noise in the logfile. In this situation, the statistical information in the four tables is still stable.

SplitJoinIndicator is one of the first process mining tools, which can deal with the less structured processes. It is still an open field in process mining and has a lot of opportunities for the coming years. The new tool has its own point to this topic, i.e. statistics. It is a challenge to find more direction of solutions in this complex, but much more challenging topic for process mining.

Keywords: Process Mining, (Statistical) Data Mining, Heuristics Miner, Splits and Joins, Aproiri Algorithm, Association Rule Mining, Knowledge Discovery, Healthcare.

Content

1. Introduction

This master thesis is the result of the graduation project for Master of Science in Industrial Engineering at Eindhoven University of Technology (TU/e). The research project is carried out at the group Information Systems of the department Technology Management. The subject of this Master thesis's is to study how a certain type of processes can be mined and analyzed e.g. in the healthcare domain.

The master thesis contains six chapters. Every chapter starts with an introduction about a certain topic and ends with a conclusion. The used definitions and formulations in this master thesis are explained by examples.

The first chapter of this assignment gives insight in what we mean by the less structured processes and their characteristics. Also part of this chapter is process mining (PM) in general, the problems of analyzing less structured processes with the current PM techniques, and the orginally formulated research objective. The second chapter describes the analysis of the research project. This analysis is about the traditional process modeling languages (PML). The third chapter is about several methods, which might be used for analyzing less structured processes. The fourth chapter describes the new knowledge discovery tool. The validation of the new tool is the penultimate chapter of this master thesis. This validation takes place with files from the healthcare and from the TU/e. The last chapter is about the conclusions and recommendations.

This assignment contains also appendices including a list of abbreviations and used mathematical notations. It contains also some background information about certain topics and statistical analysis of used examples.

The first chapter contains seven sections. It starts with the characteristics of different type of processes, followed by the characteristics of the most interesting and complex split and join in our research project, e.g. the so-called OR. The third section is about the environment of our research project, namely PM. Subsequently, starts the description of a particularly type of files, which can be generated by some information systems (IS). The fifth section describes the problems with the mining of less structured processes by the current PM techniques, which are developed in the PM tool, called *ProM*. In section six, we formulate the original research question. This chapter ends with the conclusions.

1.1 The Characteristics of different Types of Processes

This master thesis is about a certain type of processes. First, we have to specify the term processes in this master thesis. Processes consist of a number of logical units of work (called tasks) that need to be carried out by a person or machine. The performance of a single process by a person or machine is called an activity, which is related to a specific case. The activities in a business environment form a network. The network of activities has rules. These rules determine the (partial) order in which the tasks should be performed. These networks are defined as a workflow (pages 3-28 of [11]).

The processes, which are part of the workflow, might be distinguished in two types, i.e. the structured and the less structured. The "structured" processes will be performed in a certain order. The less structured processes are characterized by less ordering of performance of tasks[1]. This type of processes has more freedom in ordering to perform the tasks and is more complex. Table 1 shows some more characteristics of both types of processes.

Characteristics	***Structured processes***	***Less structured processes***
Recognition of patterns in processes	*easy*	*difficult*
Dependency between activities	*Clear*	*less clear*
Activation of fork	*a priori known*	*Uncertain*
Insight of processes	*straightforward*	*more complex*

Table 1: The characteristics of the two types of processes in general.

Some processes in practice are characterized by some flexibility. Examples of less structured processes in practice are the processes in a hospital. Various activities will take place after the intake of a patient based on medical procedures. Some of the activities are performed as a combination most of the time. Others will be canceled after a time or take place at a later stage.

Our interest is the process mining of these less structured processes. It is interesting to get more insight in this topic. These processes in our research work can be distinguished by two ways, i.e. the properties of the event log, and the behaviour of the forks in the process model. The logs, contains a lot of information about the processes. This might be not complete for the recognition of less structured processes. The reason for this is that some of the processes have a low occurrence in the data and others are missing in the logfile. Also, it is possible that certain activities will be canceled after a while, which is based on results of other factors. As already mentioned structured processes keeps their patterns in logs. This is easy to recognize even in partitions of the log. For the less structured processes the patterns are more difficult to recognize in these partitions. The partitions of the log will be mostly different for less structured processes. This hardly illustrates similarity and makes it difficult to analyze the behaviour of this type of processes. Therefore, we use the forks in the process model for analyzing the characteristics of the processes. The forks in the process model indicate the way of performance of the activities in a process model. The process model is based on the given event logs, which has the behaviour of the type of processes. Next section describes the forks and their types.

1.2 Types of Forks in Workflow Processes

As already mentioned, the best way to characterize the workflow of less structured processes is by their forks. A fork is connected to at least two activities (including it self). A fork is characterized by at least two input or output connections (arcs).

There are two forks, i.e. splits and joins. The split is the activity, which enables its direct connected activities (including itself if presence). The join is the activity, which can only be enabled by its directly connected activities (including itself if presence). A fork might have both types. Let illustrate this according a general example.

[1] Performing a task is an activity.

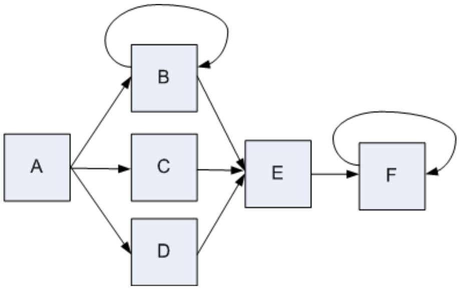


Figure 1: General example of a process model.

Figure 1 shows a process model, which contains 6 activities, namely *A* to *F*. It starts with activity *A*. This activity is a split. It enables the activities *B*, *C*, and/or *D*. Activity *B* has a loop of length one and can be active after finishing of activities *A*. Activity *B* contains both types of forks (i.e. split and join), because it is connected to two or more activities from both sides (i.e. input and output). Activities *B*, *C* and/or *D* enable activity *E*, which is a join. The process ends with activity *F*. The last activity is also a join, because it can be started by activity *E* and itself.

The splits and joins in the process model illustrate the behaviour of the (less) structured processes. Next subsection describes in more detail about the characteristics of the splits and joins.

1.2.1 The Characteristics of Splits and Joins in Workflow Processes

We distinguish three types of splits and joins, e.g. the exclusive OR (XOR), the OR and the AND. The structured processes contain splits and joins of the type XOR and/or AND. However, the less structured processes contain *at least one* split and join of the type OR. The characteristics of the XOR and AND are well known and described in many scientific papers like [14]. These types are relative easy to recognize. But the properties of the OR are different and more complex.

Let's describe the three types of the forks in more detail. The OR is a set name for all types of splits and joins. It includes the type XOR, the OR, and AND. The XOR and AND are thus also part of the OR. The OR has more flexibility than the XOR and AND. However, the types XOR and AND have a unique characteristic. They allow *only one* situation. The XOR is the type of split, enabling only one branch. In the case of a join only one of the branches must be active for enabling the join. The AND split enables all branches at the time. Also all branches just before the join must be active for enabling in case of an AND-join. In this master thesis, we mean with the OR, the combinations XOR, the AND and the remaining situations. All combinations are possible. In other words, the OR-split or –join is a control flow construct that sometimes behaves likes an AND-split or –join and sometimes like an XOR-split or –join based on the current context. Otherwise, it behaves between the characteristics of the XOR and the AND. Many variants and interpretations of the OR-split or –join have been proposed in the literature [92,93,94]. Therefore is the OR-split and -join a complex and "vague" pattern. Let's illustrate the formulation of the "vague" OR according to an example in the following subsection.

1.2.2 Example

We take an example from the healthcare. In the medical world, the processes are more complex and less insightly. The first activity is the intake of patient. In this stage, data and some first medical data will be noticed. After the first intake (split) of a patient several treatments can be started up similtaneously or not. These treatments might be for instance pre-assessment, thorax, pathology, CT, blood tests, cardiology, ECG, MRI. Figure 2 shows the first part of the medical treatments of a patient.

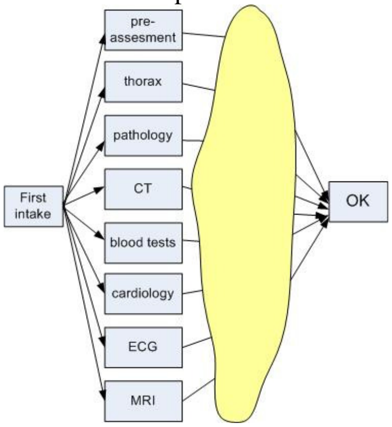


Figure 2: Example of a "vague" OR.

There isn't apriori knowledge which of these treatments will begin, because it depends on the first medical information and health situation of the patient at that moment. In our situation, there are 255 combinations of treatments possible (=2^8-1) based on these 8 treatments after the first intake (split). For instance, after the first intake of a patient starts the treatments thorax, pre-assessment and CT. This is one of all possible combination of treatments after the split (first intake). The number of possible treatments (say *n)* directly after the first intake increases the number of combinations (i.e. 2^n-*1*) very fast. In the situation of many possible combinations of treatments, it is difficult to discover any structure of the split. It makes more complex, if the size of the event log is small. This illustrates the characteristics of a "vague" OR.

The diagnosis of these selected treatments has influence for the following treatments of the patient, which leads to the decision of a possible OK of the patient. The OK is the last activity in this process model. The treatment OK is in this situation a join, which has the same characteristics as the split (first intake). These characteristics can play for both types of forks (i.e. split and join).

The analysis of the characteristics of the processes falls within the research domain of process mining (PM). Next section describes PM.

1.3 Process Mining

PM is a very young research domain. It started in the mid 90ties with new approaches to construct process models, which are proposed by Cook and Wolf [26]. One of the first applications of PM, which was related to the workflow domain, was presented by Agrawal in 1998 [27].

PM is applicable to a wide range of systems, where may be pure IS (e.g. ERP systems) or where the hardware plays a more prominent role (e.g. embedded systems). The only requirement for PM is that the systems can store information about activities which took place in this business envonriment, the so-called event logs.

PM enables the extraction of information from event logs. For example, the audit trails of a workflow management system (WfMS[2]) or the transaction logs of an enterprise resource planning (ERP) system can be used to discover models describing processes, organizations, and products [14]. The information in these logs represents a great wealth of untapped data, and until recently, it was rarely used to analyze the underlying processes.

The goal of PM is to extract knowledge from event logs. PM techniques can be deployed to use this information, for example to extract unexpected and useful knowledge about the process and then modify decision-making as appropriate for future instances (*process discovering*). Consider for example processes supported by an ERP system like SAP. Such a system logs all transactions but does not completely enforce a specific way of working. In such an environment, process mining could be used to gain insight in the actual process. Another example would be the flow of patients in a hospital. Note that in such an environment all activities are logged but information about the underlying process is typically missing. In this context it is important to stress that management information systems (MIS) typically provide information about key performance indicators (KPI) like resource utilization, throughput times, and service levels but not about the underlying business processes (e.g. causal relations, ordering of activities, etc.)

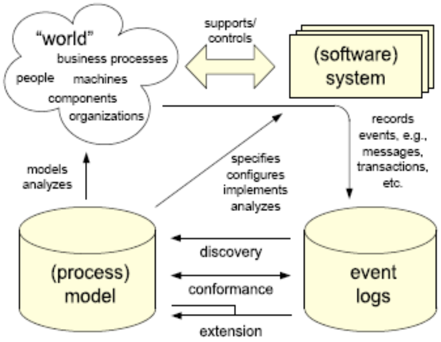


Figure 3: The three types of PM and the realition of PM and its environment.

The idea of PM is to discover, monitoring and improve real processes by extracting knowledge from event logs. We consider three basic types of PM (see Figure 3) e.g.:

- **Discovery:** There is no a-priori model, i.e. based on an event log some model is constructed. Using a PM technique a process model can be discovered based on events in the log.
- **Conformance:** There is an a-priori model. This model is used to check if reality conforms to the model.
- **Extension:** There is an a-priori model. This model is extended with a new aspect or perspective. The goal is to enrich the model with the data in the event log.

Traditionally, PM has been focusing on discovery, i.e., deriving information about the original process model, the organizational context, and execution properties from enactment logs. However, PM is not only focused on process models and recent PM techniques are more based on other perspectives, e.g. the organizational perspective, performance perspective or the data perspective. For example, PM can be used to monitor coordination in ERP systems. Some of the coordination is done by humans while other coordination tasks are done by software. As

[2] WfMS is a generic software tool which allows for definition, execution, registration and control of workflows (WF) [11,54].

indicated, similar interaction patterns occur at the level of software components, business processes, and organizations.

Conformance checking compares an a-priori model with with the observed behaviour as recorded in the event log. For example, PM can be used to identify deviations between the discovered process model and the descriptive or prescriptive process model. Consider for example the reference models in SAP. These models describe how the system should be used. Using PM it is possible to verify whether this is the case. In fact, PM could be used to compare different departments/organizations using the same ERP system.

There are different ways to extend a given process model with additional perspectives based on event logs, e.g. decision mining, performance analysis, and user profiling. For example, the process model will be extended with performance data for the goal of general optimization of business processes. This information can help to identify redundant activities and operations that require restructuring. It also enables monitoring of possible deviations from normal processing.

PM has to deal that most systems have limited information about what is actually happening. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what actually happens. Only a concise assessment of reality, which PM strives to deliver, can help in verifying process models, and ultimately be used in system or process redesign efforts. Therefore, PM can be performed at many levels and situations. For PM to be effective, the information captured on the transactions logs have a certain makeup, namely (i) each event refers to an *activity* (i.e. well-defined step in the process), (ii) each event refers to a *case* (i.e. process instance), (iii) each event has possible a *performer* also referred to as *originator* (the actor executing or initiating the activity), and (iv) events have possible a *timestamp* and are totally ordered. Next section is about this type of logs.

1.4 Event Log

The event log is the starting point of the mining. We distinguish three different mining perspectives: (1) the process perspective, (2) the organizational perspective and (3) the case perspective. The process perspective focuses on the control flow, i.e. the ordering of activities. The goal of this mining perspective is to discover a good characterization of all possible paths, expressed in terms of, for instance, a Petri Net [16]. The organizational perspective focuses on the originator field, i.e. which performers are involved in performing the activities and how they are related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show relations between individual performers (i.e build a social network [17]). The case perspective focuses on properties of cases. Cases can be characterized by their path in the process or by the originators working on a case. In other words, the process perspective is concerned with the *"How?"* question, the organizational perspective is concerned with the *"Who?"* question and the case perspective with the *"What?"* question. This master thesis will mainly focus on the process perspective, i.e. the ordering of the activities. Next subsection illustrates the idea of process discovery.

1.4.1 The Idea of Process Discovery by an event Log

Consider the process log from Table 2, which has been derivated from the event log from the hospital "AMC" [55]. Table 2 illustrates the first 20 events of the log. The case ids are the patients, which are numbered (e.g. *patient_1, patient_2,* etc.) According to this table, there are 10 patients. Apparently patientnr 3 and 9 are missing in the log. The number of different activities is 8 (e.g. *eerste consult AMC*, *pre-assessment*, etc.). The number of originators is 6 (e.g. *doctor*, *anesthesist*, etc.) Each event in the event log refers to an activity, a case, an originator and a time stamp. The log is ordered on time basis. In addition to the information shown in this table, some event logs contain more information on the case itself, i.e. data elements referring to properties of the case (i.e. in an hospital IS, age, sex, diagnose, etc. of a patient).

case id	***activity id***	***originator***	***time stamp***
Patient_1	*eerste consult AMC*	*doctor*	*2003-07-21T01:00*
Patient_2	*eerste consult AMC*	*doctor*	*2003-07-28T01:00*
Patient_1	*pre-assessment*	*anesthesist*	*2003-07-30T01:00*
Patient_4	*pre-assessment*	*doctor*	*2003-08-06T01:01*
Patient_4	*pathologie AMC*	*patholoog*	*2003-08-06T01:03*
Patient_5	*Thorax*	*radioloog*	*2003-08-11T01:02*
Patient_6	*eerste consult AMC*	*doctor*	*2003-08-12T01:00*
Patient_6	*Thorax*	*verpleegkundige*	*2003-08-12T01:02*
Patient_7	*eerste consult AMC*	*doctor*	*2003-08-13T01:00*
Patient_5	*MRI*	*radioloog-mri*	*2003-08-13T01:01*
Patient_2	*pre-assessment*	*doctor*	*2003-08-14T01:00*
Patient_1	*lich. onderzoek onder narcose*	*doctor*	*2003-08-15T01:00*
Patient_8	*pre-assessment*	*doctor*	*2003-08-18T01:02*
Patient_8	*ECG*	*verpleegkundige*	*2003-08-18T01:04*
Patient_6	*CT*	*radioloog-ct*	*2003-08-20T01:00*
Patient_7	*pre-assessment*	*anesthesist*	*2003-08-20T01:00*
Patient_10	*pre-assessment*	*anesthesist*	*2003-08-20T01:01*
Patient_11	*eerste consult AMC*	*doctor*	*2003-08-21T01:00*
Patient_12	*eerste consult AMC*	*doctor*	*2003-08-25T01:00*
Patient_11	*pre-assessment*	*anesthesist*	*2003-08-25T01:01*

Table 2: Part of an event log.

As an example for the first perspective considers Table 2, which contains the following: *Patient_1* has event trace *eerste consult AMC*, *pre-assessment*, and *lichamelijk onderzoek onder narcose*. *Patient_2* has event trace *eerste consult AMC*, *pre-assessment*. *Patient_4* has *pre-assessment*, *pathologie AMC* etc.

After analyzing all cases in the event log, we can extract information about the process assuming some notion of completeness and no noise. Each case starts almost with the execution of *eerste consult* and ends with the *OK*. After *eerste consult*, the activities *pre-assessment*, *thorax* and *pathologie AMC* can be started. In other words, after activity *eerste consult*, there is a choice in the process and one or more of these activities (*pre-assessment, thorax,* and *pathologie AMC*) can be executed next. When *eerste consult* is finished, any combination of *pre-assessment*, *thorax*, and *pathologie AMC* can be executed as a combination in any order. The time stamps of these activities confirm these assumptions.

According to the event log the activities *pre-assessment*, *thorax* and *pathologie AMC* are properly connected by a vague "OR"-split. The activity *OK* might be an OR-join. The activities *pre-assessment, MRI*, *thorax* and *pathologie AMC* can be executed in combination before activity *OK*. Further analysis is necessary to get more certainty.

The event logs can be analyzed by the PM tool, called *ProM*[3]. *ProM* framework integrates the functionality of several existing PM tools, both commercial and public (like Wolflan, Aris Toolset, etc.) and provides many additional PM algorithms, called plug-ins. It is ongoing toolset, which has been developed at the TU/e and already proved in practice like the hospitals "Catharina", "AMC" and the analysis of Workflow Management Systems (WfMS) of Dutch municipalities. Therefore, it is a necessary that the event logs have a certain format for analyzing in the PM tool *ProM*, e.g. MXML[4] or XML[5] (see appendices A.6 and A.7). Next section is about analyzing of processes with *ProM*. In particular, the experiences of analyzing of less structured processes, which has our attention.

1.5 Problems with current Process Mining Techniques

The PM tool, called *ProM*, contains several PM techniques which can deal with event logs of certain type. The current PM techniques have problems with event logs, which are characterized by less structured processes. The Heuristics Miner algorithm (HM) is one of the PM techniques, which can be used in general for analyzing of different type of event logs, including noise[6] [14]. This PM technique develops a process model, based on a frequency based metric. It is a heuristic technique and it represents the process models on two ways, namely the dependency graph (DG) and the Heuristics Net (HN). Let's start with the description of the first representation in more detail, i.e. DG.

The DG

The DG is a general representation of a process model. It is a modeling technique, which illustrates the relation between activities. The DG shows besides the relation between activities, also the number of parsing and the number of activities in the event log. The number of occurrence of a certain activity in the event log is noticed in the DG. The dependency measure and the number of parsing are written above the arcs between the linked activities. Figure 4 on page 18 shows the DG of the first medical consult for womb cancer at AMC [55].

The dependency measure (value varies between -1 and 1) is based on the frequency directly between the two involved activities. It indicates how certain we are that there is a dependency relation between two activities. A high value of the dependency measure (close to 1) means that we are very sure that there is a dependency relation between the connected activities.

The number of parsing is based on the number of occurring between two connected activities. These activities must appear in sequence (one directly following the other) in a trace from the event log, e.g. activities *A* and *B* must appear in the logfile as *…AB…*. This method counts the number of occuring *AB* in the event log.

[3] The latest version (4.1) of the *ProM* framework was realized at 15th April 2007. This version is freely available at the website http://www.processmining.org. (See also appendix A.4).

[4] MXML-format is the abbreviation for **M**ining e**X**ensible **M**arkup **L**anguage. It is the format which is necessary for *ProM* (see also appendix).

[5] XML is a simple but very flexible text format derived from *Standard Generalized Markup Language (SGML)*, and has been playing an increasingly important role in the exchange of wide variety of data over the Web [57,61].

[6] Information, which is not correct recorded in the event log.

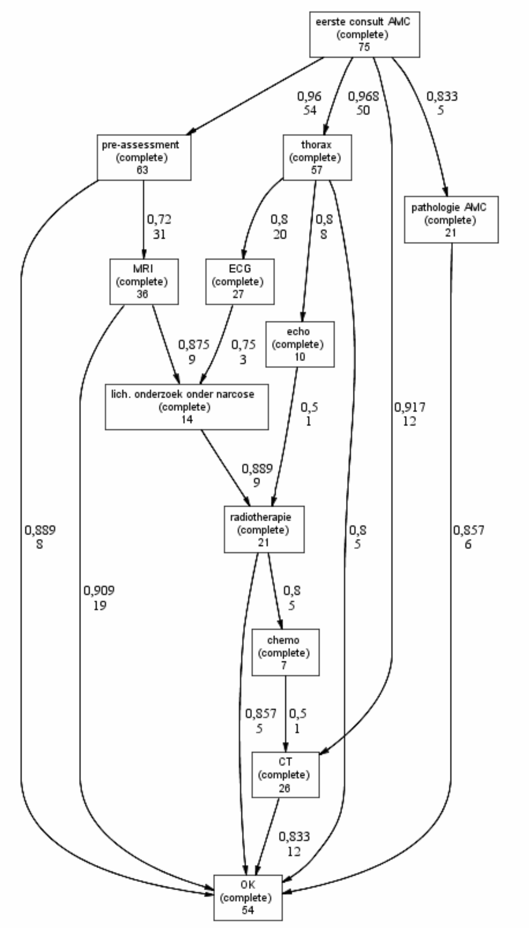


Figure 4: The DG of the first medical consult for womb cancer at AMC.

For instance, the number of occurring of activity *eerste consult AMC* and *thorax* are resp. 75 and 57 times in the event log. The number of parsing between both activities is 50 times and, the dependency measure is 0.968, which indicates a high reliability of a dependency relation between *eerste consult AMC* and *thorax*.

The number of parsing and the dependency measure gives good insight, if the split/join is the type of XOR or AND. The number of occurring indirectly can be low for the split/join of the "vague" OR, while the dependency measure between the two activities is high. These measures gives unsatisfied information about the split/join of the type OR. Therefore, it is necessary to develop new measures, which give more insight in the characteristics of the three types of splits/join in a process model.

Next subsection is the second representation of the HM.

The HN

The HN is an extension of the DG. It shows also the type of splits and joins. The HN illustrates the way of connection between the activities. Figure 5 on page 19 shows the HN of the first medical consult for womb cancer at AMC [55]. For instance, the activities *eerste consult AMC* and *thorax* are connected by an AND.

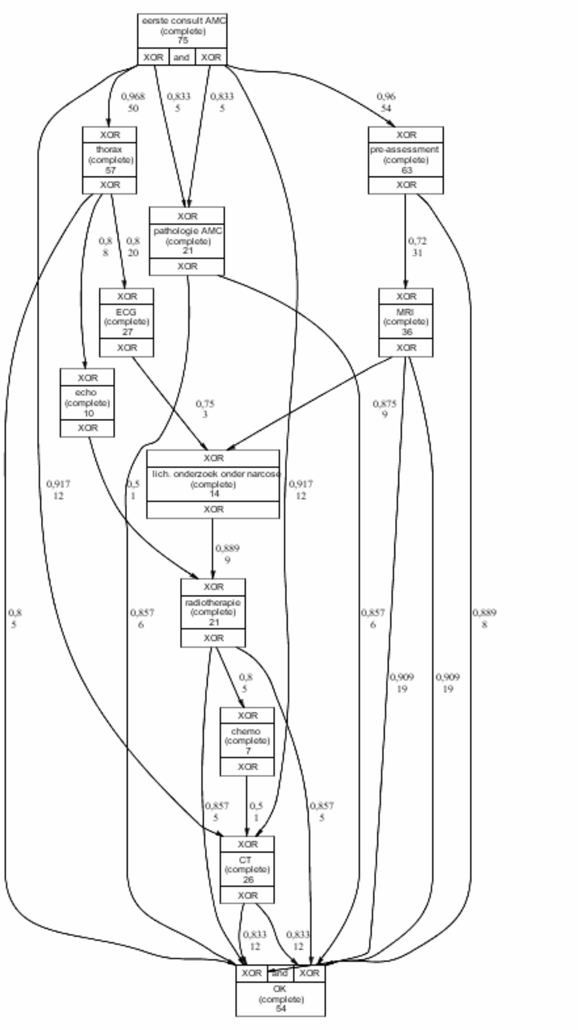


Figure 5: The HN of the first medical consult for womb cancer at AMC.

The HN has its shortcomings, in case of a linking between two activities, which has the characteristics of a vague "OR". First, the HN can only show the splits and joins of the type exclusive XOR and AND. Second, the HN has difficulties with the representation of forks, which is connected to several activities and has the combination of AND/XOR. For instance, the split *eerste consult AMC* is connected to the activitities *pre-assessment* (abbreviated *PRE),* *thorax* (abbreviated *T)*, *CT* and *pathologie AMC* (abbreviated *PAT*). As already mentioned, this split is a "vague" OR and enables one or more of these activities. The logical expression of this split would be $PAT \wedge (CT \vee (PRE \vee T))$. The HN represents this in Figure 5 namely $(PAT \vee PRE) \wedge (CT \vee T)$. This is only one solution, but there are more combinations possible like $(CT \vee PRE) \wedge (PAT \vee T)$, etc. The HN can only give one representation of the model.

Next subsection is a process modeling technique, which provide all three type of splits and joins in a process model (i.e. XOR, OR, and AND).

EPC

The process modeling language EPC[7] offers the possibility to represent process models of less structured processes. EPC has the connectors of the three type splits and joins (i.e. XOR, OR and/or AND) [95]. Appendix A.11 shows the EPC model of the first medical consult for womb cancer. EPC uses combinations of the three types between the activities to illustrate the characteristics between these activities. But this representation has a disadvantage. It is very difficult to understand as user.

Conclusion

Despite these findings about all three mentioned modeling techniques (i.e. DG. HN and EPC), the DG can be used for representation of the less structured processes. DG illustrates the process model in a general and is easy way to understand. The DG can be fit for illustrating of less structured processes without big efforts in the representation.

Part of the research study was the analysis in the predecessor of *ProM*, called *Little Thumb*. This tool has been developed at the TU/e. It has more heuristics measures than *ProM*. Also these additional measures lead to unsatisfied results and insights for the less structured processes (see appendix A.8). Based on these above mentioned findings, we formulated the research objective.

1.6 Original Research Objective

In the early stage of the research project, we formulated the research objective. The original research objective is as follows:

The interest for less structured processes is originated from the research work on data of hospitals. In the hospital, the activities are more flexible, but have also certain structure. The structure is based on the combination of medical diagnosis and treatments. There is also a certain relation between these treatments and the results. For example, in the hospital, the following activities might be taken place:

After registration of the patients, follows the identification of the medical complaint. In the case of a heart complaint might be followed by the activities of the cardiologist like ECG and blood test, the activities of the radiologist like thorax and echo, diagnosis and the decision about surgery. After the identification sometimes starts only the activity of blood test. Sometimes follows the activity of an echo by the activity of ECG, etc.

The processes of the hospital have the characteristics of less structured. The present techniques, which are implemented in the process mining tool, called ProM, are less useful for analyzing this type of data. Based on this experience and interests was the original goal formulated.

The original definition of the goal is developing a PM technique that will be more useful for the mining of data of less structured processes.

1.7 Conclusions

This chapter was about the characteristics of the less structured processes, the research area of the project, i.e. PM, and the original research objective. Less structured processes also called flexible processes are characterized by “vague” type OR. Our interest is the PM of this kind of

[7] EPC is an acronym for **E**vent **P**rocess **C**hain, which is a popular technique for business process modeling [52].

processes. In real life some processes are less structured, like the processes in a hospital or processes in a municipality.

The process modeling languages in the process mining tool *ProM* have certain problems with the representation of less structured processes. The HM is a PM technique, which is useful for analyzing of several types of event logs including noise. The HM represents the process model according the DG and the HN. The two mentioned representations of the HM give insufficient insight of the behaviour of the splits and joins of less structured processes. The number of parsing and the dependency measure in both representations gives only a good indication if the split/join of the AND or exclusive OR (XOR). The HN is an extension of the DG, which shows also the type of split/join. It illustrates only two types, e.g. the AND and the XOR. HN gives only representation of a process model, if this contain a certain combination of XOR and/or AND between activities.

The process modeling language EPC has the possibility to show the three type of connectors, i.e. XOR, OR and AND. However, the definition of the OR by EPC is unclear or too complex.

DG can be used as starting point for the representation of less structured processes. DG represents the process model in a general and easy way. This type of process model fits better for less structured processes than HN and EPC. There are fewer efforts necessary for the development of the representation for this type of processes.

In the early stage of the research project, we formulated the research objective based on the characteristics of the less structured processes. The interest for less structured processes was originated from the research work on data of the hospital “Catharina”. The current PM techniques in *ProM* can not deal with the less structured processes. The original research objective was the developing of a PM technique, which will be more useful for the mining of less structured processes.

2. Analysis of Process Modeling Languages and Research Project

This chapter starts with the analysis of the current process modeling languages, which can be used to present the result of analyzing less structured processes. The analysis of the current process modeling languages like PN, EPC and YAWL[8] are the starting point for developing a new tool. Subsequently, starts the general description of the research project. The research objective, the research questions, research domain and research strategy are part of the research project and will also described in this chapter.

2.1 Result of Literature Review

The study of the characteristics of the less structured process started with the literature review. The literature review was about the different types of splits and joins of several process modeling languages (PML). The result of the literature review is as follows:

Many Workflow Management Systems(WfMS) have been developed the last decades. The (commercial) offerings have their own modeling techniques and properties. But none of these PML is useful for implementation for our purpose. Our purpose is the characterization of different types of splits and joins especially of less structured processes.

The less structured processes are characterized by the vague OR. Some of these PMLs like YAWL and EPC have this type of fork at home. The semantics of the OR in a PML can be represented on two ways, i.e. local and non-local. ***The result of the study illustrates that PMLs with local semantics are very complex and PMLs with non-local semantics have unclear definitions [92,93,94].***

2.2 Adjusted Research Objective

The results of this research study showed that the original research objective had to be adjusted. The objective is not to develop a new PM technique which results in a formal process model with corresponding semantics, but rather a knowledge discovery tool. This new tool gives a characterization of the splits and joins in a process model. The starting point for our research project is the basic structure of the PML of the DG, which is based on the HM algorithm. The HM is a general PM technique and determines the relation between activities on a heuristic way. The DG represents the process model in a general and simple way. This graph has therefore the possibility to show the characteristics of the splits and joins (i.e. XOR, OR, and AND). In other words, the research objective is:

The research objective is the development of a knowledge discovery tool, based on the mining of the DG. This new tool characterizes the splits and joins of less structured processes and the activities, which are directly connected to the split/join.

[8] YAWL is an abbreviation for **Y**et **A**nother **W**orkflow **L**anguage. The workflow tool is based on a rigorous analysis of existing workflow management systems (WfMS) and related standards using a comprehensive set of workflow patterns. The open-source tool is founded by Queensland University of Technology researchers. The WF-tool is based on the PN and is freely available [48]. YAWL is a completely new language with independent semantics, which are mapped onto high-level PNs.

The new tool is characterized by the two words, i.e. *discovery* and *knowledge*. It *discovers* the *behaviour* of the *forks* in the model. This information can be used as *knowledge* for further (statistical) analysis of the processes in the model.

The extension of the *ProM* framework with a knowledge discovery tool leads to more practical utility of this framework. The new tool has the opportunity to deal with less structured processes and gives insight in the characteristics of these types of processes. There are only a few PM techniques in *ProM* (i.e. Fuzzy Miner, Association Rule Miner and Performance Sequence Diagram Analysis) which can also analyze less structured processes [68,69,70,71]. But their points of view are totally different (see Appendix A.18-A.20).

The research objective is the starting point for the formulation of the research questions. Next section describes the research questions of this project.

2.3 Research Questions

For the realization of this tool, we developed some research questions. These questions are derived from the adjusted research objective and support the findings of the new knowledge discovery tool. They are based on the research objective and the current options in the process mining tool. The five research questions are the following:

Q1: Which kind of information is discoverable about the splits and joins of less structured processes by the new tool?
The knowledge discovery tool has statistical measures, which must give information about the characteristics of the type of split/join of less structured processes. This question indicates the main features of the new tool.

Q2: Is it possible to extend the first part of the HM (the mining of DG) with an additional tool which can deal with the less structured processes?
The HM has already some additional options like the representation of the DG in other forms like HN, EPC, and PN. It is important to know, if the new tool can be added to the current HM as an option.

Q3: Which kind of information can be discovered by the new tool between the activities of less structured processes?
The knowledge discovery tool has statistical measures, which must give insight between the dependencies of the activities. These activities are directly connected to the same split/join. This question indicates some more statistical analysis techniques, which uses the statistical values of the new tool as starting point for getting more information about the processes.

Q4: To what extend does the knowledge discovery tool validate internal and external data of less structured process?
The new tool will be tested by internal and external data. The internal data are artificial logfiles[9]. These type of files must give insight to what extend this new tool is useful by using various event logs of less structured processes. External data are the files from healthcare.

[9] Artificial data are characterized by the generic term of the activities. The names of activities are shortened and unique by the first letter of the alphabet. The advantage of this kind of data is their generality function, but their meaning is less. Artificial data can be generated like CPN Tools (see appendix). CPN Tools (Coloured Petri Nets) is a PML for design, specification, simulation, validation and implementation of large software systems [64,65].

Q5: To what extend is the new tool suited in practice?
The new tool will be tested by external data from the hospital. It is interesting to know to what extend this new tool is useful in practice for the less structured processes. This question gives also answer for future work on the PM of less structured processes.
The above mentioned research questions will be answered in the following chapters. The research questions and the research objective are part of the research domain of this master project. Next section describes the research domain.

2.4 Research Domain

The framework of this master project is about the forks in the process model. In order words, the research domain is the analysis of the splits and joins. The forks may also have loops. Also part of this research project are the direct connected activities to the splits/joins. These activities characterize the splits and joins in the process model.

The research domain exists of two approaches, i.e. a practical and a theoretical. The healthcare is the applied approach of this research domain. The theoretical approach is formed by the artificial data. Both are also noted in this work. Let's illustrate the research domain according to a practical example.

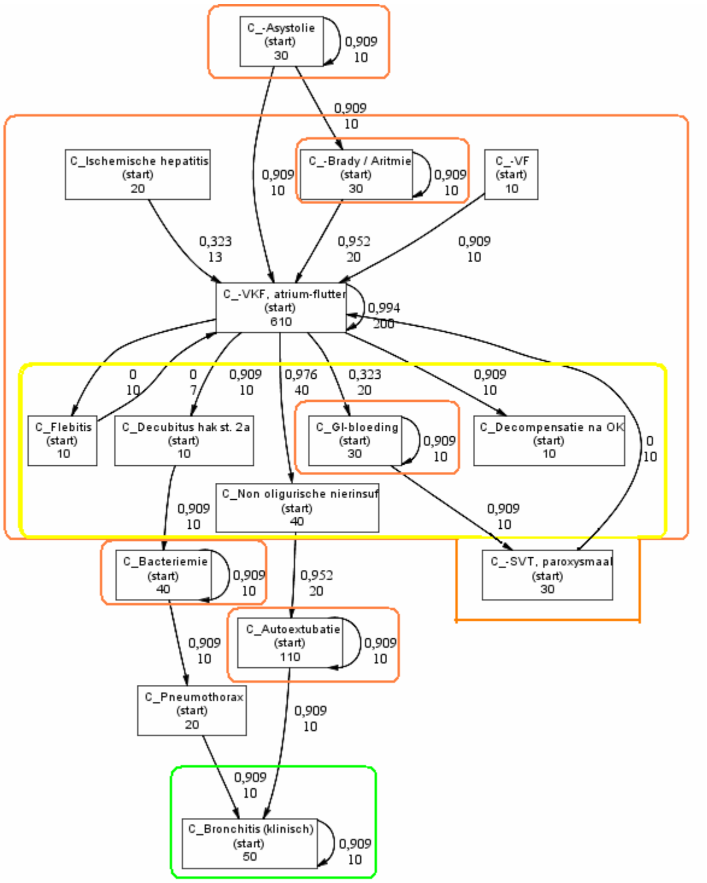


Figure 6: Process model of a treatment on patients of heart-and vascular diseases.

Figure 6 shows a process model for a treatment log of the hospital "Catharina". It concerns treatments on patients with heart- and vascular diseases [73,74].

The marking between the activities in this process model are examples, which gives the research domain of this master project. For instance, the marking about the activity *C_VKF-atrium-flutter* is a split and a join. It has also loop of length one. As split, it is directly connected to 6 activities (including itself) like *C_Flebitis*, *C_Decubitus hak st. 2a*, etc. As join, it is directly connected to 7 activities (including itself) like *C_VF*, *C_Flebitis*, etc. These activities are part of the study to this fork (as split and join).

Not part of this research study is for instance the activity *C_Pneumothorax (start)*. This activity can only be activated by the activity *C_Bacteriemie* and enables only the activity *C_Bronchitis (klinisch)*. In order words, this activity is directly related to only one activity and is not directly connected to a split/join. The research study excludes also the analysis of loops of length one between two direct related activities to a fork. For example, the activity *C_GI_bloeding* and the activity *C_decompensatie na OK*. In this situation, we analyze both activities in relation to the fork (*C_VFK, atrium-flutter*) without the cycle of *C_GI-bloeding*. This cycle has no influence on the fork (*C_VFK, atrium-flutter*).

The research strategy falls within the research domain. It gives insight in the phases of the research project, which will be described in the following section.

2.5 Research Strategy

The research project distinguishes four phases, i.e. analysis, design, implementation, evaluation and validation (Figure 7). It starts with the analysis of the current PM technique and the (statistical) measures. Subsequently starts the design phase, which exists of two parts, i.e. the functional design and technical design. The functional design contains the selection of the statistical measures for the characterization of the type splits and joins, the visualization of the statistical results by tables, and the extension of the DG (Appendix A.4). The technical design explains which way the design of the data mining results and the statistical information in the process model has to be implemented in the framework *ProM* (pages 154-155 of [64]).

After the design starts the implementation of the new tool in *ProM*. The new tool will be implemented in Java. Java is an original release redefined programming language for the Internet (page 14 of [54]).

The last step of the research project is evaluation and validation of the new tool. The validation of the knowledge discovery tool will be performed by internal as external data. The internal data are artificial event logs. These logs have certain characteristics of less structured processes and can be manipulated. The external data are the data from the “Catharina” hospital and several hospitals in Italy [50].

The conclusions and recommendations are the final part of this phase. The findings during the validation of the new tool give insight in the weaknesses and strengths of the new tool and the opportunities in the research domain PM of the less structured processes. Figure 7 shows in overview the research planning.

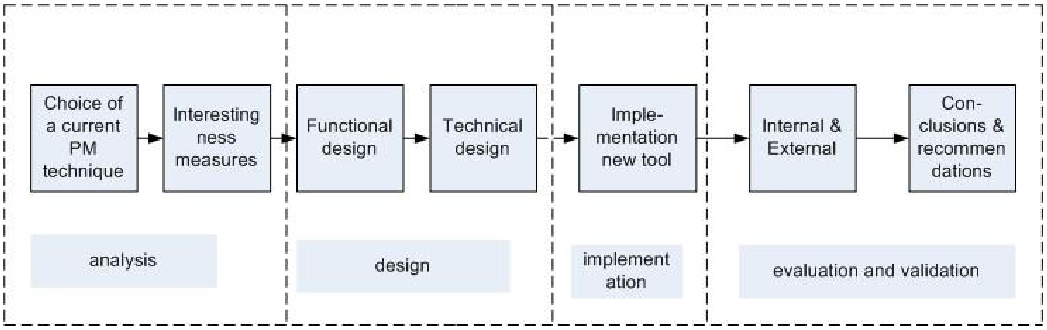

Choice of a current PM technique
Interesting ness measures
Functional design
Technical design
Imple- mentation new tool
Internal & External
Con- clusions & recommen dations
analysis
design
implement ation
evaluation and validation

Figure 7: The research planning in overview.

2.6 Conclusions

This chapter was about the analysis of the research project. The analysis started with the literature review. The literature review was about the different types of splits and joins of less structured processes and the semantics of several PMLs like EPC and YAWL, which include the characteristics of the OR. The result of the literature review is that none of the current PMLs is useful for implementing of the goal "*Characterization of the splits and joins of less structured processes*". This result has adjusted the original research objective.

The research objective is now the development of knowledge discovery tool, which characterizes the splits and joins of less structured processes and the direct related activities. The new tool uses the mining of the DG as starting point. It will be added in the PM tool, called *ProM*. The discovering of the characteristics of the process model will be applied to the forks and the activities which are directly connected to the split/join. The knowledge can be used for further analysis of the process model.

The research questions are based on the research objective. The five research questions have been formulated. The most questions will be answered after testing the new tool by using artificial data and data from the hospitals in Italy, the "AMC" at Amsterdam and "Catharina" at Eindhoven. The answers of these questions will be placed in the conclusions of the following chapters.

The research strategy has been formulated after the research objective. It contains seven steps. The research project distinghuises four phases, i.e. the analysis, design, implementation, validation and evaluation. Execution of the phases is done partly parallel and partly in series.

3. The Analysis of the Forks

This chapter describes the analysis of the splits/joins in a process model. The forks in the model give insight in the behaviour of the type of processes. Using an example we will show the characteristics of the forks from several points of view. Also part of this chapter is the answer of the first research question, namely:*”Which kind of information is discoverable about the splits and joins of less structured processes by the new tool?”*

3.1. Introduction

W is an event log and *T* are the tasks in this log. *DG(W)* is a *Dependency Graph (DG)* for the tasks in log *W*. We can use any algorithm for mining *DG(W)*. The first part of the HM appears a good candidate for mining *DG(W)*, because it determines the relation between the tasks based on the given log in a simple heuristic and general way [14]. For each fork *X* in *T*, the *DG(W)* gives you *In(X)* en *Out(X)* (i.e. the input and output connections for *X*). For instance, HM illustrates these sets in the description panel (see Appendix A.5). An input or output set with two or more elements is a *fork*. Our aim is the characterization of *less structured forks.* A more precise definition of what we mean by that will follow. As an example, we take an event log from the following Petri Net (PN) in Figure 8.

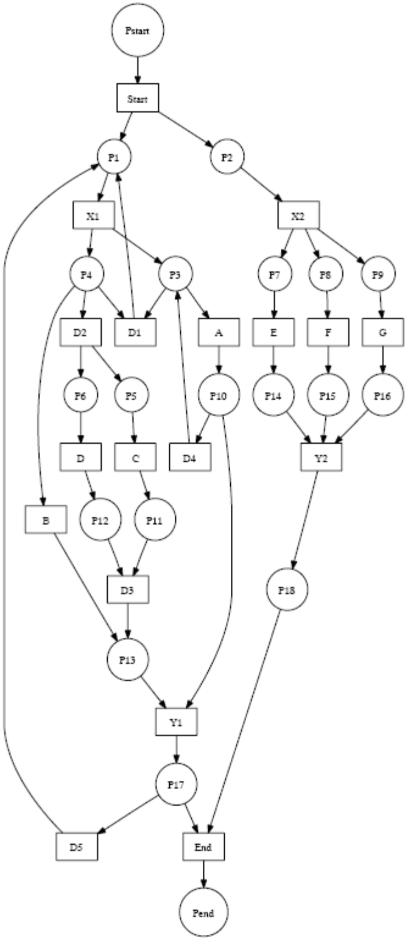


Figure 8: The PN of an example.

The first three traces of a random event log with 1,000 traces are displayed below:

```
Start, X2, G, X1, A, B, F, Y1, X1, E, Y2, X1, X1, X1,C, D, A, A, A, A, A, A, Y1, X1, D, A, C, A, A, Y1, End
Start, X1, X2, C, E, G, A, D, F, Y2, A, Y1, X1, X1, A, B, Y1, X1, B, A, Y1, X1, B, A, A, Y1, End
Start, X1, X1, B, X2, E, G, A, Y1, F, Y2, X1, B, A, A, A, Y1, X1, D, A, C, Y1, End
```

Figure 9: The First three traces of the event log.

Remark that the dummy tasks *D1,…,D5* in the process model (Figure 8) are not registered in the log. They are only used as additional tasks to model the split after task *X1* and the different loops in the model. Our mining goal is to characterize all forks in *DG(W)*, but in this illustration we focus only on output fork of *X1* (i.e. the *X1* split). In other words, we are only interested in the relation between the *X1* split and to its direct connected tasks. The result of applying the HM on the log with 1,000 cases results in a *DG(W)* with *Out(X1) = {X1, A, B, C, D}*. The set *Out(X1)* belongs to the fork *X1* with the output connections to the tasks *X1*, *A*, *B*, *C*, and *D*.

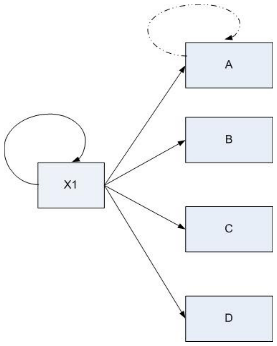


Figure 10: The relation between the X1-split and the tasks X1, A, B, C, D.

Remark that both *X1* and *A* can be repeated (i.e. loops of length one). Figure 10 illustrates the dependency relation between the *X1*-split and its direct connected tasks *X1*, *A, B, C,* and *D*.

Although that the *X1*-split is not really an example of an unstructured split, mining of the *X1*-split without information about the dummy tasks is for most mining algorithms already impossible. The PN has the characteristics of structured processes. But by using the dummies in the PN for simulation, we can apply the generated log for the analysis of less structured forks.

After the PN and the set description, we can describe the set up of the matrix. The set up of this matrix uses the set of the fork as starting point. Next section is about the set up of the matrix.

3.2. The Set up of the Matrix

The first step is the development of the corresponding binary two dimensional pattern matrix for the split *X1*. The two dimensional pattern matrix is a data layout, which contain only zeroes and ones in the cells. The columns of this matrix are the elements of the set *Out(X)* for the output side of the split or *In(X)* for the input side of the join. The elements are the related tasks to the fork (i.e. split or join). Each row in the matrix represents a "basket", which is from the "market basket analysis" [59]. This process analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets". Each basket can be represented by a Boolean expression assigned to the items (i.e. zero or one). In our situation, the items in the basket are the elements (i.e. tasks) of the set *Out(X)* or *In(X)*. The cell of a task is one, if it is present in the "basket". Otherwise, it gets zero in the cell. Let's apply this technique on the above mentioned example.

The basic idea of this technique is simple. Because the set *Out(X1)* has five elements, we start with a matrix with five columns, i.e. *X1*, *A*, *B*, *C*, and *D*. Subsequently, we walk from left to right, and trace after trace through the event log. The corresponding column gets a 1 in the cell, if we meet a task in the trace that is part of the set *Out(X1)*. If we parse an instance of tasks *X1*, we start a new row in the matrix. In case, we meet only task *X1* before the next task *X1,* the column *X1* gets a 1 in the cell, and a new row begins. The filling of the current row stops, when we approach the end of a trace (i.e. after task *End*).

Let's start with the first trace in Figure 9. We start from the left side. In this row, we indicate which elements of *Out(X1)* really follows *X1*. None of the first three tasks in the first trace (e.g. *Start*, *X2*, and *G*) are *X1*. Nothing happens so far. The fourth task in the trace is *X1*, leading to a new row. This is the also starting point for the filling of this matrix. Subsequently the tasks *A, B*, *F* and *Y1* follow before the next *X1* (split). Only the tasks *A* and *B* are part of the set *Out(X1)*. These tasks get a 1 in the cell of the corresponded column *A* and *B*. After task *Y1* follows *X1*, which leading to a new row in the matrix. The result of this first row is as follows:

X1	***A***	***B***	***C***	***D***
0	*1*	*1*	*0*	*0*

The following tasks are *E* and *Y2*, before the next *X1*. The tasks *E* and *Y2* are not part of the set *Out(X1)*. In this situation, the corresponding column *X1* gets a 1 in its cell and followed by a new row. After *Y2* appear three times sequentially *X1*. The first two *X1*s leads to the same identical row as before mentioned, because only *X1* appears after *X1*. This is the situation, where task *X1* repeats itself three times. The results of these three rows are:

X1	***A***	***B***	***C***	***D***
1	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*

The third *X1* leads to a new row, because we meet subsequently the tasks *C*, *D*, *A* and *Y1* before the next *X1* (split). This is the situation, where the split enables one or more directly connected tasks. Only, the last mentioned task *Y1* is not part of the set *Out(X1)*. Task *A* appears (sequentially) 6 times, but will be only once noticed in the matrix like the other tasks *C* and *D*. Task *A* is namely only once enabled by the *X1*-split and five times by itself. In this situation, this technique looks only between *X1*-split and its direct connected tasks. Because of this, the loop of *A1* is lightly dotted in Figure 10.

After this, follows task *X1*, which leading to a new row. After *X1*, follow the tasks *D*, *A* (3 times), *C*, *Y1*, and *End*. The new row gets only 1 in the cells of the corresponded columns *A*, *C* and *D*. This is the last row of the first trace in the matrix. After the approach of the last task *End* in the first trace, a new row in the matrix starts. Results of the two last rows for the first trace are:

X1	***A***	***B***	***C***	***D***
0	*1*	*0*	*1*	*1*
0	*1*	*0*	*1*	*1*

This procedure will also be applied on the second and third trace of the event log as given above. This leads to the following matrix of the three cases:

X1	*A*	*B*	*C*	*D*
0	*1*	*1*	*0*	*0*
1	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*
0	*1*	*0*	*1*	*1*
0	*1*	*0*	*1*	*1*
0	*1*	*0*	*1*	*1*
1	*0*	*0*	*0*	*0*
0	*1*	*1*	*0*	*0*
0	*1*	*1*	*0*	*0*
0	*1*	*1*	*0*	*0*
1	*0*	*0*	*0*	*0*
0	*1*	*1*	*0*	*0*
0	*1*	*1*	*0*	*0*
0	*1*	*0*	*1*	*1*

Figure 11: The binary matrix of the first three traces.

We have to finish the development of this matrix for all 1,000 traces, but in the end it contains the binary information about the possible *Out(X1)* patterns. In the case of a join, we walk the opposite direction through the traces (i.e. we start at the end of the trace towards its beginning). In this situation, we use the set *In(X)* instead of *Out(X)*. The rest of this above described technique stays the same.

The question now is, how to use this binary matrix to inform the user about the characteristics of the *X1*-split. Below we will sum up some possible measurements and techniques. After that we have to think which measurements to use and how we can present the results in a comprehensive way.

3.3. The Methods for Analyzing of the Forks

This section illustrates several methods that give insight in the behaviour of the forks in the model. Each method has its point of view and uses some (selected) metrics for the characterization of the forks.

3.3.1. The Frequencies and Probabilities of the individual Tasks

First, we can use the column information (the number of one's) to calculate the probability that task *X1* is followed by *X1*, *A*, *B*, *C*, or *D* respectively. This calculation method uses the vertical direction of the matrix. The probability is the number of one's of a task divide by the total number of rows in the binary matrix. For instance, task *X1* has five one's in the binary matrix (Figure 11). The total number of rows in this matrix is 15. The probability of the individual task *X1* is 5/15. The results of this method are:

Task	*Frequency*	*Probability*
X1	*5*	*5/15*
A	*10*	*10/15*
B	*6*	*6/15*
C	*4*	*4/15*
D	*4*	*4/15*

The probabilities of the individual tasks illustrate the occurring of the tasks in the binary matrix. In this example, the *X1*-split enabled five tasks with different probabilities.

Remark that the summation of the probabilities of the individual tasks varies between zero and the total number of tasks. This summation distinguishes the type of the fork (i.e. XOR, OR, AND), i.e.:

Summation of individual tasks	*Type of fork*
1	*XOR*
Unequal to 1 and the number of tasks	*OR*
The number of individual tasks	*AND*

In our example, the summation of the probabilities is 29/15. It illustrates the split of type OR. The type gives only the first impression about a fork. But to get more insight in the behaviour of the fork, we have to look to the following method, i.e. the pattern frequencies in the binary matrix.

3.3.2. Pattern Frequencies

We can calculate the frequency for the various binary patterns and order them from high to low. This calculation method uses the horizontal direction of the matrix. For instance, the first row in the binary matrix (Figure 11) has the pattern frequency *AB* (i.e. the tasks *A* and *B* are one). This pattern frequency (i.e. *AB*) has six one's and the total number of rows in this matrix are 15. The probability of this combination *AB* is 6/15. The results of this method are:

X1	*A*	*B*	*C*	*D*	*Comb. Act.*	*Frequency*	*Relative Frequency*
0	*1*	*1*	*0*	*0*	*AB*	*6*	*6/15*
1	*0*	*0*	*0*	*0*	*X1*	*5*	*5/15*
0	*1*	*0*	*1*	*1*	*ACD*	*4*	*4/15*

Two remarks about the pattern frequencies. First, it supports the type of fork, which is already distinghuised by the probabilities of the individual tasks. Let's say the total number of individual tasks is *n*. The pattern frequencies illustrate this by:

Pattern frequencies	*Total number of combinations*	*Type of fork*
All singulair elements as combinations	*n*	*XOR*
All combinations is possible	*Between 1 and* 2^n	*OR*
All elements in one combination	*1*	*AND*

All pattern frequencies of one task characterize the XOR. For instance, a split *O* has the set *Out(O)* with the elements *{P,Q,R}*. For an XOR, only task can be enabled. The split has 3 combinations, i.e. *P*, *Q*, and *R*. In the situation of an AND, all tasks are enabled. The split has only one pattern frequency, namely *PQR*. The OR, has possible and maximally 8 (= 2^3) of tasks (e.g. *PQ*, *QR*, etc.). According the three cases in the PN example, it has 3 combinations (i.e. *AB*, *X1*, and *ACD*) which support the findings of an OR-split.

Second, the relative frequency of the pattern frequencies always sums up to 1. The pattern frequency seems useful information if the number of different patterns is not too high. It illustrates the combinations of tasks in the binary matrix. In this example only 3 of possible 32 patterns appear in the log. But an important characteristic of less structured processes is exactly that many different binary patterns appear in the event log. We will use the following artificial two dimensional binary matrix to illustrate this in Figure 12. The first five columns of this matrix are the same as the previous matrix (see Figure 11).

X1	***A***	***B***	***C***	***D***	***K***	***L***	***M***	***N***
0	*1*	*1*	*0*	*0*	*1*	*1*	*1*	*1*
1	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*0*
1	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*1*
1	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*0*
0	*1*	*0*	*1*	*1*	*1*	*0*	*1*	*1*
0	*1*	*0*	*1*	*1*	*1*	*0*	*1*	*0*
0	*1*	*0*	*1*	*1*	*1*	*0*	*0*	*1*
1	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*
0	*1*	*1*	*0*	*0*	*0*	*1*	*1*	*1*
0	*1*	*1*	*0*	*0*	*0*	*1*	*1*	*0*
0	*1*	*1*	*0*	*0*	*0*	*1*	*0*	*1*
1	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*
0	*1*	*1*	*0*	*0*	*0*	*0*	*1*	*1*
0	*1*	*1*	*0*	*0*	*0*	*0*	*1*	*0*
0	*1*	*0*	*1*	*1*	*0*	*0*	*0*	*1*

Figure 12: The second binary matrix.

By adding the tasks *K*, *L*, *M*, and *N* all patterns becomes different and the pattern frequency does not inform us about the characteristics of the *X1* to *D* part of the matrix. The relative frequency in this example is for all patterns the same, i.e. 1/15. So, we need some more techniques to get insight in the behaviour of the patterns.

3.3.3. Binary Relations between Tasks

A number of measurements to calculate the dependency between tasks are available. Examples are the interestingness measures for association patterns in [34] like support, confidence, correlation coefficient, and IS, etc. These metrics are based on probabilities between two tasks and use a binary matrix as input. We selected two of these metrics, e.g. correlation coefficient and IS, for analyzing the dependency between two tasks (see also Appendix A.9).

The Correlation Coefficient

The correlation (ϕ) measures the degree of dependency between two tasks. It is a well-known metric and intuitively easy to understand. It is defined as the covariance between the two tasks divided by their standard deviations. The correlation varies between -1 and 1. The interesting correlation values are -1, 0, and 1. The following table illustrates the properties between these two tasks of these correlation values.

ϕ	***Degree of dependency between two tasks***	***Type of fork***
-1	*Strongly negative*	*XOR*
0	*Independency*	*-*
1	*Strongly positive*	*AND*

Two tasks have a strong negative dependency if the correlation is -1. They exclude each other. In this situation, the fork illustrates the characteristics of the type XOR. In the first binary matrix (see Figure 11), the correlation between *X1* and A is -1 (i.e. $\phi\,(X1,A) = -1$). Two tasks are strongly positive dependent if all rows contain both tasks in the matrix. In this situation, the fork has the characteristics of the type AND. In the first binary matrix (see Figure 11), the correlation between *C* and *D* is 1 (i.e. $\phi\,(C,D) = 1$). Two tasks are independence, if the correlation is zero. The product of the probability of the two tasks is equal to the probability of both tasks. In this situation, further analysis of the type of fork is necessary (see Appendix A.9).

These are only the boundaries, which have nice properties. But, there are more situations, where we have these characteristics. This difficult to discover with only one metric.Therefore, we need more metrics. Let's first describe the following (selected) metric, e.g. IS.

IS

The second measure IS or Cosine is based on the probability of the intersection of both tasks related to the square root of the probability of the single task. This metric looks mainly to the occurrence of two tasks and their intersection in the binary matrix. IS varies between zero and one. These are the interesting points. The IS value of zero indicates no occurring of both tasks in the binary matrix. The IS value of one shows a strong dependency between the two tasks.

Correlation versus IS

In some circumstances both measures support each other. In other situations, the metrics are complement of each other. Both measures support each other when both activities have a strong dependency. They support also each other, for instance if the correlation value is negative and the IS value 0 between two tasks. In this situation a certain rows have one for only one of the two tasks and the remaining rows have zeroes values for both tasks. In other words, both tasks do not appear in a row and the other rows contain only one task. The correlation value will decrease (more negative), if the numbers of rows increase where only one task is performed.

The correlation coefficient gives less insight in the behaviour between two tasks, if the total number of rows of the binary matrix is small. The correlation coefficient is a bad indicator, especially where co-presence of both tasks is more important than co-absences of both tasks. In this situation, the measure IS indicates better the characteristics of the dependency of two tasks. Let's the just above mentioned points about the correlation and IS translate (apply) in a more concrete situation of PM.

An Example of PM with Correlation and IS

The best way to illustrate the properties of the two above described metrics (i.e. correlation and IS) is according a process model. For instance, a fork (*P*) which is part of a (bigger) process model will not always be enabled (Figure 13). The split includes a loop of length one and is connected to the tasks *P*, *Q* and *R*. The set *Out(P)* of this fork *P* has the elements *{P,Q,R}*.

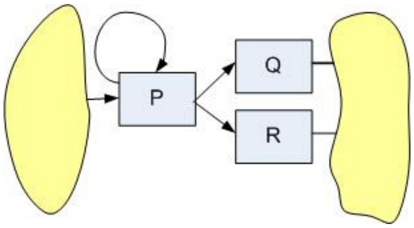


Figure 13: Part of a process model.

The split (P) can only start up the tasks Q and R, if the task P is finished. In order words, the tasks Q and R can not performed, if task P is not finished with its repeating or task P was not enabled. In the first situation, these rows in the matrix of this set (i.e. $Out(P)$) are *1 0 0* under the columns P Q R. Second situation is flavorless and not interesting, because this will not noted in the matrix.

Let's say that only one of these tasks (i.e. Q or R) after the split (P) can be enabled. In this situation, the rows in the matrix can be only *0 1 0* or *0 0 1* under the columns P Q R. These rows show the oppositely values for Q and R. Let's illustrate a matrix of this split (P). For instance, the matrix of this setup might be:

P	***Q***	***R***
1	*0*	*0*
1	*0*	*0*
0	*1*	*0*
0	*0*	*1*
1	*0*	*0*

This matrix illustrates clearly the characteristics of XOR, because only of these tasks can be enabled (i.e. P, Q or R). The matrix supports this by the only singulair *1* in every row. This example shows that three times task P is performed. The tasks Q and R are only once enabled.

However, the correlation is less than -1, which is caused by the zero rows under the columns Q and R in the matrix (i.e. *1 0 0)*. But, it is still negative (e.g. $correlation(Q,R) = -0.25$). The correlation indicates not directly these characteristics of the split. Therefore, a second metric is necessary, namely IS. The metric IS gives the value 0 in this situation(e.g. $IS(Q,R) = 0$), which means that there is no intersection between these tasks. In order words, there is no situation where both tasks were simultaneously performed after P, which is clearly the case. The matrix has no rows with *0 1 1*. By looking to the values of both measures (i.e. correlation and IS), we discover the characteristics of an XOR of this split. This example shows clearly that <u>only</u> the combination of <u>both</u> metrics gives more insight in the behaviour of this split.

It is also the case for an AND. Every row in the matrix is <u>all</u> zero <u>or</u> one under the connected tasks after or before the fork (e.g. *1 0 0* or *0 1 1*). In this situation, the correlation is less than 1, but the IS has the value of 1 between the tasks Q and R.

One remark about the example. The characteristics of the XOR-split can also be determined by the summation of the individual tasks to this split (P). In case of an XOR-split, the probabilities of these individual tasks sum up to 1 (i.e. $p(P)=0.6$, $p(Q)=p(R)=0.2$). In this simple example, the summation can be easily calculated. But, in a situation of many tasks connected to a fork, the calculation of the summated probabilties takes some more time including a chance of miscalculation. The look to the values of both metrics is faster and safer. Let's look to the values of correlation and IS in the PN example.

<u>The Results of Correlation and IS in the PN Example</u>

In the first binary matrix (see Figure 11) for example, the correlation between the tasks B and C is the same as the correlation between the tasks B and D (i.e. $\phi\,(B,C) = \phi\,(B,D)=-0.492$).

Another interesting point is the correlation between the tasks $X1$ and C and between the tasks A and B, which are exactly the opposite of each other (i.e. $\phi\,(X1,C) =-0.577$ and $\phi\,(A,B) =0.577$).

IS value between the tasks *C* and *D* is also 1, which confirms the strong degree of dependency between both tasks (i.e. $\phi(C,D) = 1$). IS value between *X1* and *A* is zero, which supports the negative dependency between both tasks (i.e. $\phi(X1,A) = -1$). The metric values of all related tasks to *X1*-split are:

Related activities	ϕ	***IS***
X1A	*-1*	*0*
X1B	*-0.289*	*0*
X1C	*-0.577*	*0*
X1D	*-0.426*	*0*
AB	*0.577*	*0.775*
AC	*0.426*	*0.632*
AD	*0.426*	*0.632*
BC	*-0.492*	*0*
BD	*-0.492*	*0*
CD	*1*	*1*

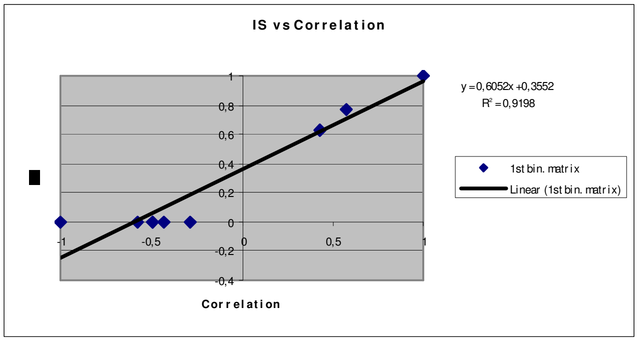


Figure 14: Table and graph of IS versus Correlation.

Figure 14 shows the table and the correlation between the two selected metrics, based on the first binary matrix (see Figure 11). IS and correlation have a strong positive relation (i.e. *R-square = 0.9198*). The article of [34] noted that the interestingness measures for association patterns are positive correlated. Another nice property of the binary relations is that the correlation and IS values will not changed by adding more columns to the matrix. These values are based between two tasks.

<u>Conclusion</u>

The measures correlation and IS gives insight in the degree of dependency between tasks. In some situations, they support each other and in other situations both metrics complement each other. Only one metric gives in some situations insufficiently insight in the dependency between tasks. For this reason, it is necessary to apply *more* measures for analysis of the behaviour (i.e. dependeny) between the tasks.

3.3.4. Grouping of the Tasks

A disadvantage of the binary relations between tasks is that they only characterize relations between two tasks. In our example, if *B* appears after *X1* then there is also a high probability that *D* appears (see table in Figure 14). However, relations between groups of tasks are also possible. This subsection describes two mining techniques, which can be used for the determination of the relation between groups of tasks, namely *Apriori* and *Tertius*. Let's start with the description of the *Apriori* algorithm in more detail.

<u>The Apriori Algorithm</u>

The *Apriori* algorithm seems a useful method for the mining of association rules. The association rules are formulated as a result of the market basis analysis ([39], pages 227-283 of [59]), which aims at finding regularities in the shopping behaviour of customers of supermarkets, on-line shops, etc. This algorithm is the best known algorithm for association rule induction and is developed by Agrawal et al. (1993). In this situation, the mining technique tries to find sets of tasks that frequently took place together, so that from the presence of certain tasks can infer (with a high probability) that certain other presence tasks will also be present. Such information, expressed in the form of association rules can increase very fast by the number of tasks in the set. Therefore, we use a *minimum support* and *minimum confidence* from the set of all possible rules. The minimum support is the percentage that the rule can be applied to. Its confidence is the number of cases in which the rule is correct relative to the number of cases in which it is applicable. The thresholds of the minimum support and minimum confidence are fixed, e.g. 0.10 respectively 0.90.

As an example, we use the *Weka*[10] association rule miner with default settings in combination with the second binary matrix (see Figure 12) with more different patterns. The result is displayed in Figure 15.

The first part of the rule shows the task, its presence (zero = not and one = presence) and its number of occurrence in the matrix. After the arrow, the rule shows the presency (zero = not and one = presence) and the number of occurrence of the related task in the matrix. The confidence is between zero and one and will be calculated by the number of occurring of both tasks divided by the occurring of the given task. For instance, the first rule shows the non presences of the tasks *C* and *D* (11 times) in the matrix. The confidence for both tasks is 11/11=1.The *Apriori* algorithm found for the second binary matrix:

1. *C=0 11➔D=0 11 conf:(1)*
2. *D=0 11➔C=0 11 conf:(1)*
3. *X1=0 10➔A=1 10 conf:(1)*
4. *A=1 10➔X1=0 10 conf:(1)*
5. *L=1 8 ➔C=0 D=0 8 conf:(1)*
6. *C=0 L=1 8 ➔D=0 8 conf:(1)*
7. *D=0 L=1 8 ➔C=0 8 conf:(1)*
8. *L=1 8 ➔D=0 8 conf:(1)*
9. *L=1 8 ➔C=0 8 conf:(1)*
10. *X1=0 N=1 7 ➔A=1 7 conf:(1)*

Figure 15: Results by Apriori algorithm.

Rule 3 and 4 indicates that *X1* and *A* are in a XOR relation, which is also indicated by the statistical measure between two tasks. Because we have applied the *Apriori* algorithm on only 15 cases, there are also a number of rules, which are more logically formulated and less related of presence of tasks. For instance, the rules 6 and 7, which tell the non presence of task *D* is related to the non-presence of task *C and* presence of task *L*. We are more interested in the relation between tasks in a group, which is based on the presence of tasks.

Tertius Algorithm

Tertius is another algorithm for the induction of association rules. It is a less known mining technique and based on first-order logic rules [90,91] like *A* ➔*B*. In contrast to the *Apriori* algorithm, *Tertius* can also discover of or-rules between groups (e.g. the rules 1, 2, and 10 in Figure 16 on page 37). It combines the support and confidence in one measure, called predictive accuracy. For instance rule 1 in Figure 16 has a predictive accuracy of 0.960392. This algorithm is like the *Apriori* algorithm part of the *Weka* tool. Tertius rules found for the second binary matrix:

[10] Weka is a data mining software tool, which is developed by the University of Waikato, Hamilton, New Zealand. **W**aikato **E**nvironment for **K**nowledge **A**nalysis, version 3.5.6 is freely available [88].

```
1.   /* 0.960392 0.000000 */ B = 0 ➔ X1 = 1 or C = 1
2.   /* 0.960392 0.000000 */ B = 0 ➔ X1 = 1 or D = 1
3.   /* 0.960392 0.000000 */ A = 1 and C = 0 ➔ B = 1
4.   /* 0.960392 0.000000 */ A = 1 and D = 0 ➔ B = 1
5.   /* 0.891806 0.000000 */ A = 0 ➔ X1 = 1
6.   /* 0.891806 0.000000 */ X1 = 1 ➔ A = 0
7.   /* 0.792811 0.000000 */ D = 1 ➔ C = 1
8.   /* 0.792811 0.000000 */ C = 1 ➔ D = 1
9.   /* 0.729018 0.000000 */ B = 0 and L = 1 ➔ X1 = 1
10.  /* 0.729018 0.000000 */ A = 1 ➔ B = 1 or L = 0
```

Figure 16: Results by Tertius algorithm.

Conclusion

The *Apriori* algorithm (and *Tertius* rules) contains useful information but they are also redundant. It has also some less interesting rules like the relation of non-presence between tasks (e.g. rule 1, 2 in Figure 15).

3.4. Conclusions

The forks in the process model characterize the type of processes. We selected the HM algorithm for the mining of the DG, which will be used as starting point for the characterization of the forks. The input and output sets of the DG are the starting point for the setup of the binary two dimensional matrix. After the setup, we use four methods for analyzing this type of data layout. These measures have their own properties and illustrate the characteristics of the forks from several points of view.

The four menthods are extensively described in this chapter. The first method gives insight in the behaviour of the individual tasks. It distuighuises also the type of split/join (i.e. XOR, OR, AND). The second method is the frequency patterns. The frequency patterns show the possible combinations of tasks, which are presence in the matrix. The third method is based on the degree of dependency between two tasks. Two metrics, i.e. correlation coefficient and IS will be applied for the analysis of dependency. The last method is based on grouping of tasks. The well known *Apriori* algorithm gives insight in the dependency between these tasks in a group. This is also the answer of the first research question:*" Which kind of information can be discovered about the splits and joins of less structured processes by the new tool?"*

As a result, these methods and their corresponding measures (except the tertius rule) will be implemented as tables in the new knowledge discovery tool. Next chapter describes the functionality of the new tool. The four methods, which are part of the new tool, will be subsequently applied to artificial logs and real data from several hospitals.

4. Description of the new Knowledge Discovery Tool

This chapter is about the description of the new knowledge discovery tool, called "*SplitJoinIndicator*" (*SJI*). It contains three sections. The first section describes the *SJI* in general. The second section is about the representation of the characteristics of the forks. The characteristics of the forks are described in the previous chapter. Also part of this chapter is the answer of the second research question. This chapter ends with the conclusion.

4.1 Introduction

SJI uses the mining of the DG as starting point. This is the first step of HM algorithm. As a consequence of these findings, the new tool can be added to the current HM. In order words, the new tool will be part of the HM algorithm.

The new tool can be seen as an extension of the current options in the HM. These options are placed in a submenu and can be selected after the start up of the HM, which shows the DG as results. This submenu will be shown in front of the tool by using the right click button of the mouse. The current DG options, i.e. DG with and without semantics will be extended with the DG including the information of the forks. The new submenu, called "*Display dependency graph*", belongs to the new tool and has four options. The first three options are the graphical representation of the HM. The fourth option is the statistical tables of the characterization of the forks. Figure 17 illustrates the extension of the current options in HM.

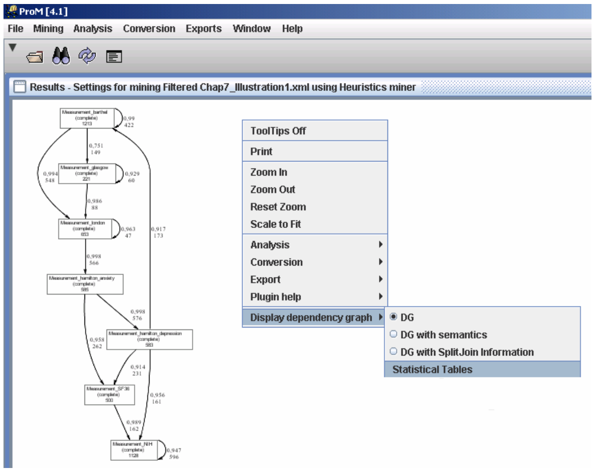

ProM [4.1]
File Mining Analysis Conversion Exports Window Help
Results - Settings for mining Filtered Chap7_Illustration1.xml using Heuristics miner
ToolTips Off
Print
Zoom In
Zoom Out
Reset Zoom
Scale to Fit
Analysis
Conversion
Export
Plugin help
Display dependency graph
● DG
○ DG with semantics
○ DG with SplitJoin Information
Statistical Tables

Figure 17: Screenshot of the new knowledge discovery tool.

The tables of the four methods, called *"Statistical Measures for Activities", "Combination table", "Statistical measures for related Activities"*, and *"Apriori table"* starts up after the selection of the option *"Statistical tables"*. The four tables and the DG are forms of the representation. Next section describes the representation of the forks.

4.2 Representation of the Forks

As already mentioned, the representation of the forks can be illustrated in two ways, i.e. the statistical tables and the visualization of the DG. Both types are based on the four methods, giving insight in the characteristics of the forks of less structured processes (see also Appendix A.4). The following subsections describe the tables and the new fitted DG in more detail.

4.2.1 The Statistical Tables

The first representation is the statistical tables. There are four tables, which are based on:

1. Individual tasks;
2. The pattern frequency of tasks;
3. The binary relation between tasks;
4. The grouping of tasks.

The above numbering is also the best way for analyzing the characteristics of the forks. In this order, the methods are also described in the previous chapter. Each table illustrates the direct connected activities of the forks and the corresponding metrics values. The first two tables characterize the forks in the process model. The first table gives already insight in the type of split resp. join (i.e. XOR, OR, AND). The second table gives insight in the possible combinations of tasks to a fork. The third table gives insight in the dependency between tasks, which are directly connected to a fork. The last table gives more insight in the relation between the tasks, which are directly related to a fork. Let's describe the tables in more detail.

Table I: Individual Tasks

The first table is called "*Statistical Measures for Activities*"and gives insight in the frequency and probability of the individual tasks, which are directly connected to the fork. Table 3 shows a part of the table of the individual tasks to a fork and its frequencies and probabilities according the example in section 3.1. The probability in this table is the frequency of the individual task divide by the frequency of its fork. For instance, the probability of the individual task *A* is equal to 2,244/3,024 = 0.742 rounded in three decimals.

Split/Join	*Type*	*Frequency*	*Activities*	*Frequency*	*Probability*
…	…	…	…	…	…
X1	*Split*	*3024*	*X1*	*780*	*0.258*
			A	*2244*	*0.742*
			B	*1048*	*0.347*
			C	*1120*	*0.37*
			D	*1120*	*0.37*
…	…	…	…	…	…

Table 3: Statistical table for the individual tasks.

Table II: The Pattern Frequency of Tasks

The second table, called *"Combination table"*, gives insight in the combination of tasks related to the split/join. The number of possible combinations characterizes the heterogeneity of a fork. It shows the complexity of the fork. The fork is more heterogenous and complex, when it contains many pattern frequencies. It illustrates all combinations and the probability always sum up 1. It is possible that rows in a matrix only contain zeroes. This means that a split/join has none tasks enabled after its finishing. In this case, the table gets a row, called *"None activities"* and its corresponding metric values. Table 4 shows a part of the combination table. The calculation methode of the probability is the same as in the previous table.

Split/Join	*Type*	*Frequency*	*Combination of activities*	*Frequency*	*Probability*
…	…	…	…	…	…
X1	*Split*	*3024*	*A C D*	*1120*	*0.37*
			A B	*1048*	*0.347*
			X1	*780*	*0.258*
			A	*76*	*0.025*
…	…	…	…	…	…

Table 4: The combination table.

Table III: The Binary Relation between Tasks

The third table, called *"Statistical Measures for Related Activities"*, shows the degree of dependency between two tasks. The metrics support, correlation coefficient and IS are part of this table. The support is the percentage of both tasks in the binary matrix (i.e. probability). It is based on the rows where both activities are present in the binary matrix. For instance, the support of *AB* is the frequency of AB (i.e. 1,048) divide by the frequency of its fork (i.e. 3,024). Rounded in three decimals 0.347. Table 5 illustrates the dependency degree between the activities to split *X1*.

Split/Join	*Type*	*Frequency*	*Related activities*	*Frequency*	*Support*	*Correlation*	*IS*
…	…	…	…		…	…	…
X1	*Split*	*3024*	*X1A*	*0*	*0*	*-1*	*0*
			X1B	*0*	*0*	*-0.429*	*0*
			X1C	*0*	*0*	*-0.452*	*0*
			X1D	*0*	*0*	*-0.452*	*0*
			AB	*1048*	*0.347*	*0.429*	*0.683*
			AC	*1120*	*0.37*	*0.452*	*0.706*
			AD	*1120*	*0.37*	*0.452*	*0.706*
			BC	*0*	*0*	*-0.559*	*0*
			BD	*0*	*0*	*-0.559*	*0*
			CD	*1120*	*0.37*	*1*	*1*
…	…	…	…		…	…	…

Table 5: The statistical table between two direct related tasks.

Table IV: The Grouping of Tasks

The fourth table, called *"Apriori table"*, illustrates the relation between tasks in a group, which is a result of the *Apriori* algorithm. Also part of this table the frequency between the related activities and their confidence. This table shows only relation between tasks that satisfy on the minimum support and minimum confidence (i.e. 0.10 respectively 0.90). The number of rules is default 10. Table 6 shows the first five rules between the activities to split *X1*, which are generated by the *Apriori* algorithm. For instance, the first rule to split *X1* is the presency of task *A* leads to a non-presency of task *X1*. The confidence interval is 2,244/2,244 = 1.

Split/Join	*Type*	*Frequency*	*Activities*	*Frequency*	*Activities*	*Frequency*	*Confidence*
…	…	…	…	…	…	…	…
X1	*Split*	*3024*	*A=1*	*2244*	*X1=0*	*2244*	*1.000*
			X1=0	*2244*	*A=1*	*2244*	*1.000*
			D=0	*1904*	*C=0*	*1904*	*1.000*
			C=0	*1904*	*D=0*	*1904*	*1.000*
			A=1 and B=0	*1196*	*X1=0*	*1196*	*1.000*
…	…	…	…	…	…	…	…

Table 6: Table of relations between activities in a group

4.2.2 The Graphical Representation

The DG will be provided with the probabilities of the several branches. It will be placed between the forks and the directly connected tasks in the process model. Figure 18 illustrates the first part of the DG with the probabilities. This DG is mined on the event log, which is described in section 3.1.

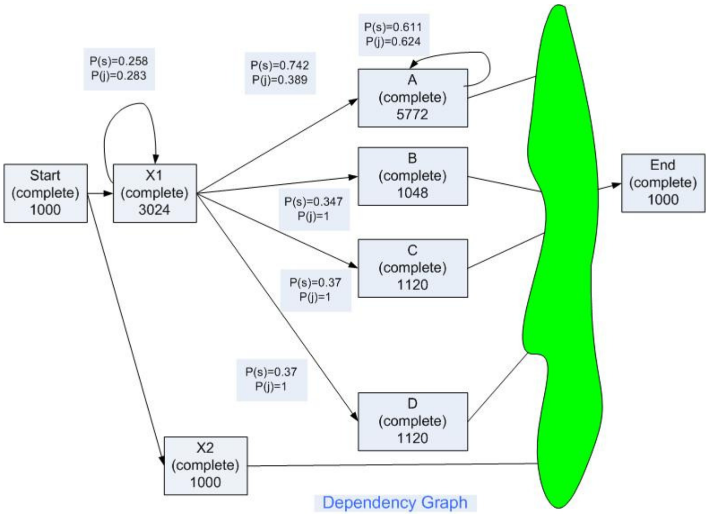


Figure 18: The DG with the probabilities of the branches.

The probabilities are derived from the statistical table of the individual tasks and based on both sides of the connection. These are rounded in three decimals. The illustrating of the probabilities of both side (i.e. *p(s)* and *p(j)*) in the DG gives already the first insight in the behaviour of the forks (structured or less structured). There are two type of probabilities noted in the process model, i.e. *p(s)* and *p(j)*.

The first probability, called "*p(s)*", is based on the split side. It is the number of occurring of the individual tasks divide by the number of occurring of the split. For instance, *X1* (split) which is 3,024 times performed, enables 1,120 times task *D*. The probability (= *p(s)*) is 1,120/3,024 = 0.37, which is the same as the probability of the individual task in table 1.

The second probability, called "*p(j)*", is based on the join side. It is the number of occurring of the individual activities divide by the number of occurring of the join. For instance, task *B* starts up (1,048 times) after the finishing of task *X1* (3,024 times). The probability (= *p(j)*) is 1,048/3,024 = 0.347. Another example, task *D* is performed 1,120 times and was 1,120 times enabled by *X1* (split). The probability (= *p(j)*) is 1,120/1,120 = 1. In other words, this probability (i.e. *p(j)*) looks from the opposite direction.

Some remarks about the probabilities in the DG. The probabilities vary between 0 and 1. The probability will be reduced to 1, if the number of individual is higher than the number of occurring of the join. This information will also placed in the loops of length one. For instance, task *A* as join is performed 5,772 times and is connected by itself and *X1*. The probability of the loop and *X1* are resp. 3,528/5,772 = 0.611 and 2,244/5,772 =0.389. Task *A* is also a split, which enables itself (3,604 times) and activity *Y1* (2,168 times). The probabilities are resp. 3,604/5,772 = 0.624 and 2,168/5,772 = 0.376. It shows that both probabilities (i.e. *p(s)* and *p(j)*) in a cycle are different.

4.3 Conclusions

The new knowledge discovery tool, called *SJI* is an extension of the current HM. *SJI* has particularly been developed for analyzing of the forks of less structured processes. It contains two options, i.e. the DG including split/join information and the statistical tables. These options will be placed as an extension in the submenu of the HM. This submenu, which belongs to the new tool, can be started after the mining of the HM algorithm. This is also the (Yes) answer of the second research question: *"Is it possible to extend the first part of the HM (the mining of DG) with an additional tool which can deal with the less structured processes?"*

The visualization of the characteristics of (less) structured forks in the process model is part of the new tool. There are two representations, i.e. the DG and the table form. The DG with split/join information illustrates the most interesting characteristics of the forks from various points of view. *SJI* has four statistical tables (i.e. table I to IV). Each table represents one method, which are described in the previous chapter. Every method has it own measures and properties. These (selected) metrics values are related to the direct connected activities to a fork.

The implementation of the new tool is based on the functional design. After the implementation the validation of the new tool starts, which will be performed with artificial and data of several hospitals. This will be described in the following chapter.

5. Validation of the New Knowledge Discovery Tool

This chapter is about the validation of the new knowledge discovery tool, called "*SplitJoinIndicator*" (*SJI*). After the introduction, first the description of the internal validation starts. Subsequently, the tests of the new tool by external data begin. The knowledge which is discovered by this new tool can be used for further analysis. Section three illustrates some more interesting opportunities by using *SJI*. This section demonstrates also some properties of the new tool. This chapter ends with a general summarization of the validation. Also part of this chapter are the answers of the third research question, namely *"Which kind of information can be discovered by the new tool between the activities of less structured processes?"* and the fourth research question, namely *"To what extend does the knowledge discovery tool validate internal and external data of less structured process?"*

5.1 Introduction

The validation of the new knowledge discovery tool starts after the implementation. It will be performed by internal and external data. The new tool will be used in the first part for discovering the characteristics of the processes, which are already known. The internal data are the artificial event logs, which are generated by CPN tools. The external data are the logs from various hospitals. Next section is about the validation of *SJI* by artificial data.

5.2 Internal Validation of the new Tool

This section is about the validation by artificial data. Two event logs will be applied for this test. The first example is based on the example in section 3.1. Some of the part of this process is structured and the other part is less structured. The less structured part of this process model contains only a certain pattern frequency. This process model is a little more complex compared to a structured part. The structured part of this process model are characterized by forks of the type AND. This information and more is already observed by the first three cases of the event log, which are mentioned in section 3.3. We use this information as check for the reliability of *SJI*.

The second log has only the characteristics of structured processes. These type of processes are characterized by the splits and joins in the process model, which only contain AND and/or XOR. Everything is known about this process model; because all is mentioned in the paper of [14] (pages 13-16). Based on this information of both process models in advance, we can easily and extensively validate the new tool given event logs from many points of view.

5.2.1 First Example

The first example uses an event log of 1,000 traces, which is noise free. The process model is based on the default settings of the HM. The four tables of this process model are noted in Appendix A.12. Figure 19 illustrates the DG (including the probabilities i.e. *p(s)* and *p(j)*) of this example. From the first point of view, the DG corresponds with the process model according the example. Let's look first in more detail to this DG.

Validation of the DG versus PN

According the DG starts the process with the activity *Start*, which has a dependency relation with the activities *X1* and *X2*. The activity *X1* has a dependency relation with the activities *A*, *B*, *C*, and *D*. The last four activities are connected to *Y1*. The second part of this model, *X2* has a dependency relation with the activities *E*, *F*, and *G*. The last three mentioned activities are connected to *Y2*. The activities *Y1*and *Y2* enables the last activity *End* in this process according the DG. The DG (Figure 19) corresponds clearly with the process model (PN) according section 3.1 (Figure 8).

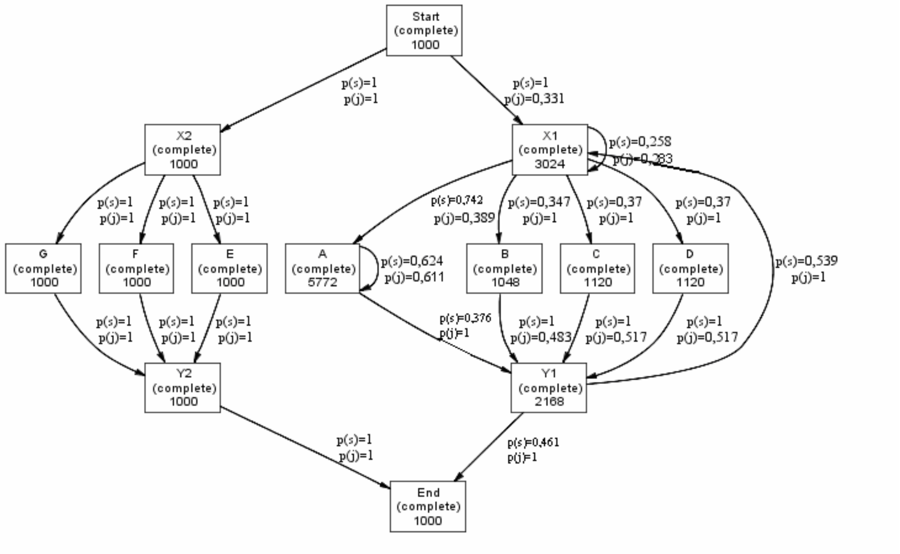
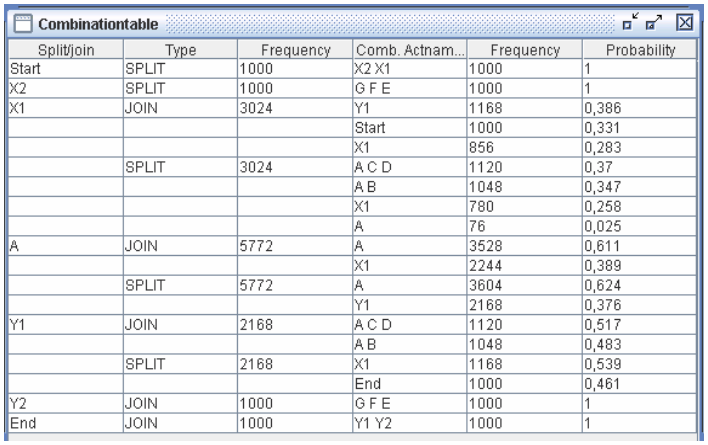


Figure 19: The DG of the second example including the probabilities.

Let's validate now the characteristics of this process model by using the DG (including the probabilities, i.e. *p(s)* and *p(j)*) and the four statistical tables (methods) from different point of views.

Validation of SJI versus PN

The first table, called *"Statistical Measures for Activities"* (see Appendix A.12) and the process model in Figure 19 show clearly that the process model has two types of processes at home (i.e. structured and less structured). Both part starts directly after the split (*Start*). The structured part of this process model is characterized by the activities *X2*, *E*, *F*, *G*, and *Y2*. The first and second table of the new tool supports these findings. The first table and the probabilities (i.e. *p(s)* and *p(j)*) illustrates that all probabilities of the activities *E*, *F*, and *G* in the process model are 1. The second table, called *"Combination table"*, shows only combination between the split (*X2*) and join *(Y2*), namely *EFG*. The third table, called *"Statistical Measures for related Activities"* shows that all three metrics (i.e. support, correlation and IS) are 1. The fourth table, *"Apriori table"* illustrates a strong relation between these activities (rules 7 to 10). This all, confirms that the split (*X2*) and join (*Y2*) are of the type of AND, which is the same as in the PN of the example in section 3.1. It shows that the four tables and the probabilities in the DG for this part of the model are correct.



Combinationtable

Split/join	Type	Frequency	Comb. Actnam...	Frequency	Probability
Start	SPLIT	1000	X2 X1	1000	1
X2	SPLIT	1000	G F E	1000	1
X1	JOIN	3024	Y1	1168	0,386
			Start	1000	0,331
			X1	856	0,283
	SPLIT	3024	A C D	1120	0,37
			A B	1048	0,347
			X1	780	0,258
			A	76	0,025
A	JOIN	5772	A	3528	0,611
			X1	2244	0,389
	SPLIT	5772	A	3604	0,624
			Y1	2168	0,376
Y1	JOIN	2168	A C D	1120	0,517
			A B	1048	0,483
	SPLIT	2168	X1	1168	0,539
			End	1000	0,461
Y2	JOIN	1000	G F E	1000	1
End	JOIN	1000	Y1 Y2	1000	1

Table 7: The combination table of the first example.

The second part of this process model is clearly less structured. The activities *X1*, *A*, *B*, *C*, *D*, and *Y1* are part of it. The second table, called *"Combination table"* (see Table 7) illustrates that only 4 combinations of possible 32 exist. The split (*X1*) enables only the combinations *ACD*, *AB*, *X1*, and/or *A*. Three of the four combinations (i.e. *ACD*, *AB*, and *X1*) were already present in the first three traces (cases) of the event log, which are mentioned in section 3.3.2. The missing of the fourth combination (*A*) in the first three cases is properly the low presency of this activity. According the *"Combination table"* appears *A* only 76 as a singular activity in the event log. This is a probability of 0.025 (= 76/3,024). The probabilities of the first combinations (i.e. *ACD*, *AB*, and *X1*) in the event log differs not much with the probabilities of these combinations in the first three cases (i.e. less than 0.1), which confirms the reliability of the new tool.

Another check are the measures correlation and IS by the new tool. Table III (*"Statistical Measures for related Activities"*) illustrates that the correlation activities *X1* and *A* are equals -1. The correlation and IS between the activities *C* and *D* are both 1. The IS between the activities *X1A*, *X1B*, *X1C*, *X1D*, *BC*, and *BD* are zero in statistical table III. All these values are the same as noted in section 3.3.3 (Figure 14). We can do one more validation on the metric support (probability). The frequency and support between the activities A and B in table III are the same as the frequency and probability of *AB* in the *"Combination table"* (resp. 1,048, and 0.347).

The fork (*X1*) has both types, namely split and join. As split (*X1*), it enables the activities *X1*, *A*, *B*, *C*, and/or *D*. Activity *A* has as individual activity the highest probability. It appears 2,244 in the matrix of the fork (*X1*), which occurs 3,024 times. The probability of the individual activity *A* equals to 2,244/3,024 = 0.742, which are mentioned in the first table (*"Statistical Measures for Activities"*) and in the DG. The probabilities of the individual activities *X1*, *A*, *B*, *C*, and *D* compare to their probabilities in section 3.3.1 are small (i.e. less than 0.13), which confirms the reliability of *SJI*. The summation of these probabilities is 2.087. The activities *A*, *B*, *C*, and *D* are connected to the join (*Y1*). The probabilities sum up to 2.893. These support that the split (*X1*) and join (*Y1*) have the characteristics of a "vague" OR, which already is mentioned in section 3.1.

One remark is about the split and join of activity *A*. This has clearly the characteristics of a XOR, which is known by the PN of this example (Figure 8). Both probabilities sum up to 1 (i.e. as split 0.624 and 0.376, and as join 0.389 and 0.611). Only one of the two activities is enabled by the split (i.e. *A* or *Y1*). As join, it starts up by itself or by activity *A*. It illustrates that the behaviour of the fork as split and join differs a little bit, because the probabilities are different (i.e. *p(s)* and *p(j)*).

Conclusion

The new tool has clearly discovered that this process model is less structured. This process model contains all three type of splits and joins (i.e. XOR, OR, and AND), which were identified by the new tool. *SJI* distinghuises also the two parts of the process model, with their own characteristics of processes (i.e. structured and less structured). The new tool discovered the type of forks in the structured part (i.e. AND). The less structured part of this process model is less complex, because it contains only a few combinations of patterns frequency. The first two tables of the new tool show already clearly the characteristics of the forks in the model. The third and fourth tables support these findings and give some additional insight between activities. *SJI* confirms the given information about this process model in the sections 3.1 to 3.3 and indicates already as a reliable tool for less structured processes.

The second example will used for getting more reliability of the new tool. Another point is the validation of *SJI* for completely structured processes. However, this is properly a less strong point of this tool. But, it is important to know that the new tool also correct works for this type of processes. After the validation phase, the tool will be part of the *ProM* tool and applied on all type of processes (including non apriori knowledge).

5.2.2 Second Example

As an example of the artificial data is generated by the process model in [14] (pages 13-16). The event log is one noise free event log with 1,000 traces. The HM contains 10 parameters, which can be changed if necessary [14] (see also Appendix A.17). By changing the parameters setting of the HM (i.e. *relative to best threshold = 0.2, positive observations = 3, dependency threshold = 0.85*), the model contains also the low frequency dependency relations (i.e. between *D* and *K,* see DG in Appendix A.13). Figure 20 shows the DG including the probabilities of the individual activities in the model (i.e. *p(s)* and *p(j)*).

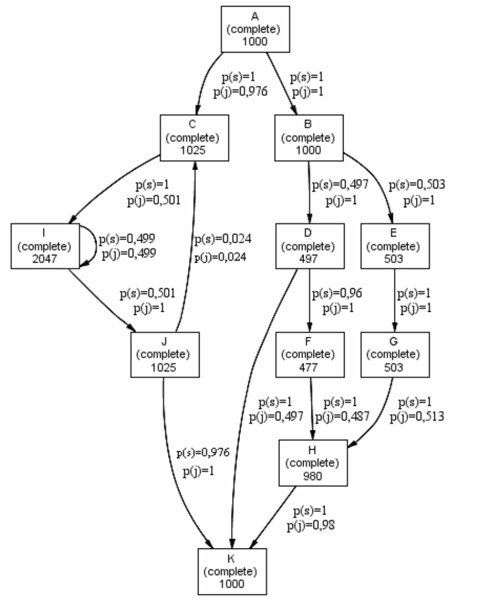


Figure 20: The DG including the probabilities of the branches.

Let's validate the characteristics of the process model by using the DG and the four tables (see appendix A.13).

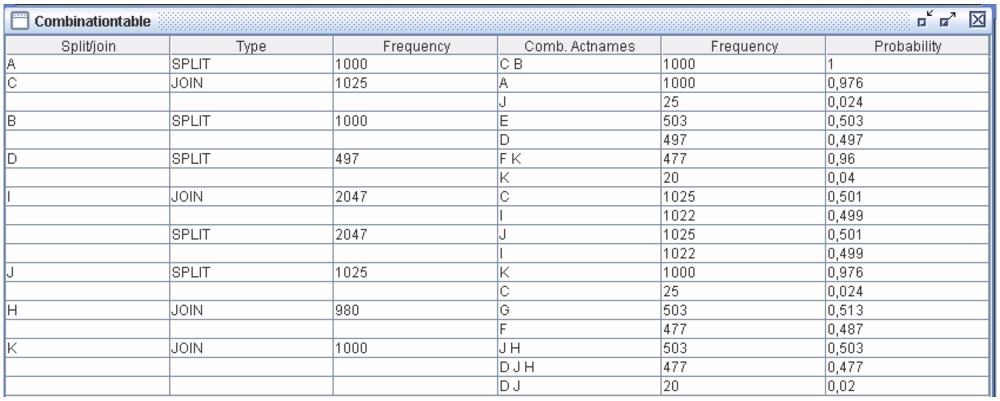
Validation of SJI versus Process Model

The first table *"Statistical Measures for Activities"*, which gives insight of the individual activities, and the DG (including the probabilities *p(s)* and *p(j)*) show already the characteristics of almost all forks in the model. The second table *"Combination table"*. Let's look to some forks in this process model.

For instance, activity *A* enables the activities, i.e. *B* and *C*. The probabilities of both individual activities are 1. This indicates the characteristics of an AND-split of activity *A* like in [14]. Activity *B* is a split, which starts up the activities *D* and *E.* According the DG including the probabilities and the first table, the probabilities of the individual activities *D* and *E* are resp. 0.497 and 0.503. This indicates the characteristics of a XOR-split of activity *B* like in [14].

Another example is activity *I*. Activity *I* has both type of forks. The split starts up the activities *I* (itself) and *J* with a probability of 0.499 and 0.501. The join (activity *I*) is enabled by the activities *I* (itself) and *C*. Their probabilities are the same. In both situations, the summations of the direct connected activities are namely 1. It means that the split and join (activity *I*) have the characteristics of an XOR like shown in [14].

Some more attention is needed for the split (activity *D*) and the join (activity *K*). The split (*D*) connects the activities *F* and *K.* Their probabilities are resp. 0.96 and 1. For findings the characteristics of this split, we need the table of the pattern frequency of this process model (i.e. statistical table III resp. table IV). Table 8 illustrates the pattern frequency of the activities.



Combinationtable

Split/join	Type	Frequency	Comb. Actnames	Frequency	Probability
A	SPLIT	1000	C B	1000	1
C	JOIN	1025	A	1000	0,976
			J	25	0,024
B	SPLIT	1000	E	503	0,503
			D	497	0,497
D	SPLIT	497	F K	477	0,96
			K	20	0,04
I	JOIN	2047	C	1025	0,501
			I	1022	0,499
	SPLIT	2047	J	1025	0,501
			I	1022	0,499
J	SPLIT	1025	K	1000	0,976
			C	25	0,024
H	JOIN	980	G	503	0,513
			F	477	0,487
K	JOIN	1000	J H	503	0,503
			D J H	477	0,477
			D J	20	0,02

Table 8: The combination table of the second example.

This table shows that activity *K* is only 20 times directly enabled by the split (*D*), like in [14]. The probability of this singulary activity (*K*) equals to 0.04. The summation of this split is 1, which characterizes the type of an XOR like in [14].

The join (*K*) is connected by the activities *J*, *D* and *H*. Activity *J* and the combination of the activities *D* (i.e. 20) and *H* (i.e. 980 times) enables 1,000 times activity *K,* which shows the characteristics of an AND-join. The activities *D* and *H* have the characteristics of an XOR. Rule 4 (i.e. *D=0 and H=1 (503 times)* ➔ *J=1 (503 times)*) and rule 7 (i.e. *D=1 (497 times)* ➔ *J=1 (497 times)*) of the *Apriori table* supports these findings. The discovery of the characteristics of the join (*K*) by the new tool (using all four tables) is the same as described in [14]. This illustrates also that the four tables of the new tool correct works. This part of analyses asks a little more attention of the user and it is a less point of the new tool.

Conclusion

The combination of the first two tables, (i.e. *"Statistical Measures for Activities"* and *"Combination table"*) and the process model shows clearly the characteristics of complete structured forks. The other two tables (i.e. *"Statistical Measures for related Activities"* and *"Apriori table"*) support these findings. The process model contains only types of splits and joins of the type XOR and AND, which belongs to the type of structured processes. This confirms complety the information about this process model described in [14]. The new tool can also be applied for completely structured processes. But in this case, the additional value of *SJI* is less. Because in some situations, a little bit more attention is needed from the user.

The new tool has been extensively validated by the two artificial data from different point of views. *SJI* has already shown as a reliable tool for the analysis of both types of processes. Now, we can apply *SJI* on logfiles in practice, which are more complex. It is also more unknown about the processes in the given logs. Next section describes the validation of two external data by the new tool.

5.3. Extern Validation of the new Tool

This section is about the validation of *SJI* by external data. The extern logs are originated from various hospitals. This field is characterized by highly *complex* and extremely *flexible* patient care processes. In this work environment, many disciplines within a hospital are involved during the treatment of a patient [75]. The medical activities, which take during the treatment of a patient, are recorded in the given logfile. Some of these activities are done simultaneously. Let's describe the well-known characteristics of the given event logs in more detail.

Characteristics of the medical Logfiles

The processes in healthcare have some typical characteristics. First, the processes in the healthcare are more unstructured as the first example (mentioned in section 5.2.1.) The two event log from the hospitals should have more pattern frequencies of activities than the first example in section 5.2.1. Second, some treatments are influenced by other factors, which are not recorded in the logfile. For instance, treatments can be started up without waiting the diagnosis of a previous medical activity or further treatments of a patient stops, because the diagnosis is positive.

The new tool must discover the typical "medical" properties in the two event logs. The first event log stored information about first medical consult for womb cancer at "AMC" in Amsterdam. The second log contains information about various measurements treatment on stroke patients, which took place in the region of Lombardia, Italy.

After the check of the above mentioned characteristics, it is also interesting to discover some more information by the new tool about the processes in the hospital, which were properly unknown for the experts in the medical world.

5.3.1 Case study 1: First Medical Consult for Womb Cancer at AMC

The first case study is about the first medical consult for womb cancer at the academic hospital in Amsterdam ([55],[72]). 75 female patients had a first consult during the period March – April 2007. The first consult takes place on the gynecological oncology and concerns research for possible cancer in the womb. This treatment contains the following activities: first consult (*eerste consult AMC*), pre-assessment, thorax, pathology (*pathologie AMC*), MRI, ECG, echo, body research including anesthesia (*lich. onderzoek onder narcose*), radiotherapy (*radiotherapie*), chemo, CT and finally OK. The result of the first consult determines which activities will be executed next in the treatment of the patient.

Process Model of Case Study 1

Figure 21 shows the DG (including the probabilities, i.e. *p(s)* and *p(j)*) of the first consult of patients (*eerste consult AMC)* at "AMC". This process model contains no loops. This means once a treatment has been performed, it never will be repeated. Second point is the activities take place in series. The following treatment starts after the previous related activity. The event log of this medical treatment has 75 patients. The four tables of this process model are noted in Appendix A.14. Let's validate first the characteristics of these medical processes.

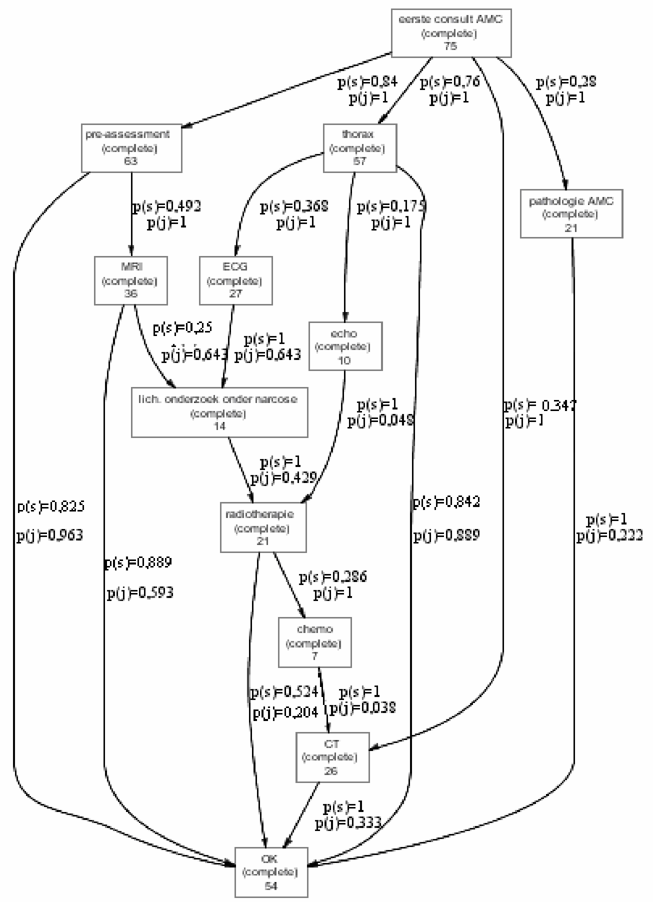


Figure 21: The DG of the first consult including the probabilities.

Validation of SJI versus Case Study 1

The first table *"Statistical Measures for individual activities"* and the probabilities in the process model (i.e. *p(s)* and *p(j)*) shows the frequency and probabilities of the treatments in the process model, which indicate the less structured forks in the model. The first activity *first consult AMC* illustrates already the characteristics of less structured processes. The split (*first consult AMC*) starts up the activities *pre-assessment*, *thorax*, *CT* and/or *pathology*. Their probabilities are resp. 0.84, 0.76, 0.347, and 0.28 (see first table or the probabilities *p(s)* in the

DG noted above the arcs between the split *first consult* and its direct connected activities). The summation of these activities is 2.227, which already indicates a “vague” OR for this fork (*first consult AMC*). This medical process ends with *OK*. This last activity can be enabled by 6 previous treatments (i.e. *thorax*, *pre-assessment*, *MRI*, *CT*, *radiotherapy* and/or *pathology*). The probabilities of these activities sum up to 3.194 (see first table or the probabilities *p(j)* in the DG noted above the arcs between the join *OK* and its direct connected activities). This indicates a “vague” OR for the join *OK*. The new tool discovers clearly type of processes, which confirms the information in advance of the processes of this medical consult for womb cancer.

Let’s look more about the complexity of this process model. The forks in this process model should have more pattern frequencies than the first example in section 5.2.1. Therefore, we need the second table *“Combination table”*. The join *OK* shows this nicely. For instance, the pattern frequencies of the fork *OK* is 16 of possible 64 (= 2^6), which indicates the heterogeneity of the join (*OK*). The heterogeneity of a fork is characterized by the number of pattern frequencies. The fork is more heterogen, if the number of combinations increases. This supports also that the join *OK* is of the type OR. The second table *“Combination table”* shows clearly that the process model of the first consult is more complex. *SJI* confirms the information in advance about this medical treatment.

The discovering of the typical “medical” processes by the new tool needs a little bit more attention. The typical “medical” processes are characterized by starting up treatments by it own and stopping further treatments. The activity *radiotherapy* illustrates nicely these typical “medical” properties. Hence, we need the first two tables ((i.e.*”Statistical Measures for Activities”* and *“Combination table”*). The fork (*radiotherapy*) has both types, i.e. split and join.

Radiotherapy as join, it will be enabled by the activities *echo* once and *body research including anesthesia* (*lichamelijk onderzoek onder narcose*) 9 times. *Radiotherapy* took place 21 times, but only 10 times starts this treatment by its direct related activities. This means that the treatment started up 11 times without waiting of the results of *chemo* and/or *body research*. That is 52.4% the case for this treatment. The second table shows these by the row *“None activities”* (11 times), which belongs to the join *radiotherapy*.

Radiotherapy, as split, enables the activities *OK* and/or *chemo*. These treatments took place 14 times. It seems that *radiotherapy* had 7 times no further treatment started up. The second table illustrates these by the row *“None activities”* (7 times), which belongs to the split *radiotherapy*. This can be possible, if the diagnosis of the radiotherapy for these concerned patients is positive. Starting up by its own and cancelling of further activities is also the case for activity *MRI* (on both sides of the forks). The fork (*radiotherapy*) has the characteristics of less structured and is also less homogenous. *SJI* confirms nicely the typical “medical” properties in the event logs.

<u>New Findings by SJI</u>

Let’s look for some interesting characteristics of this medical treatment, which gives the *SJI* more value as tool.To find some interesting points in a fork, we need table III (*“Statistical Measures for related Activities”*) and table IV (*“Apriori table”*) of this new tool. The last treatment *OK* (join) illustrates some nice properties. Table III shows independency between the treatments *CT* and *thorax* (i.e. correlation is 0). The treatments *CT* and *MRI* are negative related (i.e. correlation is -0.533). The remaining correlation between the treatments vary more or less round zero, which indicate (some) independency. Rule 3 in the *Apriori table* shows that the treatments *thorax* and *radiotherapy* are related to each other. In 98% treatment of *thorax*, also *radiotherapy* is done. There is also a relation between the treatments *thorax* and *pre-assessment*. Rule 10 in the *Apriori table* illustrates 90% treatment of *pre-asssessment*, also *thorax* is done.

Conclusion

The results of the new tool are successfully validated by the information of the medical treatment, which are recorded in the event log. First, *SJI* shows clearly the processes of the first consult are less structured and more complex than the previous two examples. The probabilities of the individual activities in the first table and the pattern frequencies in the second support these findings.

Second, the typical "medical" properties of this logfile are discovered by the new tool. The new tool discovered the typical characteristics of processes in a hospital like starting up treatments without waiting of the diagnosis of a previous treatment and stopping further treatments because of positive diagnosis of previous treatments. This all, *SJI* confirms the given information about this event log. The first two tables give already insight in these typical "medical" processes.

Third, *SJI* discovered some interesting points in this process model, which were unknown in advance. Hence, the third and fourt table of *SJI* were used for these new findings. Appendix A.16 shows some interesting (statistical) analysis of this process model.

The new tool will also be validated with the second practical event log. This logfile is a little bit different comparing the above validated file. The following logfile has more medical records and contains repetition of various treatments.

5.3.2 Case Study 2: The Measurements Treatment on Stroke Patient

The second case study is about data from a preliminary study, which was conducted on patients with acute stroke[11] and transient ischemic attack on first-ever stroke patients in four districts in the region of Lombardia in Italy [50]. It aimed at studying the effect of the American Heart Association guidelines on 386 such patients. The data contain information of patients suffering from the stroke. This information is recorded from the acute phase to the sub-acute phases of the patients from the stroke. Acute phase data pertains to the data of patients that arrive from the stroke symptoms onset. After the first six hours, the patient is considered to be in the sub-acute phase.

The event log contains information about various measurements. It contains 1,213 cases. These are performed on stroke patients. The measurements take place after admission or during the hospitalization of the patient. There are seven types of measurements, also called scales.

Process Model of Case Study 2

The process model starts with the measure *barthel* and ends with the measure *NIH*. It contains loops and has more case ids (traces) than the process model of the first consult. But it is less complex than the previous medical treatment. It is interesting to discover knowledge of this medical measurements treatment. Figure 22 on page 52 illustrates the DG (including the probabilities, i.e. *p(s)* and *p(j)*) of the measurements treatment on stroke patients including the probabilities. The measurements can be repeated (i.e. loops of length one). It depends on the diagnosis of these measures. Only the measures *hamilton anxiety*, *hamilton depression*, and *SF36* will not repeated. The four tables of this process model are noted in appendix A.15.Let first validate the characteristics of these typical medical processes.

[11] Acute stroke is a vascular condition that precipitates neurological damage and is the second leading cause of death in industrialized countries [56].

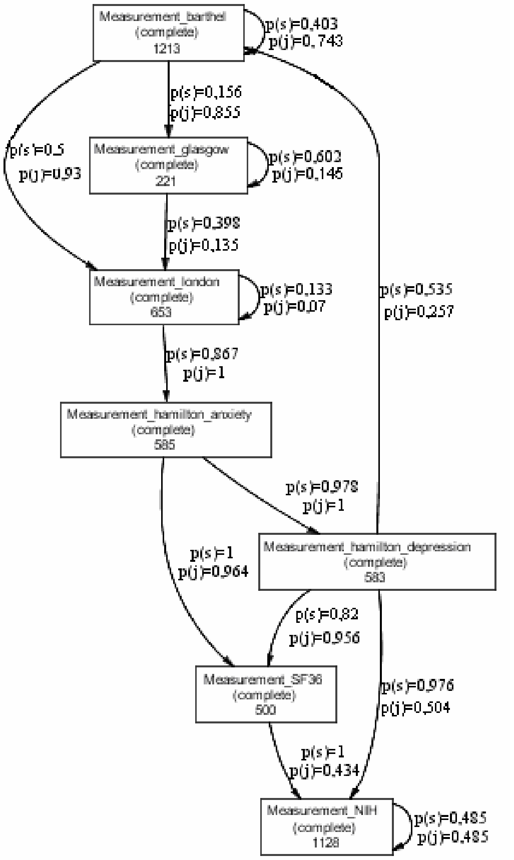


Figure 22: The DG of the measurements treatment including probabilities.

Validation of SJI versus Case Study 2

For the validation of *SJI*, we need the first table or DG including the probabilities. The first table *"Statistical Measures for individual Activities"* and the DG (Figure 22) show the frequency and probabilities of the treatments in the process model, which indicate the less structured forks. For instance, after the measure *hamilton depression* starts the measures *barthel* and/or *SF_36* and/or the last measure *NIH*. The probabilities of these activities sum up to 2.331, which indicates a vague "OR" for the split *hamilton depression* (see table *"Statistical measures of individual Activities"*). The new tool confirms the information about this medical treatment in advance.

Subsequently, we validate the complexity of this process model. Therefore, we need the second table. The second table *"Combination table"* shows clearly that the process model of the measures is less complex comparing to the process model of the first consult (case study 1). The number of combinations of activities varies less. However, the characteristics of the forks in the model are also less structured and less homogenous. For instance, the patterns frequency of the split *hamilton depression* is 7 of possible 8, which indicates the heterogeneity of the split (*hamilton depression*). The remaining splits and joins in the process model have less patterns frequency and more homogenous.

For the discovering of the typical "medical" properties in the event log by the new tool, we need the first two tables (i.e.*"Statistical Measures for Activities"* and *"Combination table"*). *SJI* shows the typical characteristics of processes in a hospital like starting up measures of its own and stopping further measures. For instance, the measures *hamilton aniexity* and *hamilton depression* were resp. 10 and 3 times the last measures of a patient. The *Combination table* illustrates this by the row *"None activities"* of the splits *hamilton aniexity* and *hamilton depression*. The measure *SF36* was 10 times performed without waiting of the results of previous measures (*"None activities"* in the second table). *SJI* confirms the information about the processes of medical measures in advance.

<u>New Findings by SJI</u>

For discovering some interesting more characteristics of this fork, we need table III (*"Statistical Measures for related Activities"*) and table 4 (*"Apriori table"*) of this new tool. Table III shows that the measures *barthel* and *NIH* (i.e. correlation (barthel,NIH) = 0.011), and the measures *NIH* and *SF36* (i.e. correlation (NIH, SF36) = 0.014) are almost independence. The *Apriori table* shows some interesting rules between the measures *barthel*, *SF36* and *NIH*. The measure *NIH* will be performed in 98%, if the measures *barthel* and *SF36* took place (rule 4). The measure *SF36* will also be done in 91%, if the measures *barthel* and *NIH* are performed (rule 9).

Another interesting point in this process model is the characteristics of the fork *barthel*, which includes a loop of length one. The fork contains both types (i.e. split and join).

The measure *barthel*, as join, it can be enabled by the measurement of *hamilton* depression or by itself. Only one activity can take place. As join, the measure *barthel* take once more place until the diagnosis of the *hamilton depression* was received. The individual probabilities of both measures (i.e. 0.743 resp. 0.257) sum up to 1 (see DG or table I). Both have a strong negative dependency (i.e. correlation (barthel, hamilton depression) = -1 in table III). The *Apriori table* supports the negative dependency between these activities by the rules 1 to 4. These findings illustrate clearly the characteristics of an XOR-join for *barthel*.

The measure *barthel,* as a split, the measurement of barthel has other characteristics. After the measurement of *barthel* starts once more *barthel*, *london* and/or *glasgow*. The measure barthel is a lot of times repeated (i.e. probability of 0.403), which indicates the importance of this measure. One of the reasons might be that the diagnosis is not always clear. The results of this measure determine the following measures on stroke patients. Measure *barthel* will only once more performed later, if the diagnosis of the measure of *hamilton depression* is negative or not clear. The performance of both measurements *london* and *glasgow* after the measure *barthel* appears very low (i.e. 72 times and probability of 0.059 in table 2). The *Apriori table* shows that the performance of the measure *barthel* indicates in 489 times non presency of the combinations *london* and *glasgow* visa versa (rule 6 and 9). The activities weak are negative related to each other (i.e. correlation (london,glasgow) = -0.103). This split (*barthel*) has more the characteristics of less structured (i.e. a "vague" OR).

Now we can explain the typical "medical" characteristics in more detail, thanks the two logfiles from healthcare and the new tool.

<u>The typical Characteristics of Logfiles from Healthcare</u>

The logfile from the healthcare have typical characteristics, which are known. The typical characteristics of medical treatments are about the following up between related activities. In the healthcare treatments can started up without waiting of the diagnosis of the previous treatment. Another characteristic in the medical world is that further treatments are not necessary, because the diagnosis of a previous treatment is positive. By using the new tool, we can quantify these circumstances. These are related to the number of activation of the forks and its directly connected activities. There are three options:

- The number of presence of the fork in the event log is higher than the total number of its direct related activities. The fork starts up fewer activities. This means that the forks cancel some further activities. For instance in the healthcare, canceling of further treatments could be the positive diagnosis (see case study 1 and 2).
- The number of presence of the fork in the event log is equal to its direct connected activities. The fork enables as much as number of activities. For instance, the measures of a treatment will be followed by an analysis and registration of this measure.
- The number of presence of the fork in the event log is lower than its direct connected activities. In this situation, the connected activities are starting up independently of their previous activity. For instance, in the medical world treatments can starting without waiting of the diagnosis of a previous treatment. The reason for this could some suspicions of disease based on other factors, which are not modeled (see case study 1 and 2).

Conclusion

The new tool discovered the characteristics of less structured processes in the logfile, which recorded information about various measures on stroke patients. *SJI* found also the characteristics, which are typical of event logs from the healthcare. The measure *SF36* started it up without waiting of the diagnosis of its previous measures. The measures *hamilton anxiety* and *hamilton depression* canceled sometimes further measures. The new tool discovered also the characteristics of forks which include loops of length one and the two types of splits and joins in this process model (i.e. XOR and the vague "OR"). This all, *SJI* confirms the given information about this process model. Finally, *SJI* illustrates that a fork (split and join) might have two different characteristics.

Next section describes more the additional properties of *SJI*.

5.4 Statistical Analysis of the new Tool

This section is about the applications of the new tool. The knowledge, which is discovered by *SJI* can be used for further statistical analysis. The statistical analysis, which will be described in the following subsections, uses the first example (section 3.1). This example contains both types of processes (i.e. structured and less structured).

This section exists of two parts.The first subsection illustrates some additional analysis, which is possible by using the information from the four tables (see Appendix A.12). The second subsection is the test of the new tool by using some noise in the event log. The goal is to see how the new tool behaves to noise in the event log. For this test, we use also the first example.

5.4.1 Some additional analysis

The knowledge of the four tables can be used for providing additional information about the process model. We show five applications of *SJI*. Let's start with the first one.

The Contribution of individual Activities to Pattern Frequencies

First, we can determine the contribution of an individual activity of the most occurrence pattern frequency. This is also known as the conditional probability[12]. For the calculation of this

[12] The conditional probability of activity Y given X, denoted as $P(Y \mid X)$ is: $\frac{P(X \wedge Y)}{P(X)}$.

measure, we need the first two statistical tables (i.e. *"Statistical Measures for Activities"* and *"Combination table"* in Appendix A.12). For instance, let's calculate the conditional probability of activity *A* to the combination of the activities *A*, *C*, and *D* in the first example. The individual activity *A* (i.e. 1,120 times) and the combination *ACD* (i.e. 2,244 times) are related to the split *X1*. The conditional probability of *A* is equal to 1,120/2,244 = 0.499. In other words, the contribution of activity *A* is almost 50% of the pattern combination *ACD*.

We can also look to the contribution of an individual activity in the patterns frequency, which contain the individual activity. For instance, the conditial probability of individual activity *A* in relation to the combinations *ACD*, *AB*, and *A*. These combinations of activities can be started up after the split *X1* in example 1. This can also be done for the remaining activities (i.e. *B*, *C*, and *D*), which can be enabled by the split *X1*. Figure 23 shows the result in a graph. Number of activities 1 is the pattern frequency *A*, 2 is *AB*, and 3 is *ACD*. This graph illustrates clearly that the contribution of *A* increases with the number of its combinations. The remaining activities *B*, *C*, and *D* have only singular points, e.g. *B* has (2,1). The contribution for *B* in the pattern frequency *AB* is 1 (or 100%). One remark, *C* and *D* have the same singulary point (i.e. (3,1)).

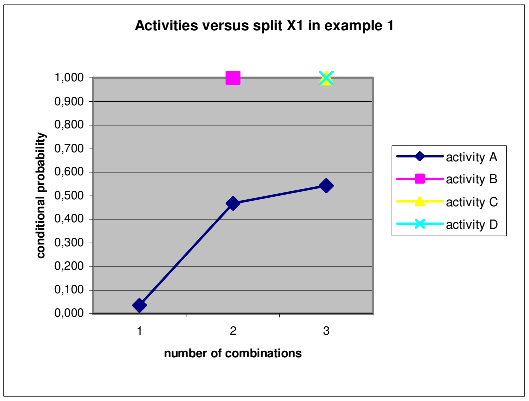


Figure 23: The conditional probabilities versus number of combinations.

Probabilities of successively Activities

Second, it is possible to calculate the probabilities between successively activities in a process model. It indicates the probabilities of certain paths in a process model. Hence, we use the first statistical table or the DG including the probabilities (i.e. *p(s)* and *p(j)*). Let's calculate, for instance the probability of the performance of activity *Y1* after the enabling of split *X1* via activity *A* without repeating of *A*. This is:
$P(A \to Y1 | X1 = 1) = P(X1 \wedge A) \cdot P(A \wedge Y1) = 0.742 \cdot 0.376 = 0.279$ (rounded in three decimals).

New Rules in a Group of Activities

Third, it is possible to derive some more interesting rules between activities in a group. Therefore, we use the fourth table (i.e. *"Apriori table"*). This table shows only the first 10 rules between activities (see Appendix A.12). Based on this information, we might determinate some more relations between activities. For instance, the derivation of one more relation after the split (activity *X1*). By using the combination of rule 7 and 9 (transivity property[13]) from the *Aprioritable* leads us to the following interesting rule: *A=1 and C=0 and D=0 (1,124)* ➔ *X1=0 (1,124) with confidence 1,124/1,124=1.*

[13] This is one of the (predicative) logical properties, which says that $(A \to B) \wedge (C \to B) \Leftrightarrow (A \wedge C) \to B$

Number of Activities versus Pattern Frequencies

Fourth, the analysis of the fork in relation to its number of connected activities and the number of combinations. The number of the activities, which are connected to a fork, is related to the pattern frequency of the fork. The combinations can vary between *1* and 2^n, where *n* is the number of the direct related activities to a fork.

The number of connected activities versus the number of combinations to a fork gives insight in the heterogeneity of a fork. For instance, an AND-split (i.e. split *X2* in example 1) is homogeneous. It has only one combination (i.e. in example 1 the combination *E F G*). An XOR-split is a little bit less homogeneous, e.g split *A* in example 1. It has some more combinations namely 2 (i.e. *A* and *Y1*). In case of a vague "OR", the fork might be more heterogeneous. The split *X1* is an OR and is connected to 5 activities and has 4 combinations. This split has a low heterogeneity. A better example of a heterogeneous fork is the join *OK* in the process model of the first medical consult for womb cancer (case study 1). The join can be enabled by 6 treatments and has 16 different pattern frequencies of possible 64. Figure 24 illustrates the number of activities versus the pattern frequencies to the forks in the process model of example 1. This graph shows a (weak) positive relation (*R-square = 0.3937*) between the number of activities and the pattern frequency according the first example (section 5.2.1).

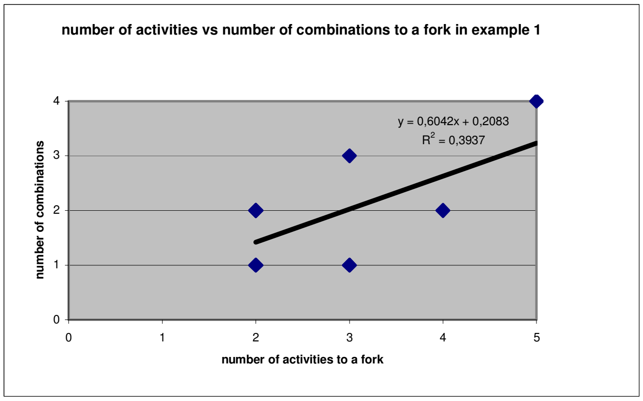


Figure 24: The number of activities to a fork vs number of combinations.

Factor Analysis of the selected Metrics

Fifth, the factor analysis of the selected metrics between activities. The factor analysis gives insight in the degree of dependency between these metrics (i.e. support, correlation and IS).

	support	correlation	IS
support	1	0.910	0.930
correlation	0.910	1	0.976
IS	0.930	0.976	1

Table 9: The correlation between the 3 selected measures.

Table 9 shows the correlation between the three selected measures, which are strongly positive related to each other in example (see also article of [34]). The correlation is based on the three metrics values of the activities to their forks in the process model of example 1 (section 5.2.1) and is calculated by SPSS[14]. As already mentioned in section 3.3.3, the selected metrics are

[14] SPSS 14.0 for Windows (release 14.0.2, 21st April 2006) is a statistical software tool.

complementing each other in some circumstances. In these situations, we need *more* measures to get better insight in the behaviour between activities.

Next section looks to another point of view to the new tool. It is interesting to known more about the strengthness of the tool. This gives insight in what the new tool is still suitable for analysis of the forks in the process model. In this case, we analyse the new tool by the event log from example 1 with some noise, which will be described in the following subsection.

5.4.2 Statistical Analysis with Noise

We create some noise in the event log from example 1. The *ProM* tool has an option for creating noise in the event log. After the selection of 5% noise, the tool will be 5% of the traces of the original event log deformed. This can be performed by removing one or more events out of the trace, interchanging two events, or by removing some activities from the begin or the end of a trace. The types of deformation are randomly distributed over the traces of the original event log [89].

We start with 5% noise and increase it with 5% in the following steps to 25%. The reasons for this, is to see the changes in behaviour of the selected metrics of *SJI* in particular to less structured processes.

Let's look mainly to one fork in the process model of example 1. We selected the split *X1*, which enables the activities *X1*, *A*, *B*, *C*, and *D*. The split has the characteristics of less structured (see also section 5.2.1). We analyze the behaviour of the activities (i.e. *X1*, *A*, *B*, *C*, and *D*) to the split *X1* from three points of view, i.e. individual activities, pattern frequency, and the selected metrics between activities.

The Behaviour of the individual Activities

First, the probabilities of the individual activities to split *X1*.The probabilities of the individual activities to split *X1* seem to be stable during every step. Figure 25 illustrates the behaviour of the individual activities to split *X1* versus percentage of noise in the event log.

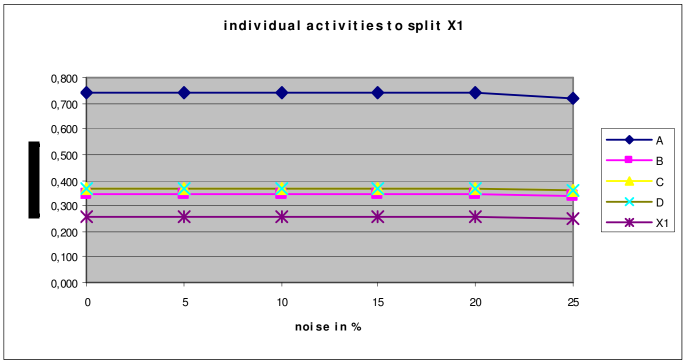


Figure 25: Noise vs probabilities of individual activities.

The Behaviour of the Pattern Frequencies

Second, the pattern frequencies to split *X1*. Hence, the probabilities of the combination of the most important activities (i.e. *ACD, AB*, *X1*, and *A*) to split *X1* seem also almost to be unchanged every step. After the first step of increasing of noise, some more combinations are observed. But, their contribution are very small (e.g. *ABCD* with a probability 0.001, etc.)

comparing the original four combinations. Figure 26 shows the behaviour of the four original combinations (i.e. *ACD, AB*, *X1*, and *A*) versus the percentage of noise in the event log.

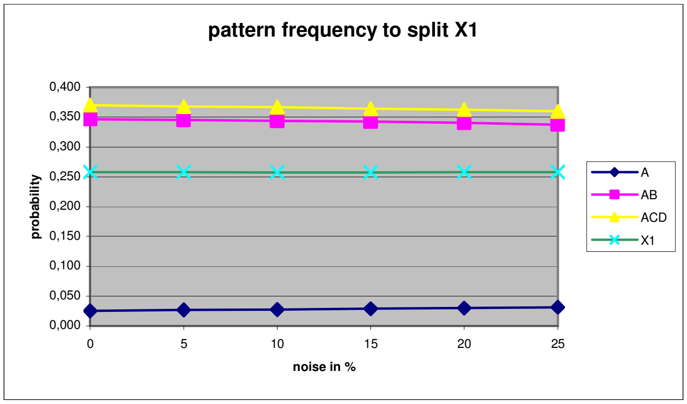


Figure 26: Noise vs combination of activities.

The Behaviour of the selected Metrics between Activities

Third, the three selected metrics (i.e. support, correlation, and IS) between activities. Hence, we describe only between the activities *A* and *B* in relation to split *X1*. The behaviour of these measures seems to be stable during every step. Figure 27 illustrates the behaviour of the three measures between *A* and *B* versus the percentage of noise in the event log. The behaviour of the selected metrics between the remaining pairs (e.g. *A* and *C*, etc.) to split *X1* is the same as it between the activities *A* and *B*.

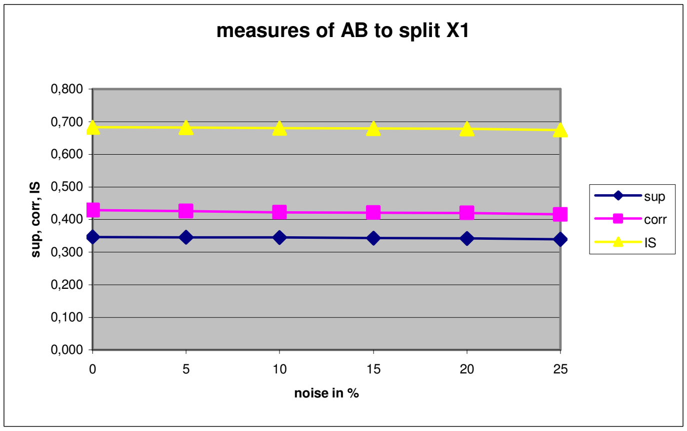


Figure 27: Noise vs the three selected measures.

The correlation between these metrics confirms this finding. The correlation between three metrics (i.e. support, correlation, and IS) is almost unchanged and is still very high, when the event log contain some noise.

	support	*correlation*	*IS*
support	*1*	*0.937*	*0.979*
correlation	*0.937*	*1*	*0.967*
IS	*0.979*	*0.967*	*1*

Table 10: The correlation of three metrics based on split X1 with 25% noise

Table 10 on page 58 shows the correlation between the measures based on the split *X1* given an event log with 25% noise. The correlation is based on the three metrics values of the activities *X1*, *A*, *B*, *C,* and *D* to the split *X1* and is calculated by SPSS. This table differs a little bit to the correlation of the complete event log without noise (see Table 9). It indicates a strong positive relation these measure despite presency of noise in the event log. Also in this situation, the remark about these metrics is *still valid* (see section 3.3.3).

Conclusion
The metrics (i.e. probability, support, correlation, and IS), used in the new tool, is insensible for a low percentage of noise in the event log.

5.5 Conclusions

The new tool is validated with two types of data, i.e. internal and external. The internal data are the artificial logs. External data are the event logs from various hospitals. *SJI* is mainly developed for analysis of less structured processes, but can also be applied for structured processes. It distinguishes clearly both types of processes by the statistical tables. The other benefits of the new tool are the illustration of the characteristics of the forks in the DG and tables from four points of view. The new tool has shown all characteristics of the two artificial data. It succeeded this test.

The characteristics of the two artificial data were discovered by the new tool. *SJI* discovered the typical characteriscs of the processes in a hospital. The two external logfiles are more complex than the artificial data. The new tool has been validated successfully for the real logfiles. The used logfiles from the healthcare has some more characteristics at home, which were unknown. In case, the new tool has shown its additional value.

The validation has also shown the approach to the tables of *SJI* in practice. The first two tables of the new tool give in most cases already insight in the characteristics of the forks in the model. The first table, called *"Statistical Measures for Activities"* distinguishes the first characteristics of the type of split and join in the process model (i.e. XOR, OR, AND). The second table, called *"Combination table"* illustrates the pattern frequencies of a fork. It gives more insight in the complexity (heterogeneity) of a fork. The third table (i.e. *"Statistical Measures for related Activities"*) discovers the degree of dependency between activities. The fourth table (*"Apriori table"*) discovers the relation between groups of activities. The last two mentioned tables are more necessary, if the process model is more complex like processes in a hospital. These tables give also more information about the characteristics of the fork and support the findings by the first two tables. But, they are more useful to discover (some) more interesting points of the process model. This is also the strengthness of the new tool.

The four tables can be used as knowledge for further (statistical) analysis of a process model. We summarize five interesting applications of the tool, which were performed.

First, the contribution of individual activities in the patterns frequency can be calculated. This is the conditional probability of an activity. It can be determinated by the frequency in the tables of individual activities and combination table (table I and II). This is the probability of an individual activity versus to its patterns frequency. The pattern frequencies contain the individual activity (see also Appendix A.16).

Second, the probabilities of successively activities. It indicates the probabilities of certain paths in a process model.

Third, more interesting rules might be discovered between activities in a group. The *Apriori* algorithm generates as default the best 10 rules between activities. As a consequence of this table, it makes possible to find more relations between activities (see also Appendix A.16).

Fourth, the analysis of the number of connected activities and the number of combinations to a fork. The number of the activities, which are connected to a fork, is related to the pattern frequency of a fork. The number of the directly related activities versus the number of combinations to a fork gives insight in the heterogeneity of the fork.

Fifth, the factor analysis between the selected metrics between activities (e.g. support, correlation and IS). The factor analysis illustrates the degree of dependency between these measures (see also Appendix A.16).

The five above mentioned points are additional applications of the new tool. This is also the answer of the third research question: *"Which kind of information can be discovered by the new tool between the activities of less structured processes?"* Some of these points can also be applied for the development of a general table in *SJI* (see appendix A.4).

SJI is still reliable, if the event log contains some noise. But the new tool has also its limitations. In some circumstance, *SJI* has fewer benefits for analyzing of the forks. These are the case, when:

- The number of cases (traces) of the event log is low. This is on a higher aggregation level, but this can also play on a lower level.
- The number of occurring of the fork is low. See example, the join *body research under anesthesia* in case study 1.
- The number of the direct connected activities to a fork is low. See example, the join *radiotherapy* in case study 1.
- The event log contains a very high percentage of noise (more than 50%).

This makes it difficult to get good insight in the behaviour of the forks in the model. The analysis will be less reliable. The above mentioned points are also the answer of the fourth research question: *"To what extend does the knowledge discovery tool validate internal and external data of less structured process?"*

6. Conclusions and recommendations

The last chapter of this master thesis is about the conclusions and the recommendations for future work on process mining of less structured processes. All these are based on the findings of the new tool, which are observed during the validation of the new tool. It also gives answer on the last research question, namely *"To what extend is the new tool suited in practice?"*

6.1 Conclusions

This is about the conclusions of the new tool, which also includes the last research question. It has more a pragmatic view, which is not less important. It is also good starting point for this section, which is: *"To what extend is the new tool suited in practice?"* This question is binary, i.e. the additional insights in the medical field by *SJI* and the contribution of the *SJI*. The insights of this research field by this new tool gives possible (new) directions for improvement.

SJI is one of the first methods, which can be used for both type of processes, i.e. structured and less structured. But their additional value and strength is the analyzing of the more complex and less structured processes, which take place in the hospitals. It is a discovery and analysis tool, which applies the statistical measures for providing insight into the real world [80]. The new knowledge discovery tool is validated in practice by the data from several hospitals in the Netherlands and Italy, and illustrates its possible contribution. It can inform other disciplines in the healthcare the degree of dependency between medical treatments, the most pattern frequency of treatments and the relation between treatments.

The new knowledge and discovery tool has some flexibility in the representation of the DG. The default parameters of HM can be easily changed, if necessary. For instance, the number of cases in an event log from the hospital is small and the number of a certain treatment in a hospital is very low. This treatment is related to or more other medical activities. By changing the default parameter of the positive observations (i.e. 10) to a lower value, we get a proper a process model that better fits the real world. All this, the *SJI* has the opportunity to analyze once more the characteristics of the forks in the new situation.

The new tool gives insight in the relations between the successive activities, e.g. the degree of dependency and also which activities are dominated in the process model. This gives opportunities for providing a solution in several fields like logistics, human resources and organization. For instance, the degree of dependency gives a prediction indication for the supply of medicals, facilities of medical systems, etc. *SJI* might also illustrate the bottlenecks of the current processes. The (selected) metrics in *SJI* give information how often certain activities are performed (thus their probability) and in relation to their direct related activities. The manager of P&O has the possibility to schedule personnel in the time, based on the statistical information of *SJI*. The final purpose is to improve the medical processes and to competitive in their health-care market by reducing costs [79]. Furthermore, also on the governmental side and on the side of the healthcare insurance companies, rising pressure is put on hospitals to work in the most efficient way possible. Another point is that in the future, an increase in the demand for care is expected by ageing of the society especially in the Netherlands. The new knowledge discovery tool can contribute to all these (practical) goals in the near future. The above mentioned points are the answer of the last research question.

6.2 Recommendations

This section describes the recommendations for the further research in particular less structured processes. The research project is characterized by a general approach of the activities in a process model. The degree of dependency between activities is analyzed after the setup of the DG. But, in the healthcare the activities are related by the results of the measures. Some of the event logs contain the information about the medical measurements. Therefore, it would be interesting to analyze the measures of these scales and their outcomes to relate to the following activities in the process model.

The new tool is a two-step mining tool. The first step of *SJI* is based on the first step of the HM algorithm, which determines the process model. This choice is not essential as starting point of the *SJI*. The first step might also be performed with other techniques, like statistical metrics in combination with hypothesis or decision tree learning (chapter 3, pages 52-80 of [63]) .The decision tree learning technique uses the entropy for measuring homogeneity. Future work could be the study of several PM techniques for the substitution of the current first step of the new tool. After the setup, it is interesting to compare the mutual results and develop a benchmark of the several used mining techniques.

The second step of the new tool is based on the two dimensional binary matrix. The statistical analysis is characterized of the presence of the activities in a row of the data set. The selected metrics in this master thesis apply the (positive) Boolean association rules of mining. Future work like two dimensional quantitative association rules, can give more insight range of values and amount of these items (pages 250-274 of [59]). This technique gives more insight in the characteristics values of an activity.

The new knowledge discovery tool has currently three metrics (i.e. support, correlation, and IS). *SJI* can be extended with more and/or other interestingness measures for association patterns (i.e. currently 21), if necessary [34]. Also the metrics of the negative association rules can be added to the new tool. The extension is very easy to add in the new tool. This might be interested for discovering of other kind of behaviour between activities in a process model.

The new tool has no opportunity to export the statistical values of the metrics in a file. Generating an export file as an option in the submenu of *SJI* would be nice. In case, the export file can be directly applied by a statistical software tool like SPSS for further analysis of the fork's characteristics in the process model.

The metrics in the third table (i.e. support, correlation and IS) gives insight in the characteristics of two activities. It would be interesting to develop metrics for three or more activities. These measures illustrate directly the behaviour of the related groups and their forks in the model. Another option is the development of a method, where the measure can be applied for analyzing of two groups.

The process models of less structured processes can be analyzed for latent variables by multivariate analysis. Latent variable can be simply described as variables that we have not directly measured, but that we can directly observe. It is another method of validation of process models. These type of models need to be hypothesized for unmeasured latent variables [76,77].

As already mentioned, the *SJI* has also to certain extent flexibility as a consequence of the default parameters of HM. Although easily changed, finding the best threshold value can be difficult for very complex processes. In this situation, an overview between the parameters of the HM and the statistical measures gives insight in the changes of these values.

SJI has no time dimension at home. Activities are time dependently. Further statistical analysis on the duration between the related activities gives insight in their degree of influence in time and also other metrics can be applied for analysis, like throughput time. This aspect gives insight in possible bottlenecks of the current processes, which gives indication for improvement of the processes. This can only be performed in combination with the analyses plug-in PSDA.

The new tool will have more benefits, when some exchange of information with the end users take place. The end users of this new tool might give some interesting feedback, which can lead some more improvement of this tool. It took only once place with one PhD-student and a post doc employee of the TU/e. Both employees have experience in the field of healthcare at AMC in Amsterdam. The information exchange between these people stimulates the research activities in this domain. More is necessary, but it was in certain way out of the scope of this research project.

SJI is validated with two practical examples from the medical world. The new tool can also be applied for analyzing of processes from other organizations. This increases the name of *ProM* and the utilization of this PM tool.

Finally, flexible processes take place in practice and have its interests in the business world. It is still an open field in PM and has a lot of research opportunities for the coming years. *SJI* has its own point of view to this topic, i.e. statistics, and it is a challenge to find more direction of solutions in this complex, but much more challenging topic for PM.

7. References

[1] *A. K. de Medeiros, PhD thesis "Genetic Process Mining", Beta Research School for Operations Management and Logistics, Technische Universiteit Eindhoven, November 2006.*

[2] *A.K. de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Using Genetic Algorithms to Miner Process Models: Representation, Operations and Results. BETA Working Paper Series, WP124, Eindhoven University of Technology, Eindhoven, The Netherlands, 2004.*

[3] *A.J.M.M. Weijters, "Heuristics Miner", Department of Technology Management, Eindhoven University of Technology.*

[4] *Douglas C. Montgomery, George C Runger, "Applied Statistics and Probability for Engineers", Third Edition, John Wiley & Sons, Inc. , 2003.*

[5] *J.E. van Aken, J.D. van der Bij, J.J. Berends, Collegedictaat Bedrijfskundige Methodologie 1Z350, Collegejaar 2003/2004, Eindhoven University of Technology, department Technology Management, Eindhoven, The Netherlands, July 2003.*

[6] *B.F. van Dongen, A.K. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters and W.M.P. van der Aalst, The ProM framework: A new era in process mining tool support, 26th International Conference on Applications and Theory of Petri Nets (CATPN 2005), G. Ciardo and P. Darondeau, LNCS 3536, pages 444-454, 2005. Springer Verlag Berlin Heidelberg 2005.*

[7] *LijleWen, W.M.P. van der Aalst, Jianmin Wang, and Jiaguang Sun, "Mining Process Models with Non-Free-Choice Constructs", pages 1-32.*

[8] *A.K. de Medeiros, A.J.M.M. Weijters, ProM Framework Tutorial, Eindhoven University of Technology, Eindhoven, The Netherlands, November 2006.*

[9] *A.K. de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst, "Genetic Process Mining: A Basic Approach and its Challenges" in C. Bussler et al., editor, BPM 2005 Workshops (Workshop on Business Process Intelligence), volume 3812 of Lecture Notes in Computer Science, pages 203-215, Springer-Verlag, Berlin, 2006.*

[10] *A.J.M.M. Weijters, W.M.P. van der Aalst, "Rediscovering Workflow Models from Event-Based Data using Little Thumb", Eindhoven, University of Technology, department Technology Management, Eindhoven, The Netherlands, pages 1-24.*

[11] *W.M.P. van der Aalst and K. van Hee, Workflow Management, MIT Press, 2002.*

[12] *A.K.A. de Medeiros, B.F. van Dongen, W.M.P. van der Aalst, and A.J.M.M. Weijters,"Process Mining: Extending the α-algorithm to Mine Short Loops", Department of Technology Management, Eindhoven University of Technology, Eindhoven, The Netherlands, pages 1-25.*

[13] *A.K.A. de Medeiros, W.M.P. van der Aalst, and A.J.M.M. Weijters,"Workflow mining: Current status and future directions." In Robert Meersman, Zahir Tari, and Douglas C. Schmidt, editors, On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, volume 2888 of LNCS, pages 389-406, Springer Verlag, 2003.*

[14] *A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros, "Process Mining with the Heuristics Miner Algorithm", Department of Technology, Eindhoven University of Technology, Eindhoven, The Netherlands.*

[15] *W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters, "Workflow Mining: A Survey of Issues and Approaches", Data and Knowledge Engineering, 47(2): pages 237-267, 2003.*

[16] *W. Reising and G. Rozenberg, editors. Lectures on Petri Nets I: Basic Models, volume 1491 of Lecture Notes in Computer Science. Springer Verlag, Berlin, 1998.*

[17] *J. Scott, Social Network Analysis. Sage, Newbury Park CA, 1992.*

[18] W.M.P. van der Aalst and M. Song,"Mining Social Networks: Uncovering interaction patterns in business processes" In J. Desel, B. Pernici and M. Weske, editors, International Conference on Business Process Management (BPM 2004), volume 3080 of Lecture Notes in Computer Science, pages 244-260. Springer-Verlag, Berlin, 2004.

[19] G. Keller and T. Teufel. SAP R./3 Process Oriented Implementation. Addison-Wesley, reading MA, 1998.

[20] E.E.M. van Berkum, A. di Bucchianico, "Statistisch Compedium", Department Mathematics and Informatics,,University of Technology, Eindhoven, The Netherlands, 2004.

[21] A.J.J.M. Weijters, W.M.P. van der Aalst, "Process Mining: Discovering Workflow Models from Event-Based Data". In Kröse, B. et. Al, (eds.): Proceedings 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'01), 25-26 October 2001, Amsterdam, The Netherlands, pages 283-290.

[22] L. Maruster, A.J.M.M. Weijters, W.M.P. van der Aalst, A. van den Bosch, "Process Mining: Discovering Direct Sucessors in Process Logs",In S. Lange, K. Satoh, and C.H. Smith (Eds.): DS 2002, LNCS 2534, pages 364-373, 2002, Springer-Verlag Berlin, Heidelberg, 2002.

[23] A.J.J.M. Weijters, W.M.P. van der Aalst, "Rediscovering Workflow Models from Event-Based Data using Little Thumb", Department of Technology, Eindhoven University of Technology, Eindhoven, The Netherlands.

[24] W.M.P. van der Aalst, M. Song, "Discovering Social Networks from Event Logs".

[25] W.M.P. van der Aalst, and B.F. van Dongen, "Discovering Workflow Performance Models from Timed Logs", In Y. Han, S. Tai, and D. Wikarski, editors, International Confiderence on Enginering and Deployment of Cooperative Information Systems (EDCIS 2002), volume 2480 of Lecture Notes in Computer Science, pages 45-63, Springer Verlag, Berlin.

[26] Cook, J. E., Wolf A. L., "Discovering Models of Software Processes from Event-Based Data", ACM Trans. Softw. Eng. Methodology, 7, 1998, pages 215-249.

[27] Agrawal, R., Gunopulos, D. Leymann, F., "Mining Process Models from Workflow Logs", In: Proceedings of the 6th International Conference on Extending Database Technology, Springer-Verlag, 1998, pages 469-483.

[28] Casati, F. Ceri, S., Pernici, B., Pozzi, G., "Workflow Evolution", In: International Conference on Conceptual Modeling / the Entity Relationship Approach, 1996, pages 438-455, Cottbus, Germany, October 1996.

[29] Extensible Markup Language (XML). http://www.w3.org/XML/. Last revisited on 4th July 2007.

[30] Homepage Process Mining TU/e. http://www.processmining.org. Last revisited on 4th of July 2007.

[31] Kurt Jensen, "Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use", volume 1, Second Edition, Second corrected printing 1997, Springer-Verlag, Berlin Heidelberg New York.

[32] W.M.P. van der Aalst, syllabus "Process Modeling" (1BB30), Department of Technology Management, Eindhoven University of Technology, The Netherlands, March 2006.

[33] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, "using association rules for product assortment decisions: A case study." In Proc. of the Fifth International Conference on Knowledge Discovering and Data Mining, 1999.

[34] P. Tan, V. Kumar, J. Srivastava, "Selecting the Right Interestingness Measure for Association Patterns", Technical Report 2002-112, Army High Performance Computing Research Center, 2002.

[35] A. Savasere, E. Omiecinski, and S. Navathe, "An efficient algorithm for mining

association rules in large databases". In U. Dayal, P.M.D. Gray, and S. Nishio, editors. "Proceedings 21th International Conference on Very Large Data Bases", Morgan Kaufmann, pages 432-444, 1995.

[36] *M.J. Zaki, "Scalable algorithms for association mining", IEEE Transactions on Knowledge and Data Engineering, 12(3), pages 372-390, May/June 2000.*

[37] *P. Shenoy, J.R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D. Shah, "Turbo-charging vertical mining of large databses". In W. Chen , J.F. Naughton, and P.A. Bernstein, editors. "Proceddings of the 2000 ACM SIGMOD International Conference on Management of Data", volume 29(2) of SIGMOD Record. ACM Press, pages 22-33, 2000.*

[38] *S. Orlando, P. Palmerini, R. Perego, and F. Silvestri, "Adaptive and resource-aware mining of frequent sets". In V. Kumar, S. Tsumoto, P.S. Yu, and N. Zhong, editors, "Proceedings of the 2002 IEEE International Conference on Data Mining", IEEE Computer Society, 2002. To appear.*

[39] *S. Morshita and A. Nakaya, "Parallel Branch-and-Bound Graph Search for Correlated Association Rules". In M.J. Zaki, C.-T. Ho editors, "Large-Scale Parallel Data Mining", LNAI 1759, pages 127-144, Springer-Verlag Berlin Heidelberg, 2000*

[40] *P.-N. Tan and V. Kumar, "Interestingness Measures for Association Patterns: A Perspective".*

[41] *B.Goethals, "Survey on Frequent Pattern Mining", pages 1-43.*

[42] *R. Agrawal, T. Imielinski, and A.. Swami, "Mining association rules between sets of items in large database". In Proceedings of ACM SIGMOD, pages 207-216, May 1993.*

[43] *R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules". In Proceedings of VLDB Conference, pages 487-499, 1994.*

[44] *S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data". In Proceedings of ACM SIGMOD International Conference on Management of Data (ACM SIGMOD '97), pages 265-276, 1997.*

[45] *IBM. Intelligent Miner Handbook, 1999.*

[46] *S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations". In Proceedings of 1997 ACM-SIGMOD International Conference on Management of Data, Tucson, AZ, 1997.*

[47] *S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to dependencies rules". Data Mining and Knowledge Discovery, 2(1), pages 39-68, 1998.*

[48] *http://www.citi.qut.edu.au/yawl, The YAWL project homepage, last revisited on 4th April 2007.*

[49] *R. Cooley, C. Clifton. Topcat: Data mining for topic identification in a text corpus. In Proceedings of the 3rd European Conference of Principles and Practice of Knowledge Discovery in Databases, 1999.*

[50] *Medical data from Università degli Studi Pavia,Italy, IRCCS C. Mondino, Pavia, Italy, IRCCS Humanitas, Rozanno, Italy.*

[51] *FI Mahoney, D. Barthel, Functional evaluation: Barthel Index, Md State Med J 1965, nr 14, pages 56-66.*

[52] *G. Keller, M. Nüttgens, A.W. Scheer, Semantische Prozessmodellierung auf der Grundlage, "Ereignisgestreuerter Prozessketten (EPK)". Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany (1992).*

[53] *W.M.P. van der Aalst, "The application of petri nets to workflow management", Journal of Circuits, Systems, and Computers, 8(1), pages 21-66, 1998.*

[54] *H. Schildt, "Java, The Complete Reference", 7th edition, McGrawHill, 2007.*

[55] *Medical data from the academic hospital at Amsterdam "AMC", March 2007.*

[56] *G. Micieli, A. Cavallini, S. Quaglini, "Guideline Complicance improves Stroke Outcome", A Preliminary Study in four districts in the Italian Region of Lombardia. Stroke 2002;33:1341-7.*

[57] *http://www.w3.org/XML/ . Homepage of XML. Last revisited on 22thJune 2007.*

[58] *R. Agrawal, D. Gunopulos and F. Leymann, "Mining Process Models from Workflow Logs", Research Report RJ 10100 (91916), IBM Almaden Research Center, San Jose, California (available from http://www.almaden.ibm.com/cs/quest), December 1997.*

[59] *I.H. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques", 2nd edition, Morgan Kaufmann, San Fransisco, 2005.*

[60] *http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/itemset_prog1.html, Website about Apriori Implementation, last revisited on 23rd June 2007.*

[61] *G.F. Goldfarb, "XML Handbook", Prentice Hll, 2003.*

[62] *J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan, Kaufmann 2006.*

[63] *T.M. Mitchell, "Machine Learning", McGraw-Hill, 1997.*

[64] *C. Ghezzi, M. Jazayeri, D. Mandriok, "Fundamentals of Software Engineering", 2nd Edition, 2003.*

[65] *http://wiki.daimi.au.dk/cpntools/cpntools.wiki, Homepage of CPN Tools, last revisited on 27th June 2007.*

[66] *http://en.wikipedia.org/wiki/Thorax. Website about thorax. This is part of the free encyclopedia. Last revisited on 28th June 2007.*

[67] *http://en.wikipedia.org/wiki/Abdomen. Webstite about adomen. This webpage is part of the free encyclopedia, called Wikipedia. Last revisited on 28th June 2007.*

[68] *C.W. Günther and W.M.P. van der Aalst, "Fuzzy Mining – Adaptive Process Simplification Base don Multi-Perspective Metrics".*

[69] *http://ga1717.tm.tue.nl/user/christian/research/fuzzyminer, Homepage of Fuzzy Miner, last revisited on 29th June 2007.*

[70] *S. Gupta, Master thesis "Workflow and Process Mining in Healthcare", Eindhoven University of Technology, department of Mathematics and Computer Science, Eindhoven, May 2007.*

[71] *The Association Rules Miner documentation in the ProM tool, version 4.1, 15th April 2007, last revisited on 29th June 2007.*

[72] *R.S. Mans, M.H. Schonenberg, M. Song, W.M.P. van der Aalst, "Process Mining in Healthcare, A Case Study", June 2007.*

[73] *http://en.wikipedia.org/wiki/Atrial_flutter. Website about atrial flutter. This webpage is part of the free encyclopedia, called Wikipedia. Last revisited on 2nd July 2007.*

[74] *http://www.holter.nl/files/AFL.pdf. Website about atrial flutter. Last revisited on 2nd July 2007.*

[75] *R. Lenz, T. Elstner, H. Siegele, and K. Kuhn, "A Practical Approach to Process Support in Health Information Systems". Journal of the American Medical Information Association, 9(6), pages 571-585, 2002.*

[76] *P.M. Bentler, "Multivariate Analysis with latent variables: Causal Modeling", Annual Reviews Inc., Psychology, 31;pages 419-456, 1980.*

[77] *B. Shipley, "Cause and Correlation in Biology", A User's Guide to Path Analysis, Structural Equations and Causal Inference, Cambridge 2000.*

[78] *Performance Sequence Diagram Analysis documentation in the ProM tool, version 4.1, 15th April 2007, last revisited on 3rd July 2007.*

[79] *K. Anyanwu, A. Sheth, J. Cardoso, J. Miller. K. Kochut, "Healthcare Enterprise Process Development and Integration". Journal of Research and Practice in Information Technology, 35(2), pages 83-98, 2003.*

[80] L. Maruster, W.M.P. van der Aalst, A.J.J.M. Weijters, A. van den Bosch, W. Daelemans, "Automated Discovery of Workflow Models from Hospital Data".

[81] F. Bodon, "Surprising results of trie-based FIM algorithms", pages 1-11.

[82] B. Goethals, M.J. Zaki, "FIMI'03: Frequent Itemset Mining Implementation", pages 1-13.

[83] F. Bodon, "A fast APRIORI implementation", pages 1-10

[84] R. Vaarandi, "A-Breadth-First Algorithm for Mining Frequent Patterns from Event Logs",A. Aagesen et al. (Eds.): INTELLCOMM 2004, LNCS 3283, pages 293-308, 2004.

[85] L. Zhung-Xun and S. Man-Kwan, "Algorithms for Discovery of Frequent Superset, Rather than Frequent Subset", Y. Kambayashi et al. (Eds.): DaWaK 2004, LNCS 3181, pages 361-370, 2004, Springer-Verlag Berlin Heidelberg 2004.

[86] A. Ceglar and J.F. Roddick, "Association Mining", ACM Computing Surveys, Vol. 38, No.2, Article 5, pages 1-42, July 2006.

[87] M.H. Maraghny and A.A. Mitwaly, "Fast Algorithm for Mining Association Rules", AIML 05 Conference, pages 1-5.

[88] http://www.cs.waikato.ac.nz/ml/weka/, Homepage of Weka, last revisited on 12th August 2007.

[89] Documentation about the Heuristics Miner, ProM tool, version 4.1, last revisited on 14th August 2007.

[90] Peter A. Flach, Nicolas Lachiche. Confirmation-Guided Discovery of First-Order Rules with Tertius, Machine Learning, Volume 42, Issue 1 - 2, Jan 2001, Pages 61 - 95, DOI10.1023/A:1007656703224, URL http://dx.doi.org/10.1023/A:1007656703224, last revisited on 15th August 2007.

[91] www.lib.ncsu.edu/theses/available/etd-05302006-170618/unrestricted/etd.pdf, information about the Tertius algorithm, last revisted on 15th August 2007.

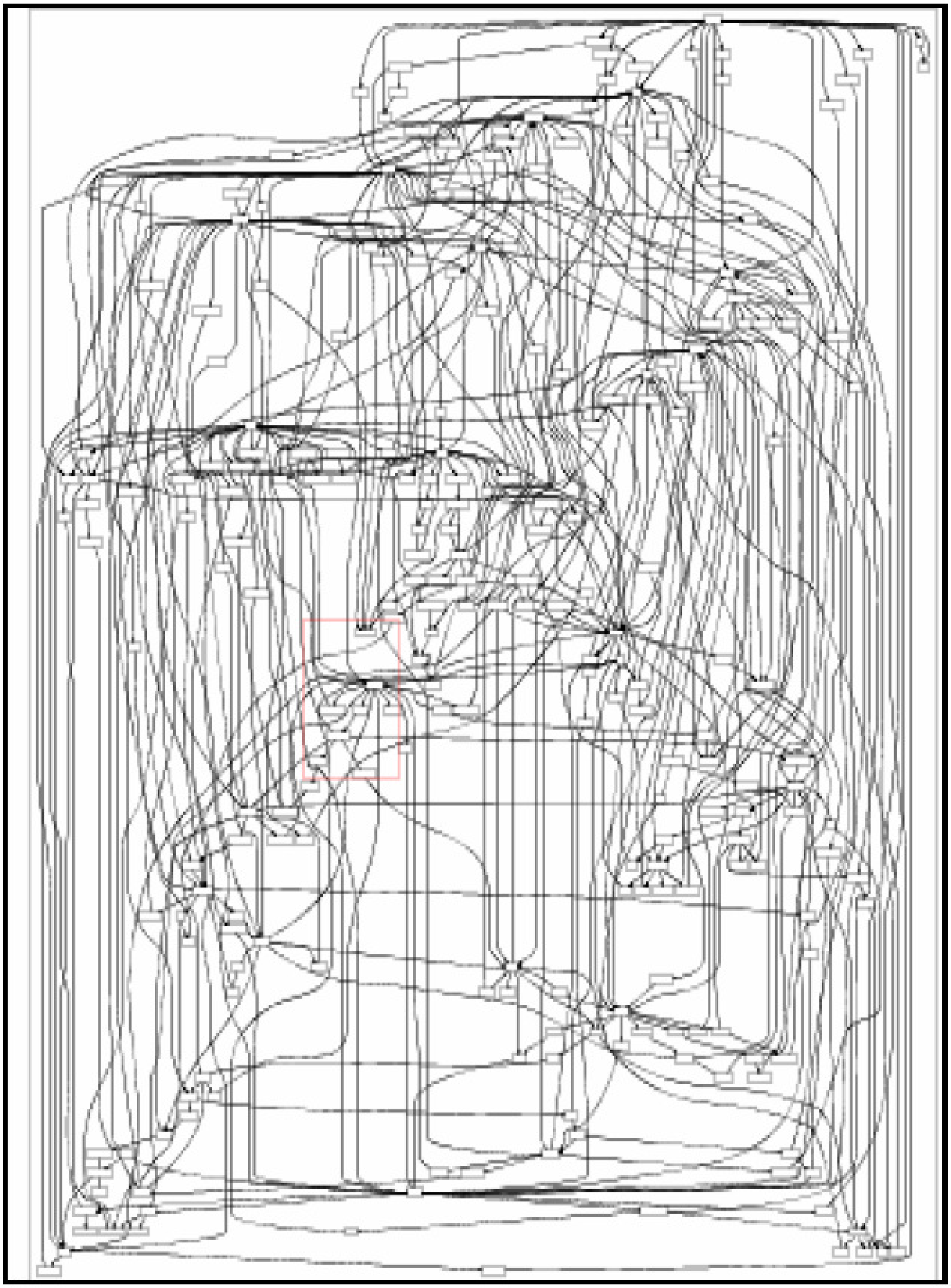
[92] Achieving a General, Formal and Decidable Approach to the OR-Join in Workflow Using Resets Net, M.T. Wynn, D. Edmond, W.M.P. van der Aalst and A.H.M. terHofstede, In G. Ciardo and P. Darondeau (Eds.): ICATPN 2005, LNCS 3536, pages 423-443, 2005, Springer-Verlag Berlin Heidelberg, 2005.

[93] Verifying Workflows with Cancellation Regions and OR-join: An Approach Based on Invariants, H.M.W. Verbeek, W.M.P. van der Aalst and A.H.M. ter Hofstede.

[94] Pattern Based Analysis of BPEL4WS, P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, Technical Report FIT-TR-2002-04, QUT.

[95] Process Management, A Guide for the Design of Business Processes, Jörg Becker, Martin Kugeler, Michael Rosemann, Springer Verlag, March 2003.

Appendix



Contents:

A.1 List of Abbreviations

AMC	Academisch Medisch Centrum Amsterdam
ATE	Audit Trail Entry
BDI	Barthel Disability Index
BPEL	**B**usiness **P**rocess **E**xecution **L**anguage
BPM	Business Process Management
Corr	Correlation coefficient
CPN	Coloured Petri Net
DG	Dependency Graph
EPC	Event driven Process Chain
ERP	Enterprise Resource Planning
HM	Heuristics Miner
HN	Heuristics Net
I	Interest
IS	Information Systems
KPI	Key Performance Indicator
LM	Local Metric
L1L	Loop of length one
L2L	Loop of length two
MIS	Management Information System
ML	Modeling Language
MXML	**M**ining **Ex**tended **M**arkup **L**anguage
P	Probability
PI	Process Instance
PM	Process Mining
PML	Process Modeling Language
PN	Petri Net
ProM	**Pro**cess **M**ining Framework
Q	Question
rel.	relative
resp.	respectively
sup	support
SGML	**S**tandard **G**eneralized **M**arkup **L**anguage
SJI	SplitJoinIndicator
sup	support
TU/e	Technische Universiteit Eindhoven
vs	versus
XML	**Ex**tended **M**arkup **L**anguage
XOR	**Ex**clusive OR
YAWL	**Y**et **A**nother **W**orkflow **L**anguage
Weka	**W**aikato **E**nvironment for **K**nowledge **A**nalysis
WF	Workflow
WfMS	Workflow Management System

A.2 List of used Mathematical Symbols

$\Phi(a,b)$	*Correlation between the activities a and b*
$cov(a,b)$	*Covariance between the activities a and b*
$\cap$	*Intersection*
$\cup$	*Union*
$\in$	*Element*
$\wedge$	*And*
$\vee$	*Or*
X	*The mean of a large sample size*
$P(X=x)$	*Probability of event x*
$P(Y\|X)$	*Conditional probability of activity Y given activity X*
$P(A\rightarrow Y\|X=1)$	*Conditional probability of activity Y via A given activity X*
$[0,1)$	*Interval between zero(included) and one (excluded)*
$>$	*Greater*
$\geq$	*Greater or equal*
$<$	*Less*
$\leq$	*Less or equal*
$=$	*Equal*
$\neq$	*Unequal*
$\approx$	*Approximately*
$\|$	*Exception*
A	*Activity a*
B	*Activity b*
T	*Tasks*
W	*Event log*
$DG(W)$	*Dependency graph of event log W*
R	*Pearson product moment correlation coefficient*

A.3 List of used Definitions

Activity	*The performance of a single process by a person or machine.*
Acute stroke	*A vascular condition that precipitates neurological damage and is the second leading cause of death in industrialized countries.*
Apriori algorithm	*It is a popular technique, which clusters activities.*
Artificial data	*Data which is characterized by the generic term of the activities.*
Audit Trail Entry	*An event type in the logfile.*
Barthel Disability Index	*Scores then separate activities of dialy living, including continue and mobility, with regard to their effect on an patient independence.*
Case identifiers	*Case ids are unique in the log file and mostly in relation with persons. Same as process ids.*
Causal Dependency Matrix	*A matrix, which contain the results of the dependency values between all activities.*
Confidence Interval	*The variation of the probability.*
CPN Tools	*It is a process modeling language for design, specification, simulation, validation and implementation of large software systems.*
Data layout	*A two binary dimensional matrix. It contains only zero or one in the cells. The columns are the related activities and the rows are the case ids.*
Dependency Graph	*A general process model which illustrates the relation between activities based on the frequency based metric.*
Event log	*A logfile which is recorded by an information system and contains the events during a certain period.*
Fork	*The connection between two or more activities in a process model (including itself if possible).*
Frequency	*Number of occurrence of activities in the data layout.*
Heuristics Miner	*A process mining algorithm for discovering relation between activities on a simple heuristic way.*
Heuristics Net	*A process model which includes also the type of splits / join (XOR, AND).*
Java	*Original release redefined programming language for the internet.*
Join	*Activity, which can only be enabled by its (one or more) directly connected activities.*
Less Structured Processes	*The processes are characterized by less ordering of performance of tasks.*
Network	*The activities in a business environment.*
Noise	*Information, which is not recorded correct in the logfile.*
Process	*Consist of a number of logical units of work.*
Processes Indentifiers	*These are unique in the log file and mostly in relation with persons.*
Process Mining	*A research study which enables the extraction of information from event logs recorded by an information*

	system.
Split	*Activity, which enables its direct connected activities.*
Structured Processes	*The processes will be performed in a certain order.*
Task	*Logical units of work.*
Threshold	*A minimum value for parameters, which the result must satify.*
Workflow	*Network of activities which has rules. These rules determine the partial order in which the tasks should be performed.*
Workflow Management System	*A generic software tool which allows for definition, execution, registration, and control of workflow.*
YAWL	*A Workflow tool, which is based on a rigorous analysis of existing workflow management system amd related standards using a comprehensive set of workflow patterns.*

A.4 Functional Design GUI of the new Knowledge Discovery Tool

The functional design of the new discovering tool is based on the four methods, which are described in chapter 3. The functional design describes the graphical part of this new tool. The design exists of two parts, e.g. the visualization of the statistical data in the DG and the representation of the statistical data by tables. The statistical information is about the splits and joins in the process model.

A.4.1 The Options of the new Knowledge Discovery Tool

The Heuristics Miner (HM) shows after the start up the DG. Currently, there are two options for illustrating the DG with and without semantics, which is realized by a checkbox option. The split/join information is a third option for representing the graph. Therefore, a radio button technique is necessary for the new visualization technique (instead of the check box). This technique makes possible to select only type of visualization of the DG. The radio button option will get three options, i.e. the DG, DG including semantics and DG including split/join information. The last one is new and shows the DG including the probabilities of the branches directly after the split and just before the join. As default option will be the DG.

There are two options for the visualization of the characteristics of the forks, e.g. the DG and the tables. Next section is about the representation by DG.

A.4.2 Dependency Graph including Fork Information

First of all, the rectangles in the DG will be expanded with the probabilities between the split/join and the directly related activities. This information gives directly insight in the type split/join. The number of parsing in the DG will be deleted and instead at this place (above the arrows) will be given the probabilities of the branches. The number of parsing is based on the direct succession between two activities and is part of the DG metric. This metric determinates the relation between activities and gives no sensible information of the forks.

The probabilities will be placed between the forks and the directly connected activities in the process model. They are derivate from the statistical table of the individual activities and based on both sides of the connection. These are rounded in three decimals. The illustrating of the probabilities of both side (i.e. *p(s)* and *p(j)*) in the DG gives already the first insight in the behaviour of the forks (structured or less structured). There are two type of probabilities noted in the process model.

The first probability, called "*p(s)*", is based on the split side. It is the number of occurring of the individual activity divide by the number of occurring of the split. For instance, *X1* (split) which is 3,024 times performed, enables 1,120 times task *D*. The probability (= *p(s)*) is 1,120/3,024 = 0.37, which is the same as the probability of the individual task in table 1.

The second probability, called "*p(j)*", is based on the join side. It is the number of occurring of the individual activities divide by the number of occurring of the join. For instance, task *B* starts up (1,048 times) after the finishing of task *X1* (3,024 times). The probability (= *p(j)*) is 1,048/3,024 = 0.347. Another example, task *D* is performed 1,120 times and was 1,120 times enabled by *X1* (split). The probability (= *p(j)*) is 1,120/1,120 = 1. In other words, this probability (i.e. *p(j)*) looks from the opposite direction.

Some remarks about the probabilities in the DG. The probabilities vary between 0 and 1. The probability will be reduced to 1, if the number of individual is higher than the number of occurring of the join. This information will also placed in the loops of length one. It is possible that both probabilities are different in a loop of length one. As split, the loop can only started before enabling one of the direct connected activities. As join, the loop can be enabled after finishing one or more of the direct connected activities to the fork. The number of repeating can be different. This depends on the output of the traces, which fills the "baskets" rows in the binary two dimensional matrix. The way of filling is different in both situations, which can lead to different rows in this matrix (see section 3.2). For instance, activity *A* as join is performed 5,772 times and is connected by itself and *X1*. The probability of the loop and *X1* are resp. 3,528/5,772 = 0.611 and 2,244/5,772 =0.389. Activity *A* is also a split, which enables itself (3,604 times) and activity *Y1* (2,168 times). The probabilities are resp. 3,604/5,772 = 0.624 and 2,168/5,772 = 0.376. Figure 28 shows the probabilities of the direct connected activities to the split (activity *X1*). This split include a loop of length one. This DG is based on the event log, which is described in section 3.1.

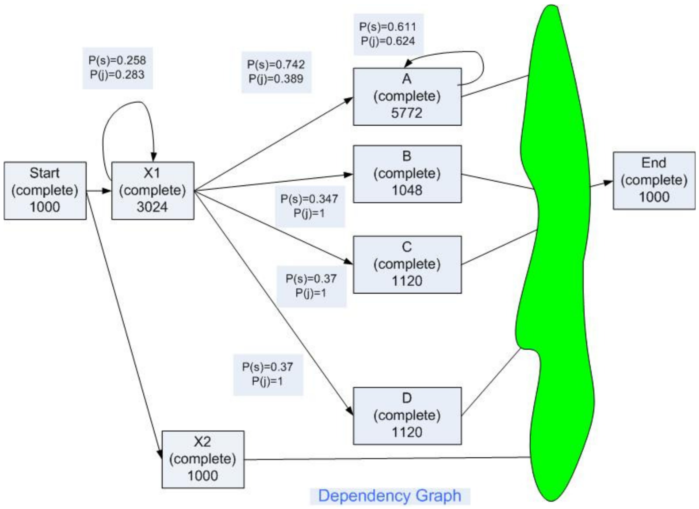


Figure 28: The DG including the probabilities of the individual activities..

A.4.3 Active statistical Information between Activities in a Dependency Graph

Another option is the activation of two rectangles by clicking. After this, the statistical information between the activated activities will be shown in a block form. Figure 29 shows two marked rectangles *B* and *C* and their statistical measurements. It shows the first three interestingness measures between both activities from the third table *"Statistical table for related activities"*.

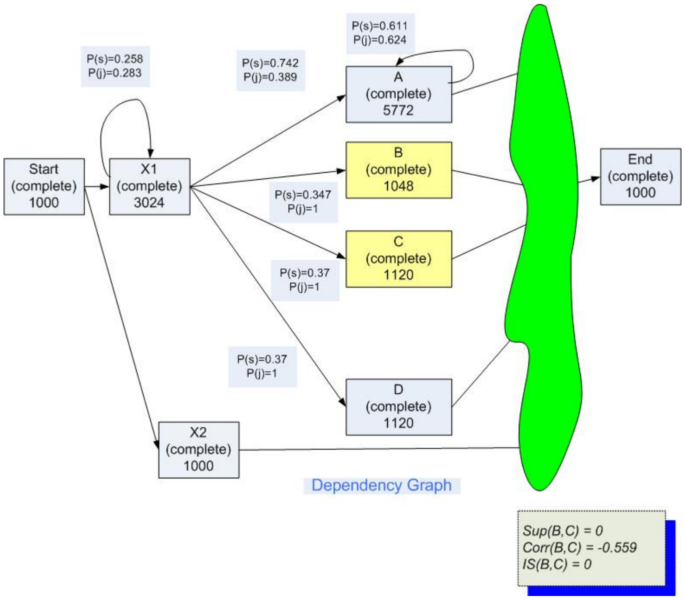


Figure 29: Two activated activities in a DG and their properties.

A.4.4 Active statistical Information between Branches in a Dependency Graph

The probability of the combination of activities gives insight which branches are mostly active between the corresponded split/join. Figure 30 illustrates the combinations of at least three activities, which were enabled by corresponded split (activity *X1*). The probabilities of these combinations are descending ordered. The probability is the frequency of the combined activities dividend by the frequency of the direct related split/join. This metric value is derivate from the second statistical table, called *"Combination table"*. The probabilities of one and two related activities are already presence in the DG.

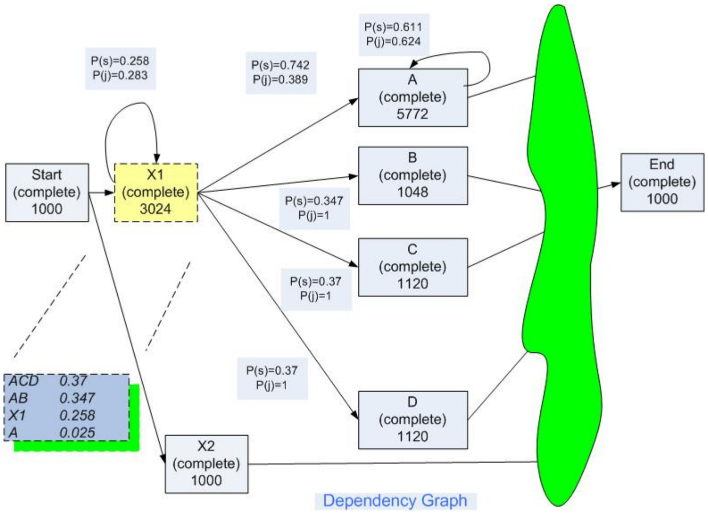


Figure 30: The percentage of active combined branches in a DG.

A.4.5 The Tables of the four Methods

The representation of the four methods is the table form. Each table represents one method. Every table provides information about the split/join and to its direct connected activities. The first table, called *"Statistical Measures for Activities"* illustrates the frequency and probabilities of the individual activities to its split/join. Table 11 shows the metric values of the individual activities to split *X1*. The probability in this table is the frequency of the individual task divide by the frequency of its fork. For instance, the probability of the individual task *A* is equal to 2,244/3,024 = 0.742 rounded in three decimals.

Split/Join	*Type*	*Frequency*	*Activities*	*Frequency*	*Probability*
…	…	…	…	…	…
X1	*Split*	*3024*	*X1*	*780*	*0.258*
			A	*2244*	*0.742*
			B	*1048*	*0.347*
			C	*1120*	*0.37*
			D	*1120*	*0.37*
…	…	…	…	…	…

Table 11: The statistical table of the individual activities.

The second table, called *"Combination table"*, gives insight in the combination of activities related to the split/join. It shows all combinations and the probability always sum up 1. It is possible that rows in a matrix only contain zeroes. This means that a split/join has none activities enabled after its finishing. In this case, the table gets a row, called *"None activities"* and its corresponding metric values. The calculation methode of the probability is the same as in the previous table.Table 12 shows the first part of the combination table.

Split/Join	*Type*	*Frequency*	*Combination of activities*	*Frequency*	*Probability*
…	…	…	…	…	…
X1	*Split*	*3024*	*A C D*	*1120*	*0.37*
			A B	*1048*	*0.347*
			X1	*780*	*0.258*
			A	*76*	*0.025*
…	…	…	…	…	…

Table 12: The combination table.

The third table, called *"Statistical Measures for Related Activities"*, shows the degree of dependency between two activities. The metrics support, correlation coefficient and IS are part of this table. The support is the percentage of both activities in the binary matrix (i.e. probability). It is based on the rows where both activities are presence in the binary matrix. For instance, the support of *AB* is the frequency of AB (i.e. 1,048) divide by the frequency of its fork (i.e. 3,024), which is 1,048/3,024 = 0.347 rounded in three decimals. Table 13 illustrates the dependency degree between the activities to split *X1*.

Split/Join	*Type*	*Frequency*	*Related activities*	*Frequency*	*Support*	*Correlation*	*IS*
…	…	…	…		…	…	…
X1	*Split*	*3024*	*X1A*	*0*	*0*	*-1*	*0*
			X1B	*0*	*0*	*-0.429*	*0*
			X1C	*0*	*0*	*-0.452*	*0*
			X1D	*0*	*0*	*-0.452*	*0*
			AB	*1048*	*0.347*	*0.429*	*0.683*
			AC	*1120*	*0.37*	*0.452*	*0.706*
			AD	*1120*	*0.37*	*0.452*	*0.706*
			BC	*0*	*0*	*-0.559*	*0*
			BD	*0*	*0*	*-0.559*	*0*
			CD	*1120*	*0.37*	*1*	*1*
…	…	…	…		…	…	…

Table 13: The statistical table between two direct related activities.

The fourth table, called *"Apriori table"*, illustrates the relation between activities in a group, which is a result of the *Apriori* algorithm. Also part of this table the frequency between the related activities and their confidence. This table shows only relation between activities that satisfy on the minimum support and minimum confidence (e.g. 0.10 respectively 0.90). The number of rules is default 10. Table 14 shows the first five rules between the activities to split *X1*, which are generated by the *Apriori* algorithm. For instance, the first rule to split *X1* is the presency of task *A* leads to a non-presency of task *X1*. The confidence interval is 2,244/2,244 = 1.

Split/Join	*Type*	*Frequency*	*Activities*	*Frequency*	*Activities*	*Frequency*	*Confidence*
…	…	…	…	…	…	…	…
X1	*Split*	*3024*	*A=1*	*2244*	*X1=0*	*2244*	*1.000*
			X1=0	*2244*	*A=1*	*2244*	*1.000*
			D=0	*1904*	*C=0*	*1904*	*1.000*
			C=0	*1904*	*D=0*	*1904*	*1.000*
			A=1 and B=0	*1196*	*X1=0*	*1196*	*1.000*
…	…	…	…	…	…	…	…

Table 14: Table of relations between activities in a group

A.4.6 General Table for the Characteristics of the Forks

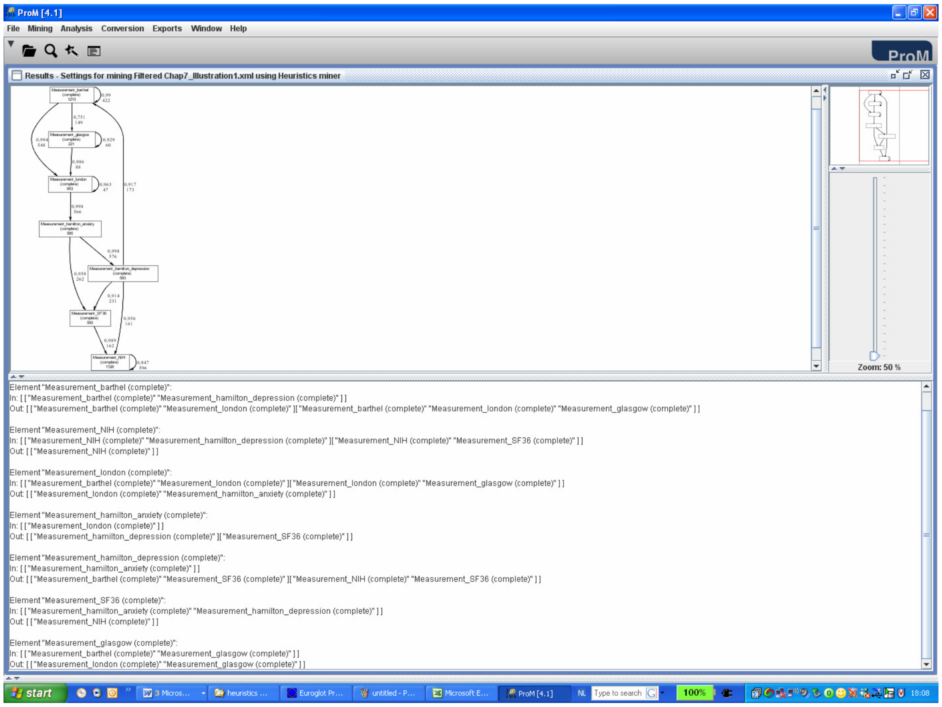
The above described four tables are based on the four methods, which characterize the forks in the process model. A general table based on the results of the four methods, gives in a short overview in the characteristics of every fork in the model. This table might be developed after the validation phase of the new tool. The advantage of this action is that it takes the experience of analysis into the setup of this fifth table. The information in this table is of a higher aggregation level. This new table stores information like total number of activities, summation of frequency of the individual tasks, total number of patterns frequency, the highest frequency of the patterns combination, the number of interesting correlations (+1, 0, and -1), number of interesting IS (0,1), and relation between three activities (if possible), etc. The interesting correlation and IS values corresponds with the number in the cell. For instance, the number of interesting correlation has the values *0,1,2*. This indicates *none* activities have a strong degree of dependency, (*1* combination of) two related activities are independent, and *2* combinations of two activities have a strong negative dependency. These are the total numbers of a process model. Table 15 shows the (possible) setup of this general table. It gives general information about the split *X1*.

Split/Join	***Type***	***Freq.***	***Description***	***value***
…	…	…	...	...
X1	*Split*	*3024*	*Number of activities*	*5*
			Summation of frequency	*4024*
			Highest frequency of patterns combination	*ACD,1120*
			Number of patterns combination	*5*
			Number of interesting correlation (1,0, -1)	*1,0,1*
			Number of interesting IS (0,1)	*6,1*
			Most interesting relation in a group	*X1=0 and D=0* ➔ *A=1*
…	…	…	…	…

Table 15: General table of the characteristics of the forks.

A.5 Subsets of Case Study 1 according the Heuristics Miner Algorithm

The HM algorithm shows the dependency graph and the subsets in the description part of the screen. Below the first figure illustrates the screen shot of the dependency and the description of the subsets. Figure 32 shows the enlargement of the description of the subsets.


ProM [4.1]
File Mining Analysis Conversion Exports Window Help
ProM
Results - Settings for mining Filtered Chap7_Illustration1.xml using Heuristics miner
Zoom: 50 %
start

Element "Measurement_barthel (complete)":
In: [["Measurement_barthel (complete)" "Measurement_hamilton_depression (complete)"]]
Out: [["Measurement_barthel (complete)" "Measurement_london (complete)"]["Measurement_barthel (complete)" "Measurement_london (complete)" "Measurement_glasgow (complete)"]]

Element "Measurement_NIH (complete)":
In: [["Measurement_NIH (complete)" "Measurement_hamilton_depression (complete)"]["Measurement_NIH (complete)" "Measurement_SF36 (complete)"]]
Out: [["Measurement_NIH (complete)"]]

Element "Measurement_london (complete)":
In: [["Measurement_barthel (complete)" "Measurement_london (complete)"]["Measurement_london (complete)" "Measurement_glasgow (complete)"]]
Out: [["Measurement_london (complete)" "Measurement_hamilton_anxiety (complete)"]]

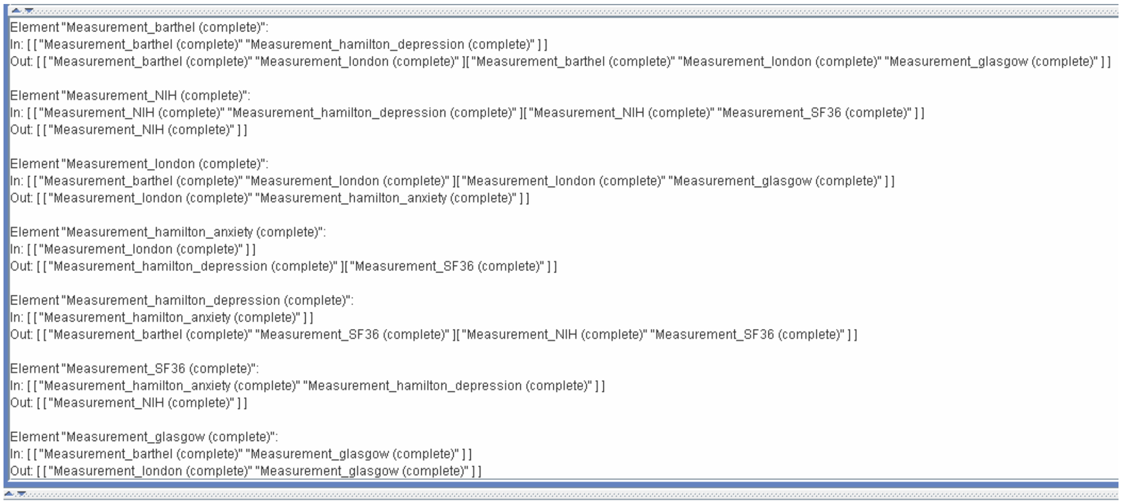
Element "Measurement_hamilton_anxiety (complete)":
In: [["Measurement_london (complete)"]]
Out: [["Measurement_hamilton_depression (complete)"]["Measurement_SF36 (complete)"]]

Element "Measurement_hamilton_depression (complete)":
In: [["Measurement_hamilton_anxiety (complete)"]]
Out: [["Measurement_barthel (complete)" "Measurement_SF36 (complete)"]["Measurement_NIH (complete)" "Measurement_SF36 (complete)"]]

Element "Measurement_SF36 (complete)":
In: [["Measurement_hamilton_anxiety (complete)" "Measurement_hamilton_depression (complete)"]]
Out: [["Measurement_NIH (complete)"]]

Element "Measurement_glasgow (complete)":
In: [["Measurement_barthel (complete)" "Measurement_glasgow (complete)"]]
Out: [["Measurement_london (complete)" "Measurement_glasgow (complete)"]]

Figure 31: The DG and the description of the subsets of the measurement log.



Element "Measurement_barthel (complete)":
In: [["Measurement_barthel (complete)" "Measurement_hamilton_depression (complete)"]]
Out: [["Measurement_barthel (complete)" "Measurement_london (complete)"]["Measurement_barthel (complete)" "Measurement_london (complete)" "Measurement_glasgow (complete)"]]

Element "Measurement_NIH (complete)":
In: [["Measurement_NIH (complete)" "Measurement_hamilton_depression (complete)"]["Measurement_NIH (complete)" "Measurement_SF36 (complete)"]]
Out: [["Measurement_NIH (complete)"]]

Element "Measurement_london (complete)":
In: [["Measurement_barthel (complete)" "Measurement_london (complete)"]["Measurement_london (complete)" "Measurement_glasgow (complete)"]]
Out: [["Measurement_london (complete)" "Measurement_hamilton_anxiety (complete)"]]

Element "Measurement_hamilton_anxiety (complete)":
In: [["Measurement_london (complete)"]]
Out: [["Measurement_hamilton_depression (complete)"]["Measurement_SF36 (complete)"]]

Element "Measurement_hamilton_depression (complete)":
In: [["Measurement_hamilton_anxiety (complete)"]]
Out: [["Measurement_barthel (complete)" "Measurement_SF36 (complete)"]["Measurement_NIH (complete)" "Measurement_SF36 (complete)"]]

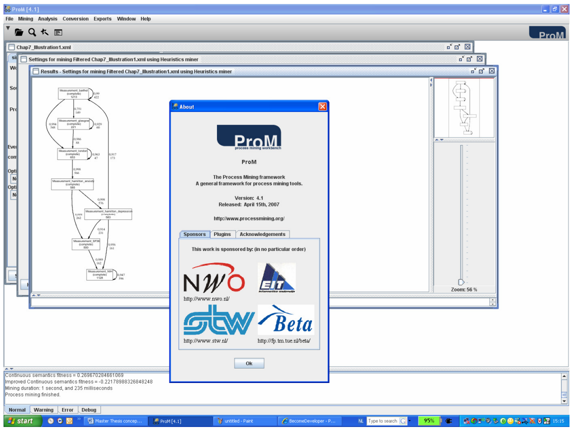
Element "Measurement_SF36 (complete)":
In: [["Measurement_hamilton_anxiety (complete)" "Measurement_hamilton_depression (complete)"]]
Out: [["Measurement_NIH (complete)"]]

Element "Measurement_glasgow (complete)":
In: [["Measurement_barthel (complete)" "Measurement_glasgow (complete)"]]
Out: [["Measurement_london (complete)" "Measurement_glasgow (complete)"]]

Figure 32: The enlargement of the subset description

A.6 ProM Framework

ProM Framework is the process mining tool, which is developed at the Eindhoven University of Technology. The predecessor of this tool is the mining tools, called Little Thumb. 15th April 2007 was released version (4.1) of *ProM* (see Figure 33) and EMiT [25]. The latest version of *ProM* is available at the website http://www.processmining.org.


ProM [4.1]
File Mining Analysis Conversion Exports Window Help
ProM
Chap7_Illustration1.xml
Settings for mining Filtered Chap7_Illustration1.xml using Heuristics miner
Results - Settings for mining Filtered Chap7_Illustration1.xml using Heuristics miner
About
Zoom: 56 %
Sponsors
Plugins
Acknowledgements
Ok
Normal
Warning
Error
Debug
start
Master Thesis concep...
ProM [4.1]
untitled - Paint
BecomeDeveloper - P...
Type to search
95%
15:15

ProM
process mining workbench

ProM

The Process Mining framework
A general framework for process mining tools.

Version: 4.1
Released: April 15th, 2007

http://www.processmining.org/

This work is sponsored by: (in no particular order)

http://www.nwo.nl/
http://www.stw.nl/
http://fp.tm.tue.nl/beta/

```
Continuous semantics fitness = 0.269670284661069
Improved Continuous semantics fitness = -0.22178988326848248
Mining duration: 1 second, and 235 milliseconds
Process mining finished.
```

Figure 33: Screenshot of the ProM tool version 4.1.

The original purpose for the *ProM* framework was to serve as a platform for process mining. As development ensued, the scope of the framework grew broader to encompass tasks ranging from process verification to social network analysis to conformance checking, and more.

Additionally, the *ProM* framework supports a wide variety of process models and operations. The *ProM* is used to mine several processes in practice, like processes in a hospital, processes of a parish, etc. Figure 34 provides an overview of PM and the various relations between entities such as IS, operational process, event logs, and process models. It also shows the environment of *ProM* framework related to PM.

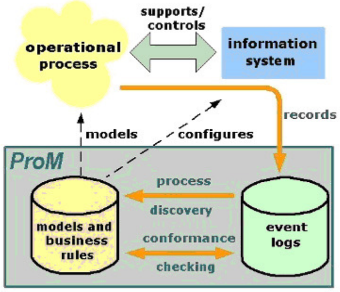


Figure 34: ProM framework and his environment related to PM.

The *ProM* framework can store a log in, which has a generic *XML* format. Next section in this appendix describes in more detail about the process log format. Another important feature of this framework is that it allows for interaction between a large numbers of plug-ins. A plug-in is basically the implementation of an algorithm that is of some use in the PM area, where the implementation agrees with the framework. Currently, there are 35 mining plug-ins, 38 analysis plug-ins, 29 Export plug-ins, 17 Import plug-ins, 24 conversion plug-ins, and 14 log filters. The list of plug-ins and log filters is available at the website http://is.tm.tue.nl/trac/prom/wiki/ProMPlugins.

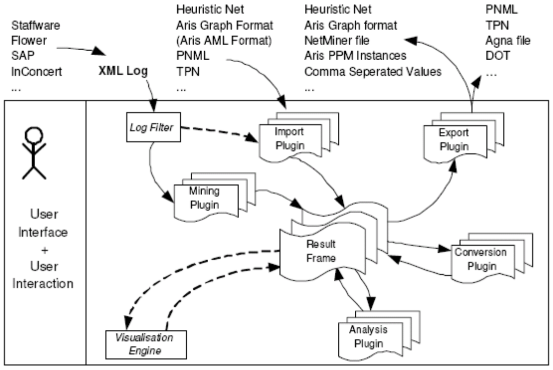
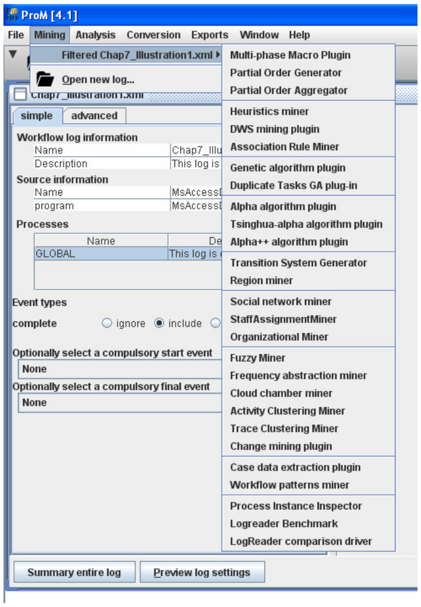


Figure 35: Overview of the ProM framework.

Figure 35 shows an overview of the framework. It explains the relations between the framework, the process log format, and the plug-ins. As Figure 35 shows, the *ProM* framework can read in the XML-format through the Log filter component. This component is able to read with large data sets and sorts the events within a case on their timestamps before the actual mining starts. Through the Import plug-ins a wide variety of models can be loaded ranging from a PN to logical formulas. The Mining plug-ins do the actual mining and the result is stored in memory, and in a window on the *ProM* desktop. The mining results contain some kind of visualization, e.g. displaying a PN [16], or a social network [18] or further analysis or conversion. The framework allows plug-ins to operate on each others results in a standardized way. The analysis plug-ins take a mining result and analyze it, e.g. calculating a place invariant for a resulting PN. The conversion plug-ins take a mining result and transform it into another format, e.g. transforming EPC (Event-Driven Process Chain) [19] into a PN. Figure 36 shows a screenshot of the mining tools of *ProM* version 4.1.



Workflow log information
Name Chap7_Illu
Description This log is

Source information
Name MsAccessI
program MsAccessI

Processes

Name	De
GLOBAL	This log is

Event types
complete ○ ignore ● include ○

Optionally select a compulsory start event
None
Optionally select a compulsory final event
None

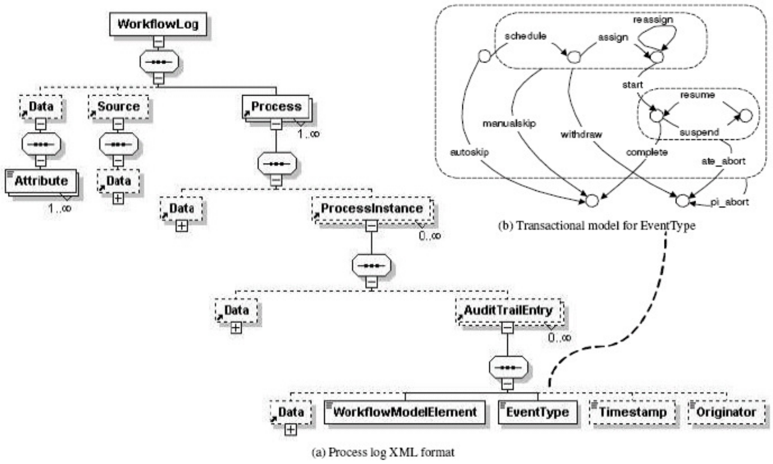
Figure 36: Screenshot of mining tools.

A.7 The Format of the event Log

This section describes the process log format, which can be stored in the ProM framework. The process log format is an XML schema, which is developed by the researchers of the University of Technology at Eindhoven. Figure 37 shows the XML schema that specifies the process log format (pages 145-146 of [1]). The root element is the *WorkflowLog*. The *WorkflowLog* elements contain in the given order the following optional elements:

- *Data*;
- *Source*;
- a number of *Process*.

A *Data* element allows for storing arbitrary textual data. It contains a list of *Attribute* elements. A *Source* element might be used to store information about the IS this log originated from. A *Process* element refers to a specific process in an information system. Since most IS typically control several processes, multiple *Process* elements may exist in a log file. Every process (element *Process*) has zero or more cases or process instances (element *ProcessInstance*). Similary, every process instance has zero or more tasks (element *Audit Trail Entry* (ATE)). Every task or *ATE* should at least have a name (element *WorkflowModelElement*) and an event type (element *EventType*). The event type determines the state of the tasks. An *ATE* may also include a timestamp (element *Timestamp*), and a originator (element *Orginator*. The *Timestamp* Element supports the logging of time for the task. The *Originator* element records the person respectively the system that performed the task.



(b) Transactional model for EventType

(a) Process log XML format

Figure 37: Process log XML format (a) and transactional model (b).

In order to be able to talk about these events in a standard way, department IS developed a transactional model. It shows the events in a log. Again this model is based on analyzing the different type of logs in real-life systems (e.g. Staffware, SAP, FLOWer, etc.). Figure 37 (b) shows the transactional model.

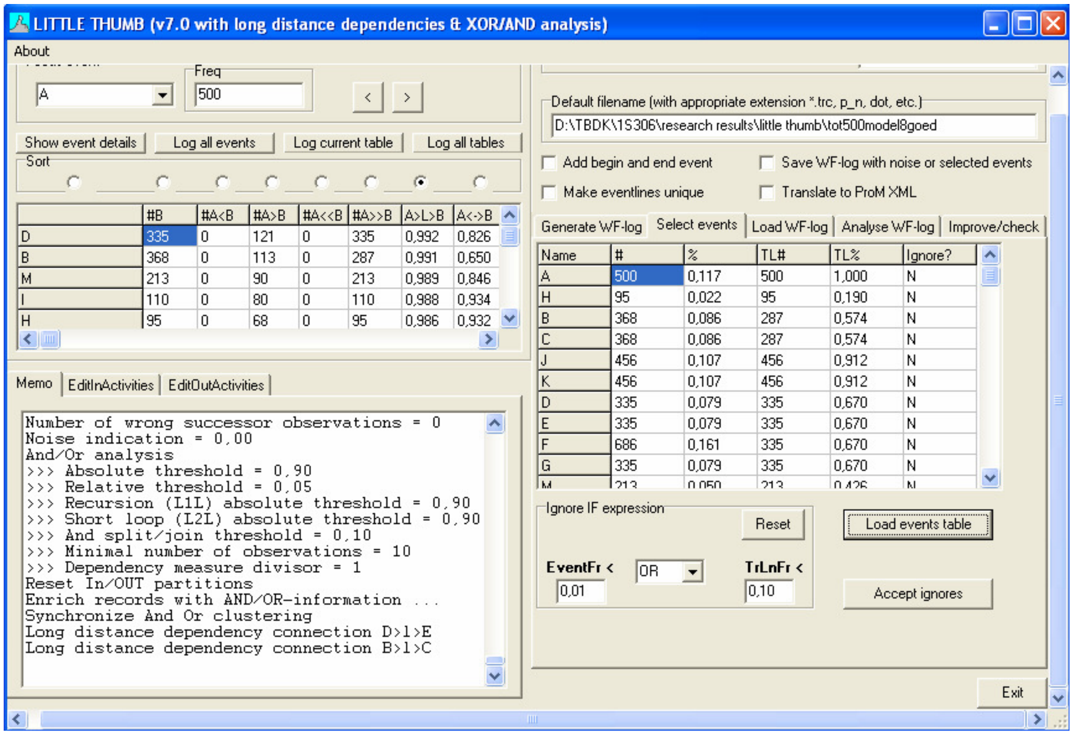
There are 13 supported event types: *schedule*, *assign*, *reassign*, *start*, *resume*, *suspend*, *autoskip*, *manualskip*, *withdraw*, *complete*, *ate_abort*, *pi_abort* and *unknown*. Figure 38 illustrates an excerpt of a log in the Mining XML format (MXML). It is an log of the first consult *(eerste consult)* at the academic hospital at Amsterdam, called “AMC”. The schema for the MXML format is available at http://www.processmining.org/-WorkflowLog.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="WorkflowLog.xsd" description="Exported
by ProM framework from Log file converted from AMC *.xls / parsed from
CSV">
<Data>
<Attribute name="java.version">1.5.0_06</Attribute>
<Attribute name="os.name">Windows XP</Attribute>
<Attribute name="os.arch">x86</Attribute>
<Attribute name="app.user">cgunther</Attribute>
<Attribute name="java.vendor">Sun Microsystems Inc.</Attribute>
<Attribute name="app.codename">Detroit</Attribute>
<Attribute name="app.name">ProM Import Framework</Attribute>
<Attribute name="os.version">5.1</Attribute>
<Attribute name="app.version">1.6.0</Attribute>
</Data>
<Source program="unknown (MS Excel)">
<Data>
<Attribute name="program">unknown (MS Excel)</Attribute>
</Data>
</Source>
<Process id="A" description="process of type 'A'">
<ProcessInstance id="patient_1" description="Treatment of patient
number 1">
<AuditTrailEntry>
<Data>
<Attribute name="ColumnNumber">column_2</Attribute>
<Attribute name="TaskClass">general</Attribute>
</Data>
<WorkflowModelElement>eerste consult AMC</WorkflowModelElement>
<EventType>complete</EventType>
<Timestamp>2003-07-21T01:00:00.000+02:00</Timestamp>
<Originator>unknown</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
<Data>
<Attribute name="ColumnNumber">column_11</Attribute>
<Attribute name="TaskClass">diagnosis</Attribute>
</Data>
<WorkflowModelElement>thorax</WorkflowModelElement>
<EventType>complete</EventType>
<Timestamp>2003-07-21T01:02:00.000+02:00</Timestamp>
<Originator>unknown</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
<Data>
. . . . .
```

Figure 38: The XML format of the first consult at "AMC".

A.8 The Results of a running Example in Little Thumb

This section describes the results of case study by using the PM tool, called *Little Thumb*. This PM tool is the predecessor of *ProM*. Figure 39 illustrates the Little Thumb tool. First of all, the logfile needs to be transformed a little bit. The activities of every trace (row) in de textfile should separated by ”,” followed by changing of the extension of the file into “.trc”.



A

Freq: 500

	#B	#A<B	#A>B	#A<<B	#A>>B	A>L>B	A<->B
D	335	0	121	0	335	0,992	0,826
B	368	0	113	0	287	0,991	0,650
M	213	0	90	0	213	0,989	0,846
I	110	0	80	0	110	0,988	0,934
H	95	0	68	0	95	0,986	0,932

```
Number of wrong successor observations = 0
Noise indication = 0,00
And/Or analysis
>>> Absolute threshold = 0,90
>>> Relative threshold = 0,05
>>> Recursion (L1L) absolute threshold = 0,90
>>> Short loop (L2L) absolute threshold = 0,90
>>> And split/join threshold = 0,10
>>> Minimal number of observations = 10
>>> Dependency measure divisor = 1
Reset In/OUT partitions
Enrich records with AND/OR-information ...
Synchronize And Or clustering
Long distance dependency connection D>l>E
Long distance dependency connection B>l>C
```

Default filename (with appropriate extension *.trc, p_n, dot, etc.): D:\TBDK\1S306\research results\little thumb\tot500model8goed

□ Add begin and end event □ Save WF-log with noise or selected events
□ Make eventlines unique □ Translate to ProM XML

Name	#	%	TL#	TL%	Ignore?
A	500	0,117	500	1,000	N
H	95	0,022	95	0,190	N
B	368	0,086	287	0,574	N
C	368	0,086	287	0,574	N
J	456	0,107	456	0,912	N
K	456	0,107	456	0,912	N
D	335	0,079	335	0,670	N
E	335	0,079	335	0,670	N
F	686	0,161	335	0,670	N
G	335	0,079	335	0,670	N
M	213	0,050	213	0,426	N

Ignore IF expression
EventFr < 0,01 OR TrLnFr < 0,10

Figure 39: Little Thumb screen shot with the results of example in section A.8.

The Little Thumbs results of example in section A.8 are:

```
D:\TBDK\1S306\research results\little thumb\tot500model8.trc open!
Make an event name list for D:\TBDK\1S306\research results\little thumb\tot500model8.trc.
Total reset all event record information!
12 events names!

Make the basic tables (#A<B, ..., #A-->B) and load D:\TBDK\1S306\research results\little
thumb\tot500model8.trc
Reset Basic tables
Reset all event records information except NAMING and IGNORING!
Event-log properties
500 event lines!
4297 event tokens!
66 different A,B-patterns (completeness indication)
D:\TBDK\1S306\research results\little thumb\tot500model8.trc open!
Make an event name list for D:\TBDK\1S306\research results\little thumb\tot500model8.trc.
Total reset all event record information!
12 events names!
NAME = A
```

```
Frequency = 500
Parse Information:
=================
NAME = A
Frequency = 500
Parse Information:
=================
NAME = H
Frequency = 145
NAME = B
Frequency = 355
Parse Information:
=================
NAME = C
Frequency = 355
Parse Information:
=================
NAME = I
Frequency = 162
Parse Information:
=================
NAME = J
Frequency = 459
Parse Information:
=================
NAME = K
Frequency = 459
Parse Information:
=================
NAME = M
Frequency = 213
Parse Information:
=================
NAME = D
Frequency = 335
Parse Information:
=================
NAME = E
Frequency = 335
Parse Information:
=================
NAME = F
Frequency = 644
Parse Information:
=================
NAME = G
Frequency = 335
Parse Information:
=================
A Freq: 500
      #B   A<B   A>B A<<<B A>>>B  A-->B   #A
D      335    0  130    0  335  0.992  0.828  0.000  0.828    0    0    0   500
B      355    0  106    0  287  0.991  0.668  0.000  0.668    0    0    0   500
M      213    0   97    0  213  0.990  0.858  0.000  0.858    0    0    0   500
I     162    0   73    0  162  0.986  0.723  0.000  0.723    0    0    0   500
H      145    0   66    0   98  0.985  0.624  0.000  0.624    0    0    0   500
```

```
A     500   0    0    0    0  0.000  0.000  0.000  0.000    0    0    0   500
C     355   0    0    0  287  0.000  0.413  0.000  0.413    0    0    0   500
J    459   0    0    0  459  0.000  0.304  0.000  0.304    0    0    0   500
K     459   0    0    0  459  0.000  0.243  0.000  0.243    0    0    0   500
E     335   0    0    0  335  0.000  0.523  0.000  0.523    0    0    0   500
F    644   0    0    0  335  0.000  0.329  0.000  0.329    0    0    0   500
G     335   0    0    0  335  0.000  0.286  0.000  0.286    0    0    0   500

A Freq: 500
        #B    A<B   A>B A<<<B A>>>B  A-->B    #A
D     335   0  130    0  335  0.992  0.828  0.000  0.828    0    0    0   500
B     355   0  106    0  287  0.991  0.668  0.000  0.668    0    0    0   500
M      213   0   97    0  213  0.990  0.858  0.000  0.858    0    0    0   500
I    162   0   73    0  162  0.986  0.723  0.000  0.723    0    0    0   500
H     145   0   66    0   98  0.985  0.624  0.000  0.624    0    0    0   500
A     500   0    0    0    0  0.000  0.000  0.000  0.000    0    0    0   500
C     355   0    0    0  287  0.000  0.413  0.000  0.413    0    0    0   500
J    459   0    0    0  459  0.000  0.304  0.000  0.304    0    0    0   500
K     459   0    0    0  459  0.000  0.243  0.000  0.243    0    0    0   500
E     335   0    0    0  335  0.000  0.523  0.000  0.523    0    0    0   500
F    644   0    0    0  335  0.000  0.329  0.000  0.329    0    0    0   500
G     335   0    0    0  335  0.000  0.286  0.000  0.286    0    0    0   500

H Freq: 145
        #B    A<B   A>B A<<<B A>>>B  A-->B    #A
J    459   0   22    0   90  0.957  0.319  0.000  0.319    0    0    0   145
I    162   0   13    0   55  0.929  0.218  0.000  0.218    0    0    0   145
E     335   0    5   21   64  0.833  0.278  0.089  0.189    0    0    0   145
B     355  10   32   31   56  0.512  0.344  0.147  0.197    0    0    0   145
M      213  11   22   25   33  0.324  0.208  0.131  0.077    0    0    0   145
D     335  14   27   31   48  0.310  0.297  0.148  0.149    0    0    0   145
F    644   2    3   21   65  0.167  0.260  0.111  0.149    0    0    0   145
H     145  14   14   47   47  0.000  0.194  0.194  0.000    0    0    0   145
K     459   0    0    0   90  0.000  0.255  0.000  0.255    0    0    0   145
C     355   9    3   22   65 -0.462  0.287  0.113  0.174    0    0    0   145
G     335  19    1   19   64 -0.857  0.169  0.131  0.038    0    0    0   145
A     500  66    0   98    0 -0.985  0.000  0.624 -0.624    0    0    0   145

B Freq: 355
        #B    A<B   A>B A<<<B A>>>B  A-->B    #A
E     335   0   27   40  191  0.964  0.404  0.059  0.345    0    0    0   355
F    644   0   15   40  191  0.938  0.359  0.079  0.280    0    0    0   355
C     355  24  179   68  355  0.760  0.821  0.130  0.691    0    0    0   355
D     335  69   74  116  102  0.035  0.286  0.278  0.008    0    0    0   355
B     355   0    0   68   68  0.000  0.086  0.086  0.000    0    0    0   355
J    459   0    0    0  287  0.000  0.415  0.000  0.415    0    0    0   355
K     459   0    0    0  287  0.000  0.332  0.000  0.332    0    0    0   355
M      213  46   38   89   49 -0.094  0.219  0.346 -0.127    0    0    0   355
H     145  32   10   56   31 -0.512  0.147  0.344 -0.197    0    0    0   355
I    162  46   12   73   49 -0.576  0.179  0.400 -0.220    0    0    0   355
G     335  32    0   40  191 -0.970  0.212  0.112  0.100    0    0    0   355
A     500 106    0  287    0 -0.991  0.000  0.668 -0.668    0    0    0   355

C Freq: 355
        #B    A<B   A>B A<<<B A>>>B  A-->B    #A
J    459   0  125    0  287  0.992  0.607  0.000  0.607    0    0    0   355
```

```
I      162    2   13   82   38  0.688  0.166  0.329 -0.163    0    0    0  355
G      335    7   37   46  184  0.667  0.373  0.109  0.264    0    0    0  355
H      145    3    9   65   22  0.462  0.113  0.287 -0.174    0    0    0  355
E      335   39   66  120   96  0.255  0.269  0.277 -0.009    0    0    0  355
F      644   55   81  123  143  0.190  0.368  0.289  0.079    0    0    0  355
A      500    0    0  287    0  0.000  0.000  0.413 -0.413    0    0    0  355
C      355    0    0   68   68  0.000  0.104  0.104  0.000    0    0    0  355
K      459    0    0    0  287  0.000  0.486  0.000  0.486    0    0    0  355
B      355  179   24  355   68 -0.760  0.130  0.821 -0.691    0    0    0  355
M      213   27    0  123    0 -0.964  0.000  0.403 -0.403    0    0    0  355
D      335   43    0  191    0 -0.977  0.000  0.404 -0.404    0    0    0  355

I Freq: 162
        #B   A<B   A>B A<<<B A>>>B  A-->B    #A
J      459    0   42    0  153  0.977  0.498  0.000  0.498    0    0    0  162
E      335    0    3   32   79  0.750  0.292  0.115  0.177    0    0    0  162
B      355   12   46   49   73  0.576  0.400  0.179  0.220    0    0    0  162
D      335   10   36   45   66  0.553  0.367  0.143  0.224    0    0    0  162
M      213   15   22   41   31  0.184  0.179  0.164  0.015    0    0    0  162
F      644    3    4   32   82  0.125  0.292  0.140  0.152    0    0    0  162
I      162    0    0    0    0  0.000  0.000  0.000  0.000    0    0    0  162
K      459    0    0    0  153  0.000  0.399  0.000  0.399    0    0    0  162
C      355   13    2   38   82 -0.688  0.329  0.166  0.163    0    0    0  162
H      145   13    0   55    0 -0.929  0.000  0.218 -0.218    0    0    0  162
G      335   23    0   29   82 -0.958  0.173  0.167  0.006    0    0    0  162
A      500   73    0  162    0 -0.986  0.000  0.723 -0.723    0    0    0  162

J Freq: 459
        #B   A<B   A>B A<<<B A>>>B  A-->B    #A
K      459    0  459    0  459  0.998  1.000  0.000  1.000    0    0    0  459
A      500    0    0  459    0  0.000  0.000  0.304 -0.304    0    0    0  459
B      355    0    0  287    0  0.000  0.000  0.415 -0.415    0    0    0  459
J      459    0    0    0    0  0.000  0.000  0.000  0.000    0    0    0  459
D      335    0    0  334    0  0.000  0.000  0.322 -0.322    0    0    0  459
E      335    0    0  334    0  0.000  0.000  0.512 -0.512    0    0    0  459
F      644    0    0  334    0  0.000  0.000  0.505 -0.505    0    0    0  459
M      213   17    0  201    0 -0.944  0.000  0.373 -0.373    0    0    0  459
H      145   22    0   90    0 -0.957  0.000  0.319 -0.319    0    0    0  459
I      162   42    0  153    0 -0.977  0.000  0.498 -0.498    0    0    0  459
C      355  125    0  287    0 -0.992  0.000  0.607 -0.607    0    0    0  459
G      335  253    0  334    0 -0.996  0.000  0.920 -0.920    0    0    0  459

K Freq: 459
        #B   A<B   A>B A<<<B A>>>B  A-->B    #A
A      500    0    0  459    0  0.000  0.000  0.243 -0.243    0    0    0  459
H      145    0    0   90    0  0.000  0.000  0.255 -0.255    0    0    0  459
B      355    0    0  287    0  0.000  0.000  0.332 -0.332    0    0    0  459
C      355    0    0  287    0  0.000  0.000  0.486 -0.486    0    0    0  459
I      162    0    0  153    0  0.000  0.000  0.399 -0.399    0    0    0  459
K      459    0    0    0    0  0.000  0.000  0.000  0.000    0    0    0  459
M      213    0    0  201    0  0.000  0.000  0.298 -0.298    0    0    0  459
D      335    0    0  334    0  0.000  0.000  0.257 -0.257    0    0    0  459
E      335    0    0  334    0  0.000  0.000  0.409 -0.409    0    0    0  459
F      644    0    0  334    0  0.000  0.000  0.404 -0.404    0    0    0  459
G      335    0    0  334    0  0.000  0.000  0.736 -0.736    0    0    0  459
J      459  459    0  459    0 -0.998  0.000  1.000 -1.000    0    0    0  459
```

```
M Freq: 213
       #B    A<B    A>B A<<<B A>>>B  A-->B    #A
C      355    0   27     0   123  0.964  0.403  0.000  0.403    0     0     0   213
J     459    0    17     0   201  0.944  0.373  0.000  0.373    0     0     0   213
E      335    0    14     0   145  0.933  0.471  0.000  0.471    0     0     0   213
F     644    0    13     0   145  0.929  0.441  0.000  0.441    0     0     0   213
D      335   34    68    49    96  0.330  0.421  0.213  0.208    0     0     0   213
B      355   38    46    49    89  0.094  0.346  0.219  0.127    0     0     0   213
K      459    0     0     0   201  0.000  0.298  0.000  0.298    0     0     0   213
M       213    0     0     0     0  0.000  0.000  0.000  0.000    0     0     0   213
G      335    0     0     0   145  0.000  0.258  0.000  0.258    0     0     0   213
I     162    22    15    31    41 -0.184  0.164  0.179 -0.015    0     0     0   213
H      145    22    11    33    25 -0.324  0.131  0.208 -0.077    0     0     0   213
A      500   97     0   213     0 -0.990  0.000  0.858 -0.858    0     0     0   213

D Freq: 335
       #B    A<B    A>B A<<<B A>>>B  A-->B    #A
E      335    0   106     0   335  0.991  0.795  0.000  0.795    0     0     0   335
F     644    0    59     0   335  0.983  0.746  0.000  0.746    0     0     0   335
C      355    0    43     0   191  0.977  0.404  0.000  0.404    0     0     0   335
J     459    0     0     0   334  0.000  0.322  0.000  0.322    0     0     0   335
K      459    0     0     0   334  0.000  0.257  0.000  0.257    0     0     0   335
D      335    0     0     0     0  0.000  0.000  0.000  0.000    0     0     0   335
G      335    0     0     0   335  0.000  0.435  0.000  0.435    0     0     0   335
B      355   74    69   102   116 -0.035  0.278  0.286 -0.008    0     0     0   335
H      145   27    14    48    31 -0.310  0.148  0.297 -0.149    0     0     0   335
M       213   68    34    96    49 -0.330  0.213  0.421 -0.208    0     0     0   335
I     162    36    10    66    45 -0.553  0.143  0.367 -0.224    0     0     0   335
A      500  130     0   335     0 -0.992  0.000  0.828 -0.828    0     0     0   335

E Freq: 335
       #B    A<B    A>B A<<<B A>>>B  A-->B    #A
G      335    0    48     0   335  0.980  0.695  0.000  0.695    0     0     0   335
F     644  114   248   120   272  0.369  0.798  0.355  0.443    0     0     0   335
A      500    0     0   335     0  0.000  0.000  0.523 -0.523    0     0     0   335
J     459    0     0     0   334  0.000  0.512  0.000  0.512    0     0     0   335
K      459    0     0     0   334  0.000  0.409  0.000  0.409    0     0     0   335
E      335    0     0     0     0  0.000  0.000  0.000  0.000    0     0     0   335
C      355   66    39    96   120 -0.255  0.277  0.269  0.009    0     0     0   335
I     162     3     0    79    32 -0.750  0.115  0.292 -0.177    0     0     0   335
H      145    5     0    64    21 -0.833  0.089  0.278 -0.189    0     0     0   335
M       213   14     0   145     0 -0.933  0.000  0.471 -0.471    0     0     0   335
B      355   27     0   191    40 -0.964  0.059  0.404 -0.345    0     0     0   335
D      335  106     0   335     0 -0.991  0.000  0.795 -0.795    0     0     0   335

F Freq: 644
       #B    A<B    A>B A<<<B A>>>B  A-->B    #A
G      335    0   249     0   335  0.996  0.940  0.000  0.940    0     0     0   644
A      500    0     0   335     0  0.000  0.000  0.329 -0.329    0     0     0   644
J     459    0     0     0   334  0.000  0.505  0.000  0.505    0     0     0   644
K      459    0     0     0   334  0.000  0.404  0.000  0.404    0     0     0   644
F     644  221   221   309   309  0.000  0.448  0.448  0.000    0     0     0   644
I     162     4     3    82    32 -0.125  0.140  0.292 -0.152    0     0     0   644
H      145    3     2    65    21 -0.167  0.111  0.260 -0.149    0     0     0   644
C      355   81    55   143   123 -0.190  0.289  0.368 -0.079    0     0     0   644
```

```
E      335  248  114  272  120 -0.369  0.355  0.798 -0.443    0    0    0  644
M       213   13    0  145    0 -0.929  0.000  0.441 -0.441    0    0    0  644
B      355   15    0  191   40 -0.938  0.079  0.359 -0.280    0    0    0  644
D      335   59    0  335    0 -0.983  0.000  0.746 -0.746    0    0    0  644

G Freq: 335
       #B    A<B   A>B A<<<B A>>>B  A-->B    #A
J      459    0  253    0  334  0.996  0.920  0.000  0.920    0    0    0  335
B      355    0   32  191   40  0.970  0.112  0.212 -0.100    0    0    0  335
I      162    0   23   82   29  0.958  0.167  0.173 -0.006    0    0    0  335
H       145    1   19   64   19  0.857  0.131  0.169 -0.038    0    0    0  335
A      500    0    0  335    0  0.000  0.000  0.286 -0.286    0    0    0  335
K      459    0    0    0  334  0.000  0.736  0.000  0.736    0    0    0  335
M       213    0    0  145    0  0.000  0.000  0.258 -0.258    0    0    0  335
D      335    0    0  335    0  0.000  0.000  0.435 -0.435    0    0    0  335
G      335    0    0    0    0  0.000  0.000  0.000  0.000    0    0    0  335
C      355   37    7  184   46 -0.667  0.109  0.373 -0.264    0    0    0  335
E      335   48    0  335    0 -0.980  0.000  0.695 -0.695    0    0    0  335
F      644  249    0  335    0 -0.996  0.000  0.940 -0.940    0    0    0  335

Log all tables rD:\TBDK\1S306\research results\little thumb\tot500model8.trc open!
Make an event name list for D:\TBDK\1S306\research results\little thumb\tot500model8.trc.
Total reset all event record information!
12 events names!
Make the basic tables (#A<B, ..., #A-->B) and load D:\TBDK\1S306\research results\little
thumb\tot500model8.trc
Reset Basic tables
Reset all event records information except NAMING and IGNORING!
Event-log properties
500 event lines!
4297 event tokens!
66 different A,B-patterns (complet
```

For separating of activities in every trace, I had to develop a Visual Basic program. This program put between the activities the necessary separated sign, e.g. "," for Little Thumb.

```
Sub aanmaken_textfile_gescheiden_door_komma()
  Dim wvCasenr, i, j, j1, j2, k As Integer
  Dim wvlog, wvlogd1, wvtijd, wvcomplete, wvcontrole As String

 'Schoonvegen van worksheet
  For i = 1 To 200
    For j = 10 To 15
      Cells(i, j).Clear
      Cells(i, j).Interior.ColorIndex = xlNone
    Next j
  Next i

  j = 1
  i = 0
  k = 1
  wvlog = ""
  wvlogd1 = ""
  wvtijd = ""
  wvcontrole = "start"
  Do Until (wvcontrole = "")
    i = i + 1
    j = Cells(i, 1)
    j1 = j
    j2 = Cells(i + 1, 1)
    If j1 = j2 Then
        wvlogd1 = Cells(i + 1, 2)
        wvlog = wvlog & "," & wvlogd1
        'MsgBox (wvlog)
    Else
      If i > 2 Then
        Cells(k + 1, 10) = j1
        Cells(k + 1, 11) = wvlog
        wvtijd = Cells(i, 3)
        wvtijd = Mid(wvtijd, 12, 6)
        Cells(k + 1, 12) = wvtijd
        wvorg = Cells(i, 2)
        If wvorg = "K" Then
          Cells(k + 1, 13) = "Yes"
        Else
          Cells(k + 1, 13) = "No"
        End If
        k = k + 1
      Else
        Cells(1, 10) = "Case Id"
        Cells(1, 11) = "Event log"
        Cells(1, 12) = "Duration (hh:mm)"
        Cells(1, 13) = "Complete (Y/N)"
      End If
      wvlog = Cells(i + 1, 2)
    End If
    wvcontrole = Cells(i, 1)
  Loop
  MsgBox ("Operatie aanmaken event log is gereed!")
End Sub
```

After running the above mentioned VB program, the logfile looks like:

```
A,H,B,C,J,K
A,D,E,F,F,G,J,K
A
A,H,D,B,C,E,F,G,J,K
A,H,M
A,H,B,D,E,F,G,C,B,C,J,K
A,I,B,M,D,F,E,C,F,F,G,J,K
A,B,D,C,E,F,G,B,C,J,K
A,D,M,B,E,C,F,G,B,C,J,K
A,B,C,B,C,J,K
A,B,C,J,K
A,I,B,C,J,K
A,H,J,K
A,B,C,J,K
A,H,B,C,B,C,B,C,J,K
A,M,B,D,E,F,C,G,B,C,J,K
A
…
```

A.9 The Correlation Coefficient versus IS

This section is about the two selected interesting measures for patterns. Both metrics are described in the article of [34]. It describes the characteristics of both measures and their differences according a calculation example. The correlation and IS indicate the dependency between tasks.

A.9.2 The Correlation Coefficient

The correlation, also called Pearson's coefficient, is a well-known metric and intuitively easy to understand. It measures the degree of linear relationship between a pair of random variables (pages 171-177 of [4]). The correlation varies between -1 and 1. Theoretically, it is defined as the covariance between two variables (*cov*($a \Rightarrow b$)), divided by their standard deviations (σ). In our research domain, the variables are the tasks. Thus:

Definition 1: Correlation measures between tasks. *Let W be an event log over T, and A, B* $\in$ *T. The tasks A and B are part of the binary matrix BM.* $\phi(A \Rightarrow B) \in [-1,1]$ *is the correlation between A and B, where* $\sigma_a \neq 0$ *and* $\sigma_b \neq 0$*. General:*

$$\phi(A \Rightarrow B) = \left(\frac{\operatorname{cov}(A \Rightarrow B)}{\sigma_A \cdot \sigma_B}\right) = \left(\frac{P(A \wedge B) - P(A) \cdot P(B)}{\sqrt{P(A) \cdot (1 - P(A))} \cdot \sqrt{P(B) \cdot (1 - P(B))}}\right) \qquad (1.1)$$

where $\phi(A \Rightarrow B) \in [-1,1]$

P(A) and *P(B)* are the probabilities of the tasks *A* and *B*. $P(A \wedge B)$ is the probability of the combination of the tasks *A* and *B*. The probability is the total number of occurring of the task in the binary matrix divide by the total rows of the binary matrix. In order words, it is counting the ones of a task in every rows of the binary matrix (i.e. "baskets"). The probability varies between 0 and 1.

Interesting points of the correlation are the values -1, 0 and 1. In the case of a correlation value of -1, the dependency between the two tasks is strong negative. Two direct related tasks are independent, if the correlation value of the intersection equals to 0. In this situation is the probability of the intersection of both tasks equal to its separated probabilities, i.e. $P(A \wedge B) = P(A) \cdot P(B)$. There is a strong relation between two tasks, if the correlation coefficient between two tasks, e.g. *A* and *B*, equals to 1.

In some situations, it is possible that the standard deviation of one or both variables is equal to zero. The determination of the correlation with these tasks and other directly related tasks is impossible. We have to substitute the formula, if the probability of one task is 1 and the probability of the other task is between 0 and 1. The probability of a singular task can not be zero. In this situation, the probability between the task and the direct connected split/join would be zero. In other words, there is no number of occurring between the task and its split and join in a trace (case id). This means, no connection between these tasks. That is a contradiction to the DG, which illustrates the connections between the related tasks.

One of the important characteristics of the correlation coefficient are the boundaries -1 and 1. This must be saved, also after the substitution of the standard deviation by a certain value. This leads us to the definition of the substitution of the correlation values:

Definition 2: The substitution of the correlation values. *Let W be an event log over T, and A, B ∈ T. The tasks A and B are part of the binary matrix BM. If the standard deviation of task a is zero (e.g. σ(a) = 0) or the standard deviation of task b is zero (e.g. σ(b) = 0), the probability of this task will be replaced by the approximated probability value of the remaining task. This leads to the following substitution formulas:*

$$\phi(A \Rightarrow B) = P(B)\text{, } when\ P(A) = 1 \wedge P(B) \in (0,1) \qquad (1.2)$$

$$\phi(A \Rightarrow B) = P(A)\text{, } when\ \ P(B) = 1 \wedge P(A) \in (0,1) \qquad (1.3)$$

where $\phi(A \Rightarrow B) \in [-1,1]$.

Next subsection describes is about the selected statistical measure, e.g. IS.

A.9.3 IS

The second selected statistical measure is the IS, also called Cosine. It has many desirable properties as an interestingness measures. The IS is a product of two important quantities, interest factor[15] and support. This measures takes into account both the interestingness and support aspects of a pattern. A high value of IS indicates a high dependency between two tasks. This leads us to the following definition:

Definition 3: IS between two direct related tasks. *Let W be an event log over T, A, B ∈ T. The tasks A and B are part of the binary matrix BM. Let* $P(A) \neq 0 \wedge P(B) \neq 0$*, then is the interestingness measure IS between the tasks A and B:*

$$IS(A \Rightarrow B) = \frac{P(A \wedge B)}{\sqrt{P(A) \cdot P(B)}}\text{, } where\ \ IS(A,B) \in [0,1] \qquad (1.4)$$

According to formula of IS, there is a problem, if the probability one of the tasks (or both) is zero. In this research domain, the probabilities of the separate tasks can not be zero.

The boundaries of IS are also interesting points, i.e. 0 and 1. The value of IS is equal to zero, if none rows in the binary matrix contain both tasks. The percentage of both tasks in this matrix is zero (i.e. $P(a \wedge b)$ = 0). The value of IS is equal to one, if the binary matrix contains only both tasks. The percentage of both tasks in this matrix is one (i.e. $P(A \wedge B) = P(A) \cdot P(B)$).

IS is one of the null-invariant measures. The null-invariant is an operation that corresponds to adding more records that do not contain the two variables under consideration. This property is useful for domains having sparse data sets, where co-presence of items is more important than

[15] Interest factor is another widely used measure for association patterns [46,47,33,49]. This metric is defined to be the ratio between the joint probability of two variables with respect to their expected probabilities under the independence assumption. The interest factor is a non-negative real number; with a value of 1 corresponding independence. The interest factor I(A⇒B) is closely related to the correlation coefficient. The formula of Interest is:

$$I(A \Rightarrow B) = \frac{P(A \wedge B)}{P(A) \cdot P(B)}$$

co-absence. The next example illustrates nicely the difference between the two last described metrics, e.g. correlation coefficient and IS.

Calculation Example: IS versus Correlation Coefficient
For example, Table 16 shows the binary table of two tasks *A* and *B.* Assume that the total number of rows in the binary matrix is very low, e.g. 10. The value of IS is 0.875 comparing to the correlation coefficient for these tasks is 0.375. The IS is still a good indicator for the dependency between both tasks, conversely the correlation. The number where both columns have the same value are high (i.e. 8), which means a strong dependency between both tasks.

A	***B***
1	*1*
1	*1*
1	*1*
0	*1*
1	*0*
1	*1*
1	*1*
0	*0*
1	*1*
1	*1*

Table 16: Example of a binary table of 2 tasks.

Appendix A.10 illustrates the checks of the exceptions of both metrics in Java.

A.10 Check of the Exceptions of the statistical Measures

Below shows the Java-script, which checks the exceptions of the selected statistical measures, e.g.CI, IS and correlation.

```
import java.awt.Graphics;
import java.lang.*;
import java.awt.*;
import java.util.*;
import java.applet.*;

public class metrics extends Applet{

 DecimalFormat df = new DecimalFormat();

   public void init() {
    df.setMaximumIntegerDigits(1);
    df.setMaximumFractionDigits(4);
   }

   public void paint(Graphics g){
    double p, prob1, prob2, prob12;

    System.out.println("Testing IS and Correlation");

    prob1 = 1; prob2 = 1; prob12 = 1;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.001; prob2 = 0.001; prob12 = 0;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.001; prob2 = 0.999; prob12 = 0;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.999; prob2 = 0.001; prob12 = 0;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.999; prob2 = 0.999; prob12 = 0.998;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.4; prob2 = 1; prob12 = 0.4;
    except_corr(prob1, prob2, prob12);

    prob1 = 1; prob2 = 0.3; prob12 = 0.4;
    except_corr(prob1, prob2, prob12);

    prob1 = 0.25; prob2 = 0.75; prob12 = 0.1;
    except_corr(prob1, prob2, prob12);
   }

 public void except_corr(double prob1, double prob2, double prob12) {
   double IS=0, Corr=0;

    // IS bepaling
    if (prob1==0 || prob2==0) {IS=0;}
```

```
        if (prob1==0 && prob12==0) {IS=1;}
        if ((!(prob1==0))&& (!(prob2==0))) {IS =
prob12/(Math.sqrt(prob1*prob2));}

        //Correlatie bepaling
        //prob12=0->corr=-1 en prob1=0 en prob2=1!!!
        if ((prob1 == 0 && prob2 == 1 && prob12==0 )|| (prob1==1 &&
prob2==0 && prob12==0))  {Corr=-1;}
        if ((prob1 == 1 && prob2 == 1) || (prob1==0 && prob2==0))
{Corr=1;}
        if ((prob1 == 1)&& (!(prob2 == 0))) {Corr = prob2;}
        if ((prob2 == 1)&& (!(prob1 == 0))) {Corr = prob1;}
        if ((!(prob1 == 0 || prob1 == 1) && (!( prob2 == 0 || prob2 ==
1))))
          {Corr = (prob12-(prob1*prob2))/(Math.sqrt((prob1*prob2*(1-
prob1)*(1-prob2))));}

        System.out.println("prob1: "+prob1+" prob2: "+prob2+" prob12:
"+prob12+" IS: "+df.format(IS)+" Corr: "+df.format(Corr));
  }
}
```

The results of these exceptions are:

```
Testing IS and Correlation
prob1: 1.0 prob2: 1.0 prob12: 1.0 IS: 1 Corr: 1
prob1: 0.0010 prob2: 0.0010 prob12: 0.0 IS: 0 Corr: -0,001
prob1: 0.0010 prob2: 0.999 prob12: 0.0 IS: 0 Corr: -1
prob1: 0.999 prob2: 0.0010 prob12: 0.0 IS: 0 Corr: -1
prob1: 0.999 prob2: 0.999 prob12: 0.998 IS: 0,999 Corr: -0,001
prob1: 0.4 prob2: 1.0 prob12: 0.4 IS: 0,6325 Corr: 0,4
prob1: 1.0 prob2: 0.3 prob12: 0.4 IS: 0,7303 Corr: 0,3
prob1: 0.25 prob2: 0.75 prob12: 0.1 IS: 0,2309 Corr: -0,4667
```

A.11 EPC Model

EPC is an acronym for Event-Driven Process Chains. It is besides Petri Nets (PN) another popular technique for business process modeling. Comparing to PN is their focus rather related to semi-formal process documentation than formal process specification. Figure 41 illustrates the EPC model of event log, according case study 1 (next page).

The symbols in EPC are the following:

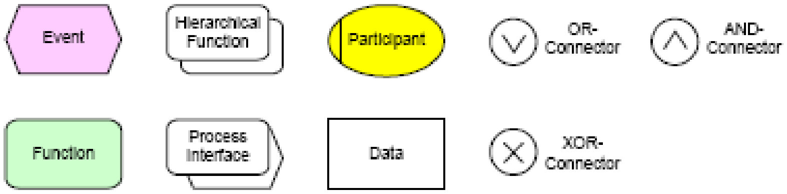


Figure 40: Symbols of EPC.

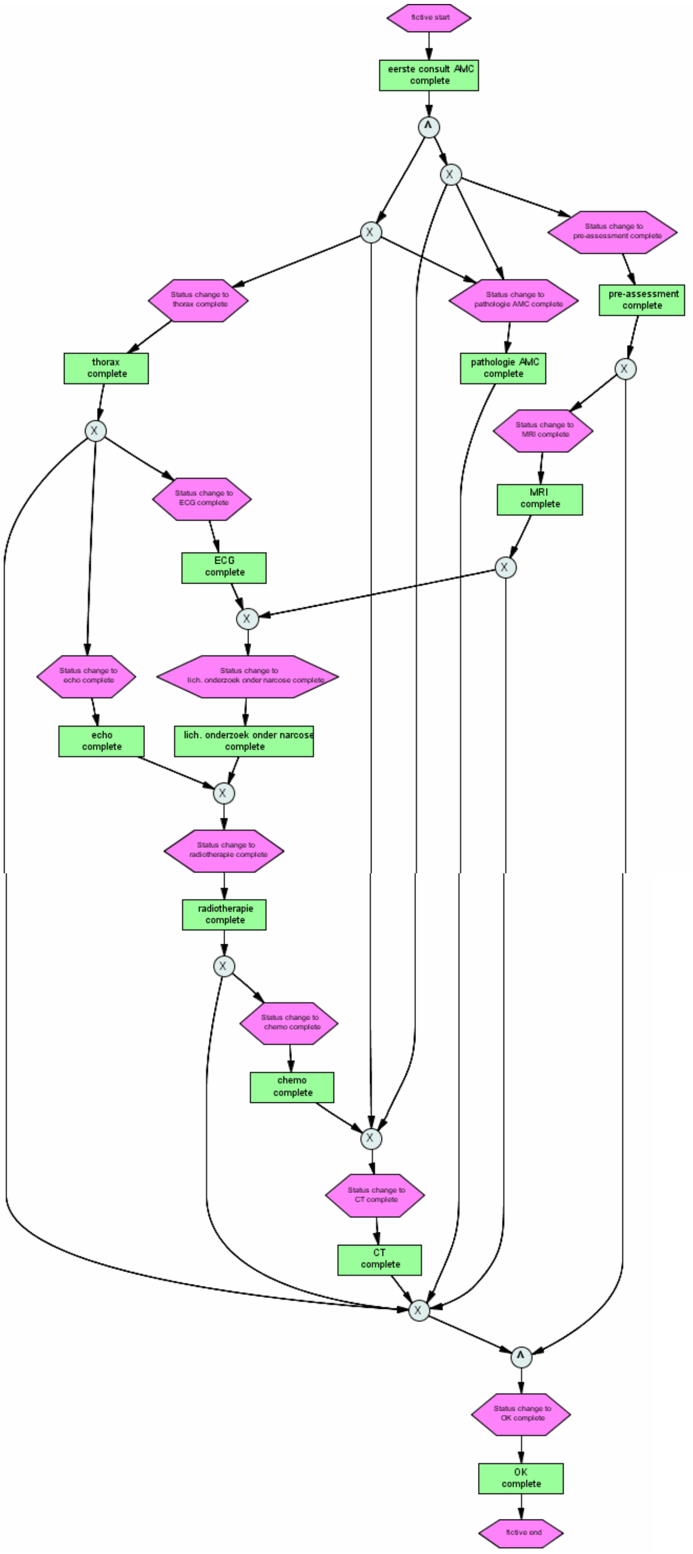
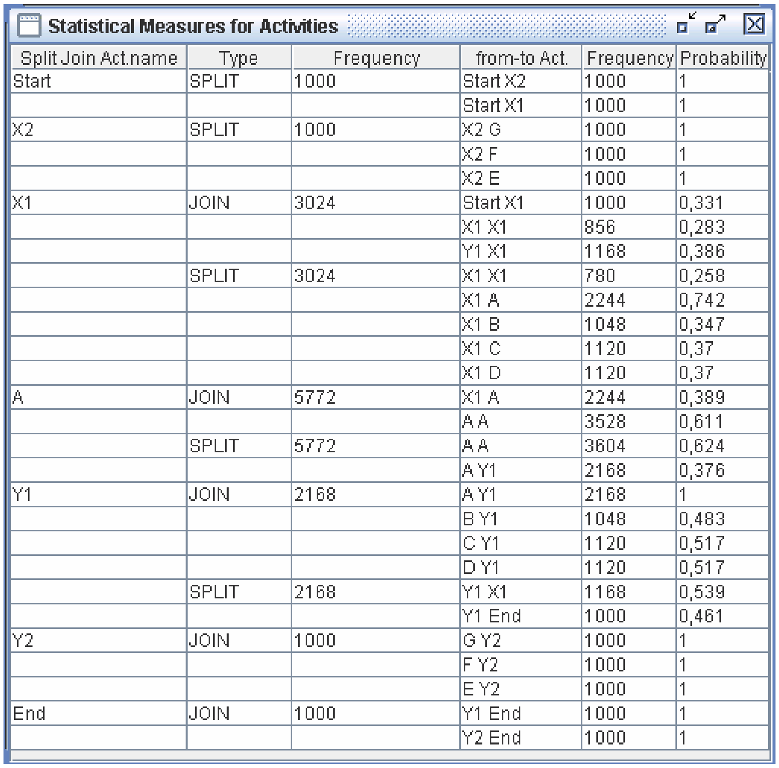

fictive start
eerste consult AMC complete
Status change to pre-assessment complete
Status change to thorax complete
Status change to pathologie AMC complete
pre-assessment complete
thorax complete
pathologie AMC complete
Status change to MRI complete
MRI complete
Status change to ECG complete
ECG complete
Status change to echo complete
Status change to lich. onderzoek onder narcose complete
echo complete
lich. onderzoek onder narcose complete
Status change to radiotherapie complete
radiotherapie complete
Status change to chemo complete
chemo complete
Status change to CT complete
CT complete
Status change to OK complete
OK complete
fictive end

Figure 41: The EPC model of the first medical consult for womb cancer at AMC.

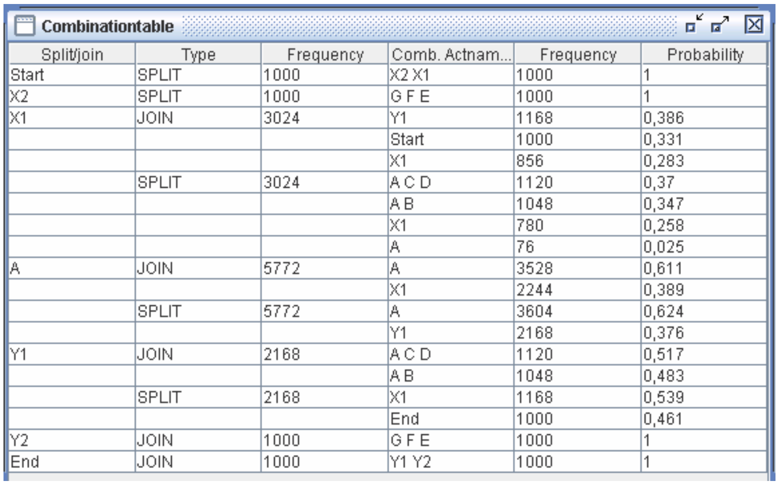
A.12 Statistical Tables of the first Example

This section illustrates the statistical tables of the first example. The first example is based on the PN example described in section 3.1. The log is generated by the process model in section 3.1 of [14]. The event log is one noise free event log with 1,000 traces. Below illustrate the table of the methods.



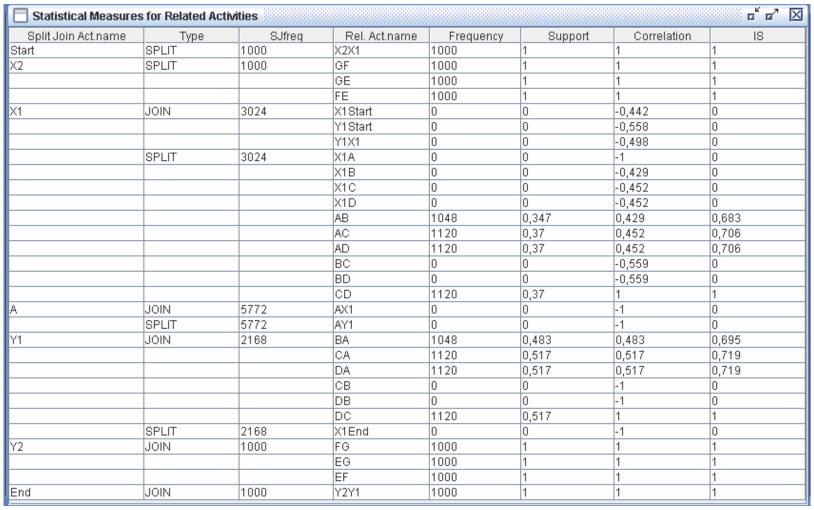
Split Join Act.name	Type	Frequency	from-to Act.	Frequency	Probability
Start	SPLIT	1000	Start X2	1000	1
			Start X1	1000	1
X2	SPLIT	1000	X2 G	1000	1
			X2 F	1000	1
			X2 E	1000	1
X1	JOIN	3024	Start X1	1000	0,331
			X1 X1	856	0,283
			Y1 X1	1168	0,386
	SPLIT	3024	X1 X1	780	0,258
			X1 A	2244	0,742
			X1 B	1048	0,347
			X1 C	1120	0,37
			X1 D	1120	0,37
A	JOIN	5772	X1 A	2244	0,389
			A A	3528	0,611
	SPLIT	5772	A A	3604	0,624
			A Y1	2168	0,376
Y1	JOIN	2168	A Y1	2168	1
			B Y1	1048	0,483
			C Y1	1120	0,517
			D Y1	1120	0,517
	SPLIT	2168	Y1 X1	1168	0,539
			Y1 End	1000	0,461
Y2	JOIN	1000	G Y2	1000	1
			F Y2	1000	1
			E Y2	1000	1
End	JOIN	1000	Y1 End	1000	1
			Y2 End	1000	1

Table 17: The statistical table of the individual activities.



Split/join	Type	Frequency	Comb. Actnam...	Frequency	Probability
Start	SPLIT	1000	X2 X1	1000	1
X2	SPLIT	1000	G F E	1000	1
X1	JOIN	3024	Y1	1168	0,386
			Start	1000	0,331
			X1	856	0,283
	SPLIT	3024	A C D	1120	0,37
			A B	1048	0,347
			X1	780	0,258
			A	76	0,025
A	JOIN	5772	A	3528	0,611
			X1	2244	0,389
	SPLIT	5772	A	3604	0,624
			Y1	2168	0,376
Y1	JOIN	2168	A C D	1120	0,517
			A B	1048	0,483
	SPLIT	2168	X1	1168	0,539
			End	1000	0,461
Y2	JOIN	1000	G F E	1000	1
End	JOIN	1000	Y1 Y2	1000	1

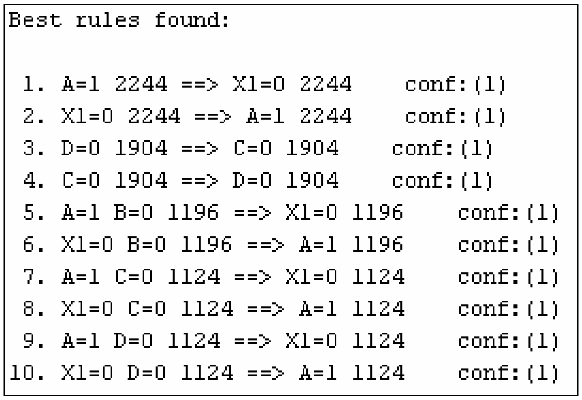
Table 18: The combination table.



Split Join Act.name	Type	SJfreq	Rel. Act.name	Frequency	Support	Correlation	IS
Start	SPLIT	1000	X2X1	1000	1	1	1
X2	SPLIT	1000	GF	1000	1	1	1
			GE	1000	1	1	1
			FE	1000	1	1	1
X1	JOIN	3024	X1Start	0	0	-0,442	0
			Y1Start	0	0	-0,558	0
			Y1X1	0	0	-0,498	0
	SPLIT	3024	X1A	0	0	-1	0
			X1B	0	0	-0,429	0
			X1C	0	0	-0,452	0
			X1D	0	0	-0,452	0
			AB	1048	0,347	0,429	0,683
			AC	1120	0,37	0,452	0,706
			AD	1120	0,37	0,452	0,706
			BC	0	0	-0,559	0
			BD	0	0	-0,559	0
			CD	1120	0,37	1	1
A	JOIN	5772	AX1	0	0	-1	0
	SPLIT	5772	AY1	0	0	-1	0
Y1	JOIN	2168	BA	1048	0,483	0,483	0,695
			CA	1120	0,517	0,517	0,719
			DA	1120	0,517	0,517	0,719
			CB	0	0	-1	0
			DB	0	0	-1	0
			DC	1120	0,517	1	1
	SPLIT	2168	X1End	0	0	-1	0
Y2	JOIN	1000	FG	1000	1	1	1
			EG	1000	1	1	1
			EF	1000	1	1	1
End	JOIN	1000	Y2Y1	1000	1	1	1

Table 19: The table of statistical measures for related activities.

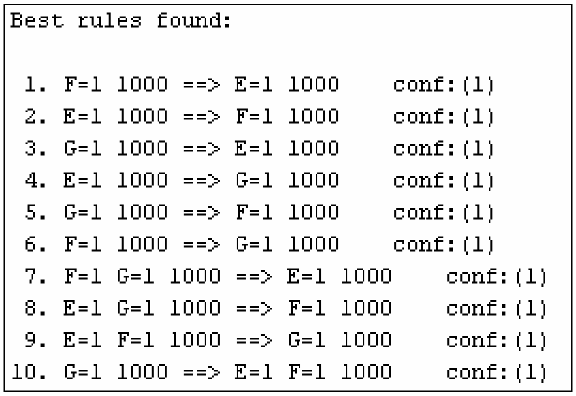
Association rules by using *Apriori* algorithm to split (activity *X1*).



```
Best rules found:

 1. A=1 2244 ==> X1=0 2244    conf:(1)
 2. X1=0 2244 ==> A=1 2244    conf:(1)
 3. D=0 1904 ==> C=0 1904    conf:(1)
 4. C=0 1904 ==> D=0 1904    conf:(1)
 5. A=1 B=0 1196 ==> X1=0 1196    conf:(1)
 6. X1=0 B=0 1196 ==> A=1 1196    conf:(1)
 7. A=1 C=0 1124 ==> X1=0 1124    conf:(1)
 8. X1=0 C=0 1124 ==> A=1 1124    conf:(1)
 9. A=1 D=0 1124 ==> X1=0 1124    conf:(1)
10. X1=0 D=0 1124 ==> A=1 1124    conf:(1)
```

Association rules by using *Apriori* algorithm to split (activity *X2*).



```
Best rules found:

 1. F=1 1000 ==> E=1 1000    conf:(1)
 2. E=1 1000 ==> F=1 1000    conf:(1)
 3. G=1 1000 ==> E=1 1000    conf:(1)
 4. E=1 1000 ==> G=1 1000    conf:(1)
 5. G=1 1000 ==> F=1 1000    conf:(1)
 6. F=1 1000 ==> G=1 1000    conf:(1)
 7. F=1 G=1 1000 ==> E=1 1000    conf:(1)
 8. E=1 G=1 1000 ==> F=1 1000    conf:(1)
 9. E=1 F=1 1000 ==> G=1 1000    conf:(1)
10. G=1 1000 ==> E=1 F=1 1000    conf:(1)
```

A.13 Statistical Tables of the second Example

This section illustrates the statistical tables of the second example. The second example concerns the statistical information of an artificial data. The log is generated by the process model in section 3.1 of [14]. The event log is one noise free event log with 1,000 traces. The parameter settings of the HM are changed as follows: relative to best threshold = 0.2, positive observations = 3, dependency threshold = 0.85. Below illustrate the DG, the HN, and the table of the methods.

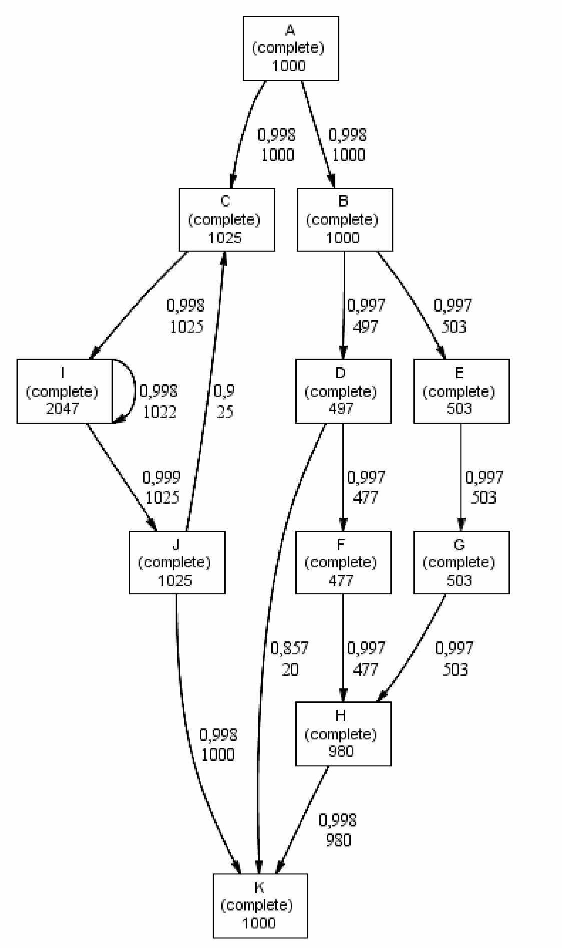


Figure 42: The DG of the first example.

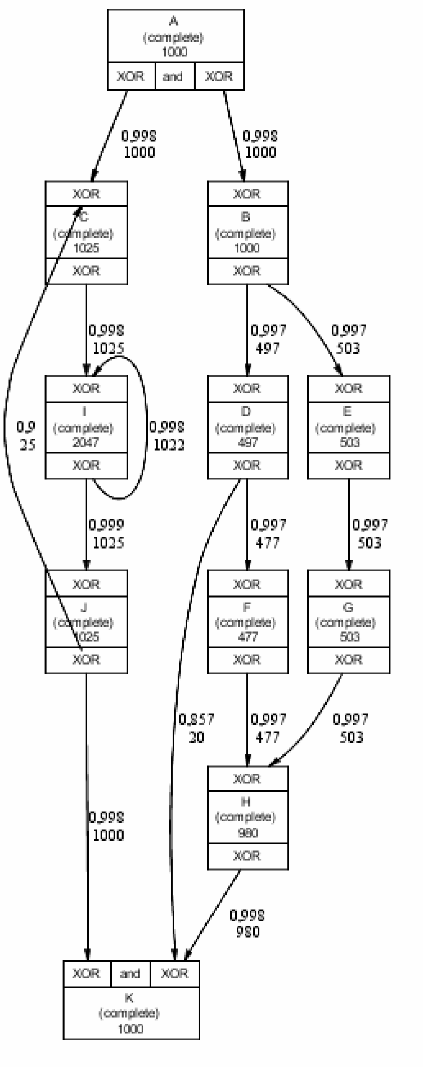
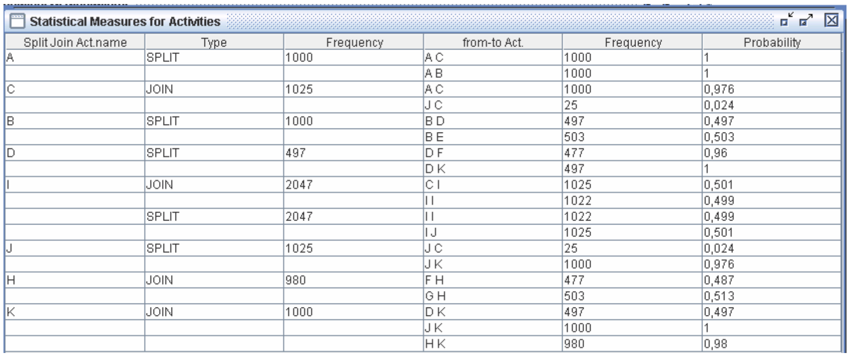
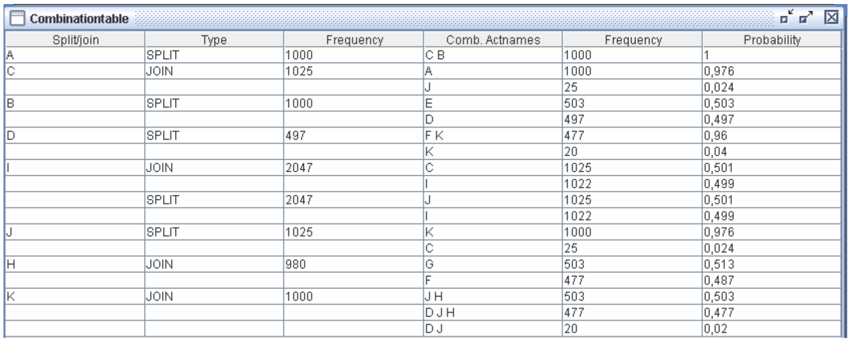


Figure 43: The HN of the first example.



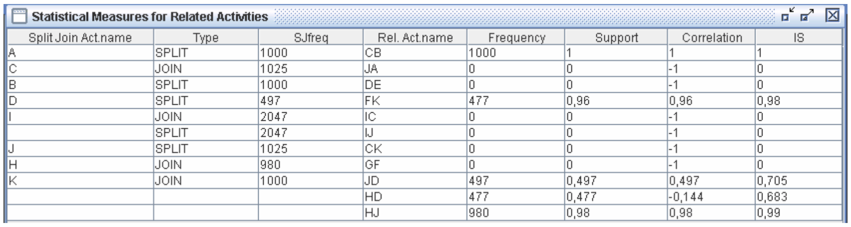
Split Join Act.name	Type	Frequency	from-to Act.	Frequency	Probability
A	SPLIT	1000	A C	1000	1
			A B	1000	1
C	JOIN	1025	A C	1000	0,976
			J C	25	0,024
B	SPLIT	1000	B D	497	0,497
			B E	503	0,503
D	SPLIT	497	D F	477	0,96
			D K	497	1
I	JOIN	2047	C I	1025	0,501
			I I	1022	0,499
	SPLIT	2047	I I	1022	0,499
			I J	1025	0,501
J	SPLIT	1025	J C	25	0,024
			J K	1000	0,976
H	JOIN	980	F H	477	0,487
			G H	503	0,513
K	JOIN	1000	D K	497	0,497
			J K	1000	1
			H K	980	0,98

Table 20: The statistical table of the individual activities.



Split/join	Type	Frequency	Comb. Actnames	Frequency	Probability
A	SPLIT	1000	C B	1000	1
C	JOIN	1025	A	1000	0,976
			J	25	0,024
B	SPLIT	1000	E	503	0,503
			D	497	0,497
D	SPLIT	497	F K	477	0,96
			K	20	0,04
I	JOIN	2047	C	1025	0,501
			I	1022	0,499
	SPLIT	2047	J	1025	0,501
			I	1022	0,499
J	SPLIT	1025	K	1000	0,976
			C	25	0,024
H	JOIN	980	G	503	0,513
			F	477	0,487
K	JOIN	1000	J H	503	0,503
			D J H	477	0,477
			D J	20	0,02

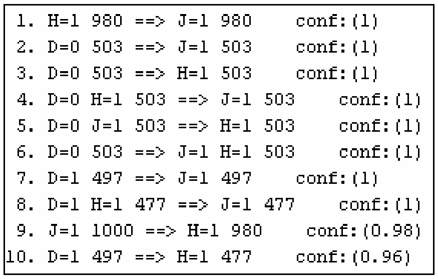
Table 21: The combination table.



Split Join Act.name	Type	SJfreq	Rel. Act.name	Frequency	Support	Correlation	IS
A	SPLIT	1000	CB	1000	1	1	1
C	JOIN	1025	JA	0	0	-1	0
B	SPLIT	1000	DE	0	0	-1	0
D	SPLIT	497	FK	477	0,96	0,96	0,98
I	JOIN	2047	IC	0	0	-1	0
	SPLIT	2047	IJ	0	0	-1	0
J	SPLIT	1025	CK	0	0	-1	0
H	JOIN	980	GF	0	0	-1	0
K	JOIN	1000	JD	497	0,497	0,497	0,705
			HD	477	0,477	-0,144	0,683
			HJ	980	0,98	0,98	0,99

Table 22: The table of the statistical measures for related activities.

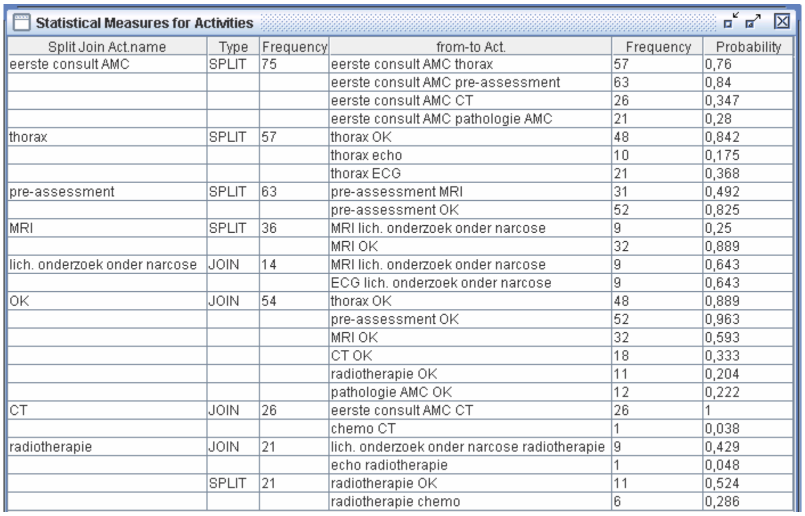
Association rules by using *Apriori* algorithm to join (activity *K*).



```
 1. H=1 980 ==> J=1 980    conf:(1)
 2. D=0 503 ==> J=1 503    conf:(1)
 3. D=0 503 ==> H=1 503    conf:(1)
 4. D=0 H=1 503 ==> J=1 503    conf:(1)
 5. D=0 J=1 503 ==> H=1 503    conf:(1)
 6. D=0 503 ==> J=1 H=1 503    conf:(1)
 7. D=1 497 ==> J=1 497    conf:(1)
 8. D=1 H=1 477 ==> J=1 477    conf:(1)
 9. J=1 1000 ==> H=1 980    conf:(0.98)
10. D=1 497 ==> H=1 477    conf:(0.96)
```

A.14 Statistical Tables of the first Case Study

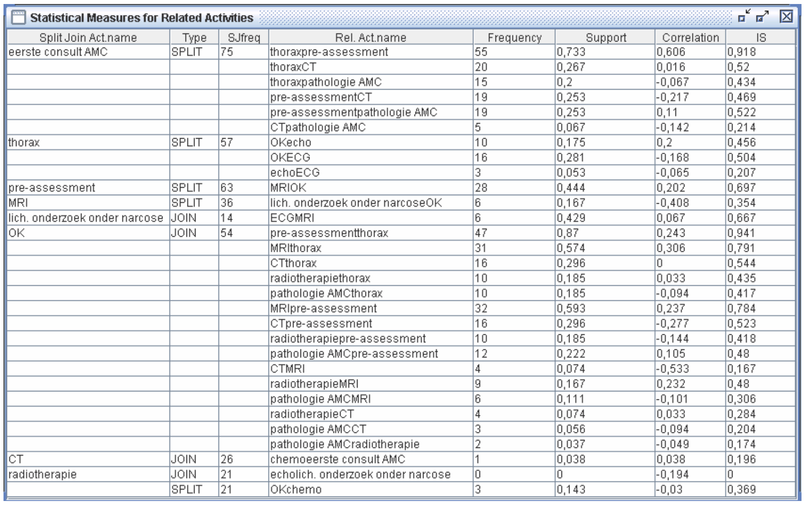
The first case study is about the first medical consult for womb cancer, which took place at the academic hospital in Amsterdam, called “AMC” [55,72]. 75 female patients had a first consult during the period March – April 2007. The first consult take place on the gynecological oncology and concerns research for possible cancer in the womb. This treatment contains the following activities: first consult *(eerste consult AMC*), pre-assessment, thorax, pathology (*pathologie AMC*), MRI, ECG, echo, body research under narcosis (*lich. onderzoek onder narcose*), radiotherapy (*radiotherapie*), chemo, CT and finally OK. The result of the first consult determines which activities will be executed next for the patient during treatment of the patient. Below illustrates the statistical results of the four mining techniques.



Statistical Measures for Activities

Split Join Act.name	Type	Frequency	from-to Act.	Frequency	Probability
eerste consult AMC	SPLIT	75	eerste consult AMC thorax	57	0,76
			eerste consult AMC pre-assessment	63	0,84
			eerste consult AMC CT	26	0,347
			eerste consult AMC pathologie AMC	21	0,28
thorax	SPLIT	57	thorax OK	48	0,842
			thorax echo	10	0,175
			thorax ECG	21	0,368
pre-assessment	SPLIT	63	pre-assessment MRI	31	0,492
			pre-assessment OK	52	0,825
MRI	SPLIT	36	MRI lich. onderzoek onder narcose	9	0,25
			MRI OK	32	0,889
lich. onderzoek onder narcose	JOIN	14	MRI lich. onderzoek onder narcose	9	0,643
			ECG lich. onderzoek onder narcose	9	0,643
OK	JOIN	54	thorax OK	48	0,889
			pre-assessment OK	52	0,963
			MRI OK	32	0,593
			CT OK	18	0,333
			radiotherapie OK	11	0,204
			pathologie AMC OK	12	0,222
CT	JOIN	26	eerste consult AMC CT	26	1
			chemo CT	1	0,038
radiotherapie	JOIN	21	lich. onderzoek onder narcose radiotherapie	9	0,429
			echo radiotherapie	1	0,048
	SPLIT	21	radiotherapie OK	11	0,524
			radiotherapie chemo	6	0,286

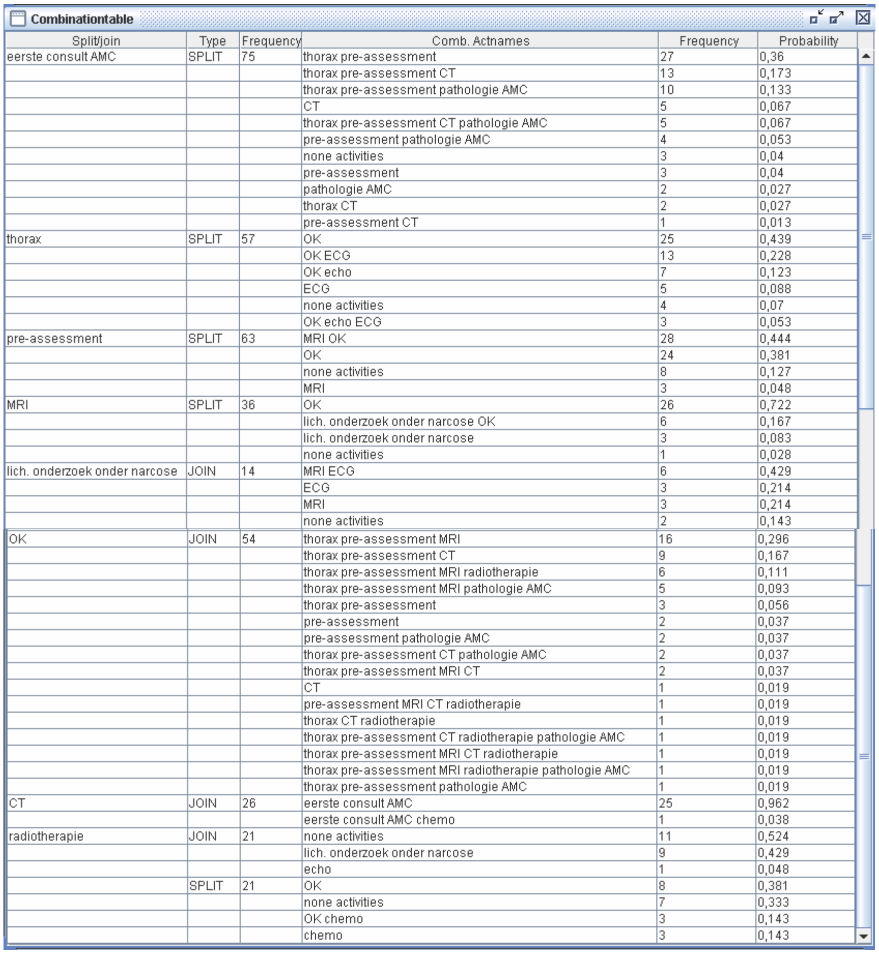
Table 23: The table of the statistical measures for individual activities.



Statistical Measures for Related Activities

Split Join Act.name	Type	SJfreq	Rel. Act.name	Frequency	Support	Correlation	IS
eerste consult AMC	SPLIT	75	thoraxpre-assessment	55	0,733	0,606	0,918
			thoraxCT	20	0,267	0,016	0,52
			thoraxpathologie AMC	15	0,2	-0,067	0,434
			pre-assessmentCT	19	0,253	-0,217	0,469
			pre-assessmentpathologie AMC	19	0,253	0,11	0,522
			CTpathologie AMC	5	0,067	-0,142	0,214
thorax	SPLIT	57	OKecho	10	0,175	0,2	0,456
			OKECG	16	0,281	-0,168	0,504
			echoECG	3	0,053	-0,065	0,207
pre-assessment	SPLIT	63	MRIOK	28	0,444	0,202	0,697
MRI	SPLIT	36	lich. onderzoek onder narcoseOK	6	0,167	-0,408	0,354
lich. onderzoek onder narcose	JOIN	14	ECGMRI	6	0,429	0,067	0,667
OK	JOIN	54	pre-assessmentthorax	47	0,87	0,243	0,941
			MRIthorax	31	0,574	0,306	0,791
			CTthorax	16	0,296	0	0,544
			radiotherapiethorax	10	0,185	0,033	0,435
			pathologie AMCthorax	10	0,185	-0,094	0,417
			MRIpre-assessment	32	0,593	0,237	0,784
			CTpre-assessment	16	0,296	-0,277	0,523
			radiotherapiepre-assessment	10	0,185	-0,144	0,418
			pathologie AMCpre-assessment	12	0,222	0,105	0,48
			CTMRI	4	0,074	-0,533	0,167
			radiotherapieMRI	9	0,167	0,232	0,48
			pathologie AMCMRI	6	0,111	-0,101	0,306
			radiotherapieCT	4	0,074	0,033	0,284
			pathologie AMCCT	3	0,056	-0,094	0,204
			pathologie AMCradiotherapie	2	0,037	-0,049	0,174
CT	JOIN	26	chemoeerste consult AMC	1	0,038	0,038	0,196
radiotherapie	JOIN	21	echolich. onderzoek onder narcose	0	0	-0,194	0
	SPLIT	21	OKchemo	3	0,143	-0,03	0,369

Table 24: The table of the statistical measures for related activities.

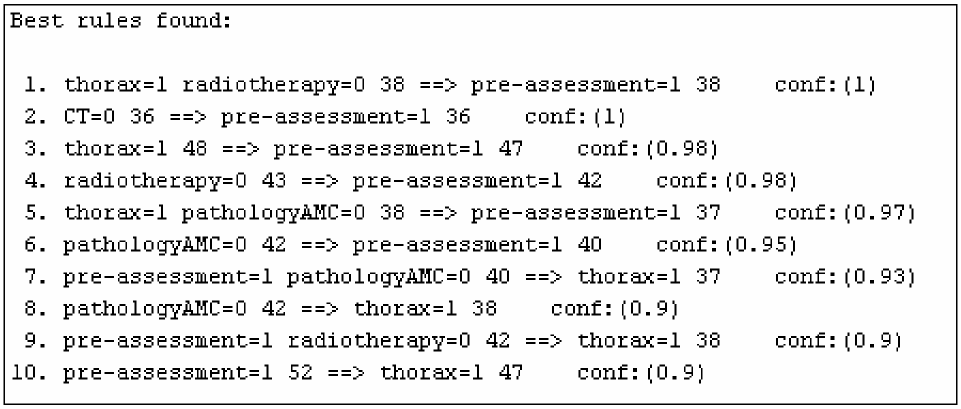


Combinationtable

Split/join	Type	Frequency	Comb. Actnames	Frequency	Probability
eerste consult AMC	SPLIT	75	thorax pre-assessment	27	0,36
			thorax pre-assessment CT	13	0,173
			thorax pre-assessment pathologie AMC	10	0,133
			CT	5	0,067
			thorax pre-assessment CT pathologie AMC	5	0,067
			pre-assessment pathologie AMC	4	0,053
			none activities	3	0,04
			pre-assessment	3	0,04
			pathologie AMC	2	0,027
			thorax CT	2	0,027
			pre-assessment CT	1	0,013
thorax	SPLIT	57	OK	25	0,439
			OK ECG	13	0,228
			OK echo	7	0,123
			ECG	5	0,088
			none activities	4	0,07
			OK echo ECG	3	0,053
pre-assessment	SPLIT	63	MRI OK	28	0,444
			OK	24	0,381
			none activities	8	0,127
			MRI	3	0,048
MRI	SPLIT	36	OK	26	0,722
			lich. onderzoek onder narcose OK	6	0,167
			lich. onderzoek onder narcose	3	0,083
			none activities	1	0,028
lich. onderzoek onder narcose	JOIN	14	MRI ECG	6	0,429
			ECG	3	0,214
			MRI	3	0,214
			none activities	2	0,143
OK	JOIN	54	thorax pre-assessment MRI	16	0,296
			thorax pre-assessment CT	9	0,167
			thorax pre-assessment MRI radiotherapie	6	0,111
			thorax pre-assessment MRI pathologie AMC	5	0,093
			thorax pre-assessment	3	0,056
			pre-assessment	2	0,037
			pre-assessment pathologie AMC	2	0,037
			thorax pre-assessment CT pathologie AMC	2	0,037
			thorax pre-assessment MRI CT	2	0,037
			CT	1	0,019
			pre-assessment MRI CT radiotherapie	1	0,019
			thorax CT radiotherapie	1	0,019
			thorax pre-assessment CT radiotherapie pathologie AMC	1	0,019
			thorax pre-assessment MRI CT radiotherapie	1	0,019
			thorax pre-assessment MRI radiotherapie pathologie AMC	1	0,019
			thorax pre-assessment pathologie AMC	1	0,019
CT	JOIN	26	eerste consult AMC	25	0,962
			eerste consult AMC chemo	1	0,038
radiotherapie	JOIN	21	none activities	11	0,524
			lich. onderzoek onder narcose	9	0,429
			echo	1	0,048
	SPLIT	21	OK	8	0,381
			none activities	7	0,333
			OK chemo	3	0,143
			chemo	3	0,143

Table 25: The combination table.

Association rules by using *Apriori* algorithm to join (activity *OK*).



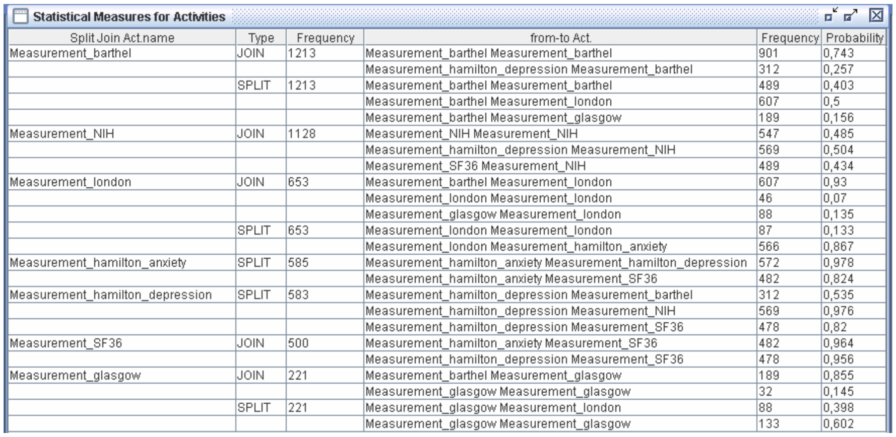
```
Best rules found:

 1. thorax=1 radiotherapy=0 38 ==> pre-assessment=1 38    conf:(1)
 2. CT=0 36 ==> pre-assessment=1 36    conf:(1)
 3. thorax=1 48 ==> pre-assessment=1 47    conf:(0.98)
 4. radiotherapy=0 43 ==> pre-assessment=1 42    conf:(0.98)
 5. thorax=1 pathologyAMC=0 38 ==> pre-assessment=1 37    conf:(0.97)
 6. pathologyAMC=0 42 ==> pre-assessment=1 40    conf:(0.95)
 7. pre-assessment=1 pathologyAMC=0 40 ==> thorax=1 37    conf:(0.93)
 8. pathologyAMC=0 42 ==> thorax=1 38    conf:(0.9)
 9. pre-assessment=1 radiotherapy=0 42 ==> thorax=1 38    conf:(0.9)
10. pre-assessment=1 52 ==> thorax=1 47    conf:(0.9)
```

A.15 Statistical Tables of the second Case Study

The second case study is about data from a preliminary study, which was conducted on patients with acute stroke[16] and transient ischemic attack on first-ever stroke patients in four districts in the region of Lombardia in Italy [50]. It aimed at studying the effect of the American Heart Association guidelines on 386 such patients. The data contains information of patients suffering from the stroke. This information is recorded from the acute phase to the sub-acute phases of the patients from the stroke. Acute phase data pertains to the data of patients that arrive from the stroke symptoms onset. After the first six hours, the patient is considered to be in the sub-acute phase.

The event log contains information about various measurements. These are performed on stroke patients. The measurements take place after admission or during the hospitalization of the patient. There are seven types of measurements, also called scales. Below illustrates the statistical results of the four mining techniques.

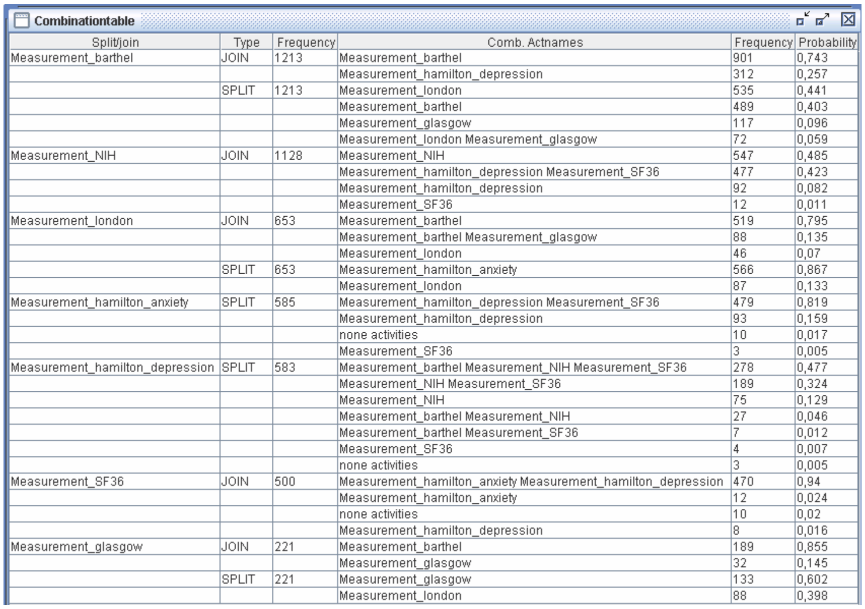


Statistical Measures for Activities

Split Join Act.name	Type	Frequency	from-to Act.	Frequency	Probability
Measurement_barthel	JOIN	1213	Measurement_barthel Measurement_barthel	901	0,743
			Measurement_hamilton_depression Measurement_barthel	312	0,257
	SPLIT	1213	Measurement_barthel Measurement_barthel	489	0,403
			Measurement_barthel Measurement_london	607	0,5
			Measurement_barthel Measurement_glasgow	189	0,156
Measurement_NIH	JOIN	1128	Measurement_NIH Measurement_NIH	547	0,485
			Measurement_hamilton_depression Measurement_NIH	569	0,504
			Measurement_SF36 Measurement_NIH	489	0,434
Measurement_london	JOIN	653	Measurement_barthel Measurement_london	607	0,93
			Measurement_london Measurement_london	46	0,07
			Measurement_glasgow Measurement_london	88	0,135
	SPLIT	653	Measurement_london Measurement_london	87	0,133
			Measurement_london Measurement_hamilton_anxiety	566	0,867
Measurement_hamilton_anxiety	SPLIT	585	Measurement_hamilton_anxiety Measurement_hamilton_depression	572	0,978
			Measurement_hamilton_anxiety Measurement_SF36	482	0,824
Measurement_hamilton_depression	SPLIT	583	Measurement_hamilton_depression Measurement_barthel	312	0,535
			Measurement_hamilton_depression Measurement_NIH	569	0,976
			Measurement_hamilton_depression Measurement_SF36	478	0,82
Measurement_SF36	JOIN	500	Measurement_hamilton_anxiety Measurement_SF36	482	0,964
			Measurement_hamilton_depression Measurement_SF36	478	0,956
Measurement_glasgow	JOIN	221	Measurement_barthel Measurement_glasgow	189	0,855
			Measurement_glasgow Measurement_glasgow	32	0,145
	SPLIT	221	Measurement_glasgow Measurement_london	88	0,398
			Measurement_glasgow Measurement_glasgow	133	0,602

Table 26: The statistical table of the individual activities.

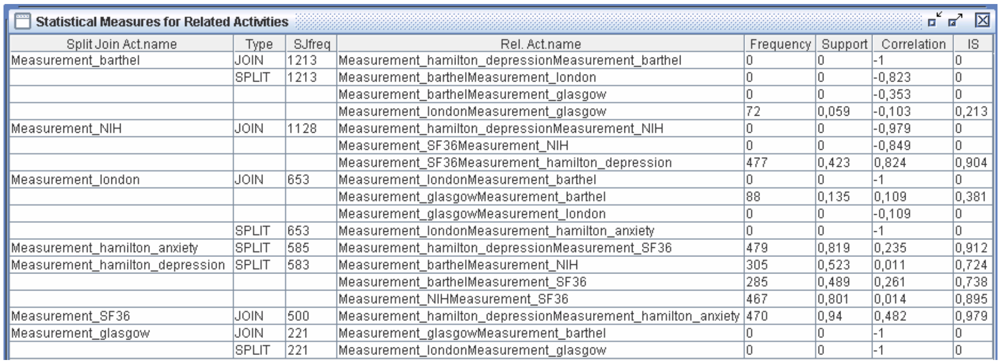
[16] Acute stroke is a vascular condition that precipitates neurological damage and is the second leading cause of death in industrialized countries [56].



Combinationtable

Split/join	Type	Frequency	Comb. Actnames	Frequency	Probability
Measurement_barthel	JOIN	1213	Measurement_barthel	901	0,743
			Measurement_hamilton_depression	312	0,257
	SPLIT	1213	Measurement_london	535	0,441
			Measurement_barthel	489	0,403
			Measurement_glasgow	117	0,096
			Measurement_london Measurement_glasgow	72	0,059
Measurement_NIH	JOIN	1128	Measurement_NIH	547	0,485
			Measurement_hamilton_depression Measurement_SF36	477	0,423
			Measurement_hamilton_depression	92	0,082
			Measurement_SF36	12	0,011
Measurement_london	JOIN	653	Measurement_barthel	519	0,795
			Measurement_barthel Measurement_glasgow	88	0,135
			Measurement_london	46	0,07
	SPLIT	653	Measurement_hamilton_anxiety	566	0,867
			Measurement_london	87	0,133
Measurement_hamilton_anxiety	SPLIT	585	Measurement_hamilton_depression Measurement_SF36	479	0,819
			Measurement_hamilton_depression	93	0,159
			none activities	10	0,017
			Measurement_SF36	3	0,005
Measurement_hamilton_depression	SPLIT	583	Measurement_barthel Measurement_NIH Measurement_SF36	278	0,477
			Measurement_NIH Measurement_SF36	189	0,324
			Measurement_NIH	75	0,129
			Measurement_barthel Measurement_NIH	27	0,046
			Measurement_barthel Measurement_SF36	7	0,012
			Measurement_SF36	4	0,007
			none activities	3	0,005
Measurement_SF36	JOIN	500	Measurement_hamilton_anxiety Measurement_hamilton_depression	470	0,94
			Measurement_hamilton_anxiety	12	0,024
			none activities	10	0,02
			Measurement_hamilton_depression	8	0,016
Measurement_glasgow	JOIN	221	Measurement_barthel	189	0,855
			Measurement_glasgow	32	0,145
	SPLIT	221	Measurement_glasgow	133	0,602
			Measurement_london	88	0,398

Table 27: The combination table.

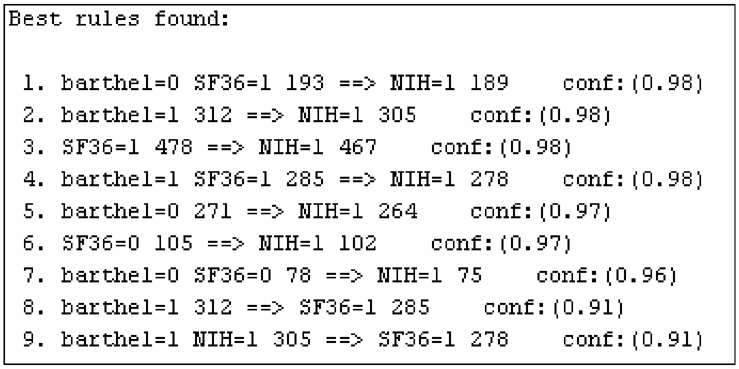


Statistical Measures for Related Activities

Split Join Act.name	Type	SJfreq	Rel. Act.name	Frequency	Support	Correlation	IS
Measurement_barthel	JOIN	1213	Measurement_hamilton_depressionMeasurement_barthel	0	0	-1	0
	SPLIT	1213	Measurement_barthelMeasurement_london	0	0	-0,823	0
			Measurement_barthelMeasurement_glasgow	0	0	-0,353	0
			Measurement_londonMeasurement_glasgow	72	0,059	-0,103	0,213
Measurement_NIH	JOIN	1128	Measurement_hamilton_depressionMeasurement_NIH	0	0	-0,979	0
			Measurement_SF36Measurement_NIH	0	0	-0,849	0
			Measurement_SF36Measurement_hamilton_depression	477	0,423	0,824	0,904
Measurement_london	JOIN	653	Measurement_londonMeasurement_barthel	0	0	-1	0
			Measurement_glasgowMeasurement_barthel	88	0,135	0,109	0,381
			Measurement_glasgowMeasurement_london	0	0	-0,109	0
	SPLIT	653	Measurement_londonMeasurement_hamilton_anxiety	0	0	-1	0
Measurement_hamilton_anxiety	SPLIT	585	Measurement_hamilton_depressionMeasurement_SF36	479	0,819	0,235	0,912
Measurement_hamilton_depression	SPLIT	583	Measurement_barthelMeasurement_NIH	305	0,523	0,011	0,724
			Measurement_barthelMeasurement_SF36	285	0,489	0,261	0,738
			Measurement_NIHMeasurement_SF36	467	0,801	0,014	0,895
Measurement_SF36	JOIN	500	Measurement_hamilton_depressionMeasurement_hamilton_anxiety	470	0,94	0,482	0,979
Measurement_glasgow	JOIN	221	Measurement_glasgowMeasurement_barthel	0	0	-1	0
	SPLIT	221	Measurement_londonMeasurement_glasgow	0	0	-1	0

Table 28: The table of statistical measures for related activities.

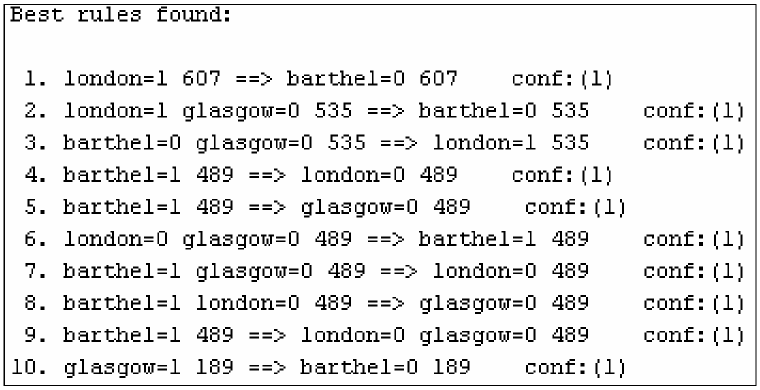
Association rules by using *Apriori* algorithm to split (activity *Hamilton depression*).



```
Best rules found:

 1. barthel=0 SF36=1 193 ==> NIH=1 189    conf:(0.98)
 2. barthel=1 312 ==> NIH=1 305    conf:(0.98)
 3. SF36=1 478 ==> NIH=1 467    conf:(0.98)
 4. barthel=1 SF36=1 285 ==> NIH=1 278    conf:(0.98)
 5. barthel=0 271 ==> NIH=1 264    conf:(0.97)
 6. SF36=0 105 ==> NIH=1 102    conf:(0.97)
 7. barthel=0 SF36=0 78 ==> NIH=1 75    conf:(0.96)
 8. barthel=1 312 ==> SF36=1 285    conf:(0.91)
 9. barthel=1 NIH=1 305 ==> SF36=1 278    conf:(0.91)
```

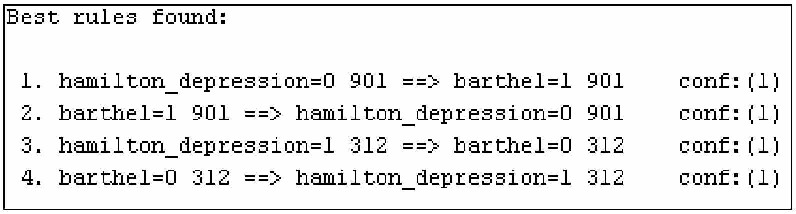
Association rules by using *Apriori* algorithm to split (activity *Barthel*).



```
Best rules found:

 1. london=1 607 ==> barthel=0 607    conf:(1)
 2. london=1 glasgow=0 535 ==> barthel=0 535    conf:(1)
 3. barthel=0 glasgow=0 535 ==> london=1 535    conf:(1)
 4. barthel=1 489 ==> london=0 489    conf:(1)
 5. barthel=1 489 ==> glasgow=0 489    conf:(1)
 6. london=0 glasgow=0 489 ==> barthel=1 489    conf:(1)
 7. barthel=1 glasgow=0 489 ==> london=0 489    conf:(1)
 8. barthel=1 london=0 489 ==> glasgow=0 489    conf:(1)
 9. barthel=1 489 ==> london=0 glasgow=0 489    conf:(1)
10. glasgow=1 189 ==> barthel=0 189    conf:(1)
```

Association rules by using *Apriori* algorithm to join (activity *Barthel*).



```
Best rules found:

 1. hamilton_depression=0 901 ==> barthel=1 901    conf:(1)
 2. barthel=1 901 ==> hamilton_depression=0 901    conf:(1)
 3. hamilton_depression=1 312 ==> barthel=0 312    conf:(1)
 4. barthel=0 312 ==> hamilton_depression=1 312    conf:(1)
```

A.16 Statistical Analyses of the first Consult at AMC

This section contains two subsections, e.g. factor analysis and fork analysis.

A.16.1 Factor Analysis

This subsection describes the factor analysis, which is performed by the popular statistical software program, called SPSS. It is about the event log of case study 1, e.g. OK of the female patients (join). Below shows the SPSS results of the measures support, correlation and IS.

	Correlation between Vectors of Values		
	support	correlation coefficient	IS
Support	1.000	0.639	0.957
correlation coefficient	0.639	1.000	0.685
IS	0.957	0.685	1.000

Table 29: The correlation between the three selected metrics.

It illustrates a high positive degree of dependency between these metrics in case study 1.

A.16.2 Fork Analysis

This subsection is about the statistical analysis between the activities thorax and pre-assessment. Both are interesting treatments of case study 1. The first part of this analysis is the number of combinations versus the probability in relation to their split (first consult AMC) and join (OK).

case study 1: split	*eerste consult AMC (number=75)*			
number of combinations	*pre-assessment*	*thorax*	*p(pre-assessment)*	*p(thorax)*
1	*3*	*0*	*0.040*	*0.000*
2	*32*	*29*	*0.427*	*0.387*
3	*23*	*23*	*0.307*	*0.307*
4	*5*	*5*	*0.067*	*0.067*

split	*eerste consult AMC (number=54)*			
number of combinations	*pre-assessment*	*thorax*	*p(pre-assessment)*	*p(thorax)*
1	*2*	*0*	*0.027*	*0.000*
2	*5*	*3*	*0.067*	*0.040*
3	*26*	*27*	*0.347*	*0.360*
4	*16*	*15*	*0.213*	*0.200*
5	*3*	*3*	*0.040*	*0.040*

Table 30: The table of the fork analysis.

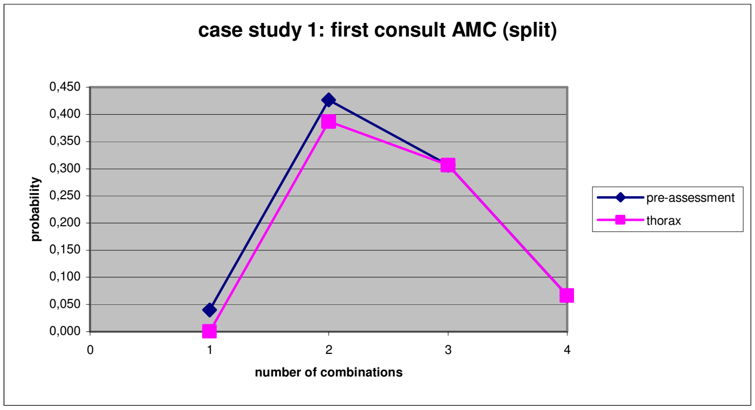


Figure 44: Number of combinations versus conditional probability of activities to fork first consult.

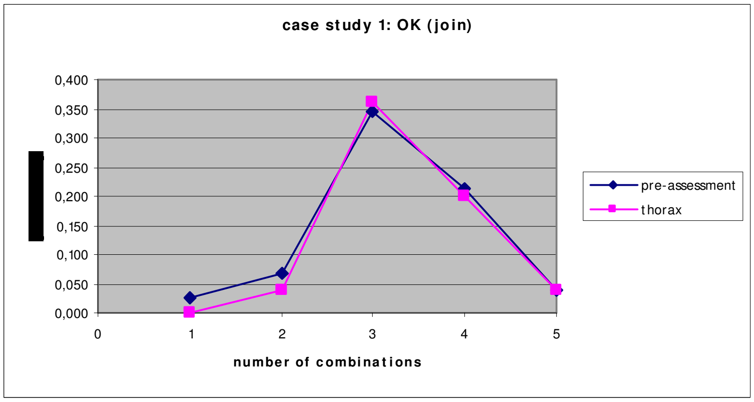


Figure 45: Number of combinations versus conditional probability of activities to fork OK.

The second part of this analysis is about the number of activities versus the average of patterns frequency in relation to the forks of case study.

case study 1	eerste consult AMC			
Task	number of activities		number of patterns frequency	
	split	join	split	Join
Eerste consult	0	4	0	11
Thorax	0	3	0	7
pre-assessment	0	2	0	3
MRI	0	2	0	4
lich. Onderzoek onder narcose	2	0	4	0
OK	6	0	16	0
CT	2	0	2	0
radiotherapie	2	2	2	4
pathologie AMC	0	0	0	0
Echo	0	0	0	0
ECG	0	0	0	0
chemo	0	0	0	0

Table 31: General overview of the forks in the process model.

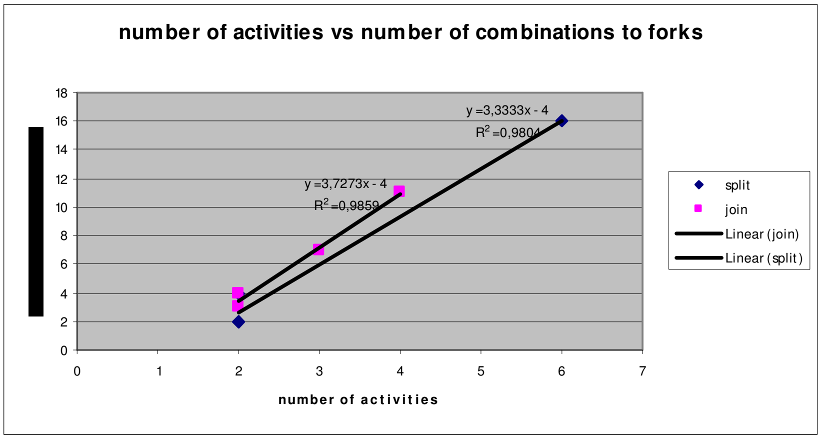


Figure 46: The number of activities versus number of combinations to all forks.

Another point of view is analyzing of the forks, their number of connections versus the average number of combinations.

split	average combinations	max.	ratio	join	average comb.	max.	ratio
2	2.667	4	0.667	2	3.667	4	0.917
6	16.000	64	0.250	3	7.000	8	0.875
				4	11.000	16	0.688

Table 32: Overview of the forks in the process model average based.

The ratio is based of the dividing of the average combinations and the maximum number of combinations to a fork. The maximum number of combination is two power of the total number of direct connected activities to a fork.

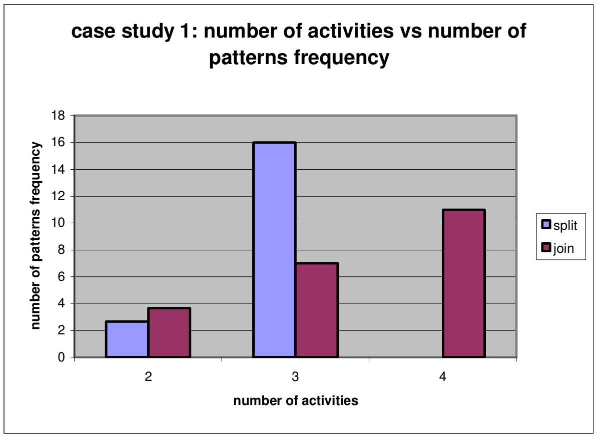
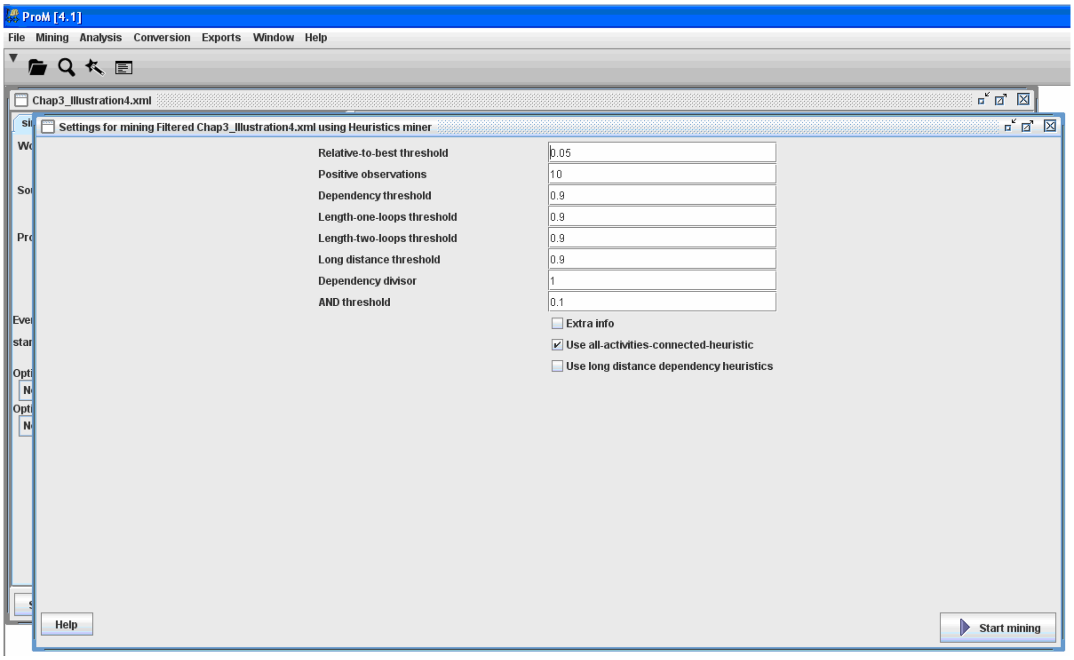


Figure 47: Number of activities vs number of combinations average based.

A.17 The Parameters in Heuristics Miner

This section is about the parameters in the HM. Figure 48 illustrates the start screen of HM with the default parameters. In this screen can the parameters of HM be changed for finding of better DG for the given event log.



Settings for mining Filtered Chap3_Illustration4.xml using Heuristics miner

Relative-to-best threshold	0.05
Positive observations	10
Dependency threshold	0.9
Length-one-loops threshold	0.9
Length-two-loops threshold	0.9
Long distance threshold	0.9
Dependency divisor	1
AND threshold	0.1

□ Extra info
☑ Use all-activities-connected-heuristic
□ Use long distance dependency heuristics

Figure 48: The start screen of the HM and its parameters.

The HM algorithm has the following parameters option: Dependency threshold, dependency divisor, AND threshold, positive observations, Relative-to-best threshold, length-one-loops threshold, length-two-loops threshold, long distance threshold, long distance dependency heuristics and extra information.

Relative-to-best threshold
This parameter indicates that we will accept a dependency measure for which the difference with the "best" dependency measure is lower than the value of relative-to-best threshold. A high value of relative-to-best threshold shall generate detailed behaviour as then the model would as include dependency relations with low dependency values. The default value of this parameter is 0.05.

Positive observations
This parameter enforces that the HM accepts only those dependency relations between activities whose frequency is higher than the value of the positive observations threshold. It helps to filter out low frequent patterns in the log and enables to focus on the main behaviour of the log. A high value assigned to this parameter indicates a user's interest in high reliability of the fact that an activity is directly followed by another activity. The default value of this parameter is 10.

Dependency threshold

This parameter represents a measure which is an indicator of how sure we are of a dependency relation. Using the dependency threshold means that we will accept all those dependency relations from the event log whose value of dependency measure is equal to or greater than the value of the dependency threshold. The default value of the dependency threshold is 0.9. If the option for selecting the *all-events-connected heuristic* is selected then it overrides the use of this parameter.

Length-one-loops threshold

The formula of the length-one-loop represents that we will accept dependency relations between activities in L1L that has a dependency value higher than or equal to the value of L1L threshold. A lower value to this parameter discovers low frequent length-one loops. The default value of this parameter is 0.9.

Length-one-loop formula: $A \Rightarrow_w A = \left(\frac{|A >_w A|}{|A >_w A| + 1} \right)$

Length-two-loops threshold

The formula of the length-two-loop represents that we will accept dependency relations between activities in L2L that has a dependency value higher than or equal to the value of L2L threshold. A lower value to this parameter discovers low frequent length-two loops. The default value of this parameter is 0.9.

Length-two-loop formula: $A \Rightarrow_{2w} B = \left(\frac{|A >>_w B|}{|A >>_w B| + 1} \right)$

Long distance threshold

The value of this parameter specifies which long distance dependencies to accept/reject. If the value of the long distance dependency measure is less than the value of the long distance threshold, the dependency wil be rejected. Otherwise, it will accept. The default value of this threshold is 0.9.

Long distance dependency measure: $A \Rightarrow_{2w} B = \left(\frac{|A >>>_w B| - |B >>>_w A|}{|A >>>_w B| + |B >>>_w A| + 1} \right)$

Dependency divisor

It is the denominator of the formula of the dependency measure, which is default 1. This value is 1 because it is a small number that can affect logs containing few traces in a significant way and the same it has a less significant effect on big logs.

Dependency formula: $A \Rightarrow_w B = \left(\frac{|A >_w B| - |B >_w A|}{|A >_w B| + |B >_w A| + 1} \right)$

AND threshold

This parameter refers to discover the AND/XOR-splits cq. –joins. It indicates that two activities in a log are parallel if their calculated AND-measure (dependency value) is greater than the specified value for the AND-threshold. The default value of the AND-threshold is 0.1.

Extra information

HM generates some additional mining information in the description panel. This information helps to understand how and why a particular output is generated.

Use-all-events-connected heuristic

If this parameter is selected, then a DG is obtained on the basis of the fact that each non-initial activity must have at least one other activity that is its reason for execution and non-final activity must have at least one activity that depends on it for its execution. This heuristic is used during the generation of a DG based on the dependency values. When this parameter is used it ignores all other parameters.

Long distance threshold dependency heuristics

Sone choices are controlled in some other part of the process model, far from weher actually the effect of the choice is realized. These types of process models contain some non-free choice construct. This non-local behaviour is captured by the long distance dependency heuristic. This parameter indicates to the HM that we are also interested in those dependencies which are not only indirect but long distance in nature.

A.18 Fuzzy Miner Results

This section shows the results of example 2 in section 1.4.2. (page 12-13) according the Fuzzy Miner algorithm. It is new plug-in, which can be applied for less structured processes. It is used on the second example in section 1.4.2. The following figures are results in FM using the default options.

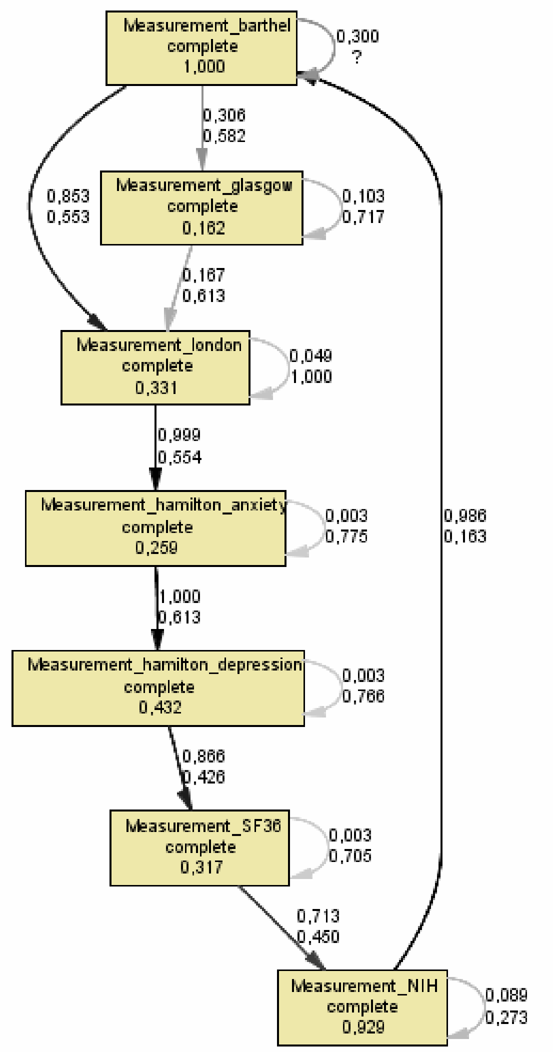


Figure 49: FZ graph default results of case study 2.

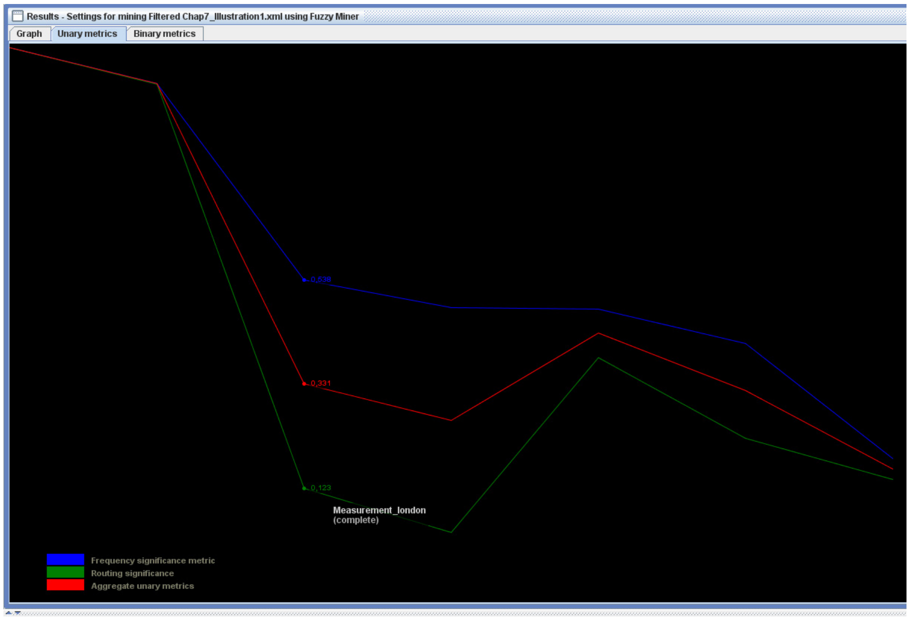
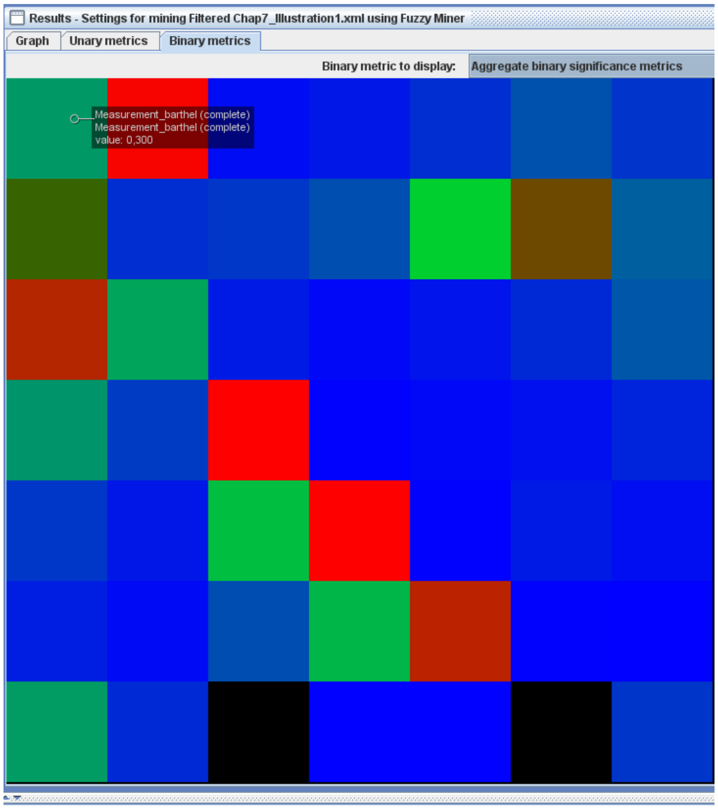

Results - Settings for mining Filtered Chap7_Illustration1.xml using Fuzzy Miner
Graph
Unary metrics
Binary metrics
0,538
0,331
0,123
Measurement_london (complete)
🟦 Frequency significance metric
🟩 Routing significance
🟥 Aggregate unary metrics

Figure 50: FZ unary metrics results of case study 2.


Results - Settings for mining Filtered Chap7_Illustration1.xml using Fuzzy Miner
Graph
Unary metrics
Binary metrics

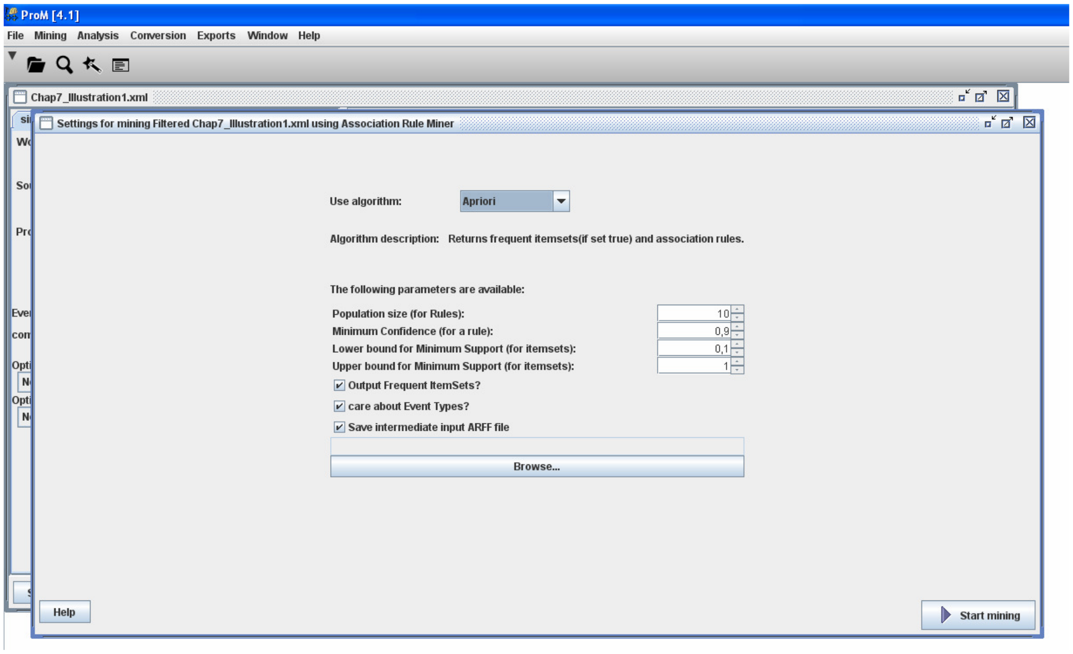
Binary metric to display: Aggregate binary significance metrics

Measurement_barthel (complete)
Measurement_barthel (complete)
value: 0,300

Figure 51: The FZ binary metrics of case study 2.

A.19 Association Rules Miner Results

This section shows the results of example 2 in section 1.4.2. (page 12-13) according the Association Rules Miner algorithm. It is new plug-in, which can be applied for less structured processes. It is used on the second example in section 1.4.2. We used the following options in the start menu of ARM:



Settings for mining Filtered Chap7_Illustration1.xml using Association Rule Miner

Use algorithm: Apriori

Algorithm description: Returns frequent itemsets(if set true) and association rules.

The following parameters are available:

Population size (for Rules):	10
Minimum Confidence (for a rule):	0,9
Lower bound for Minimum Support (for itemsets):	0,1
Upper bound for Minimum Support (for itemsets):	1

☑ Output Frequent ItemSets?

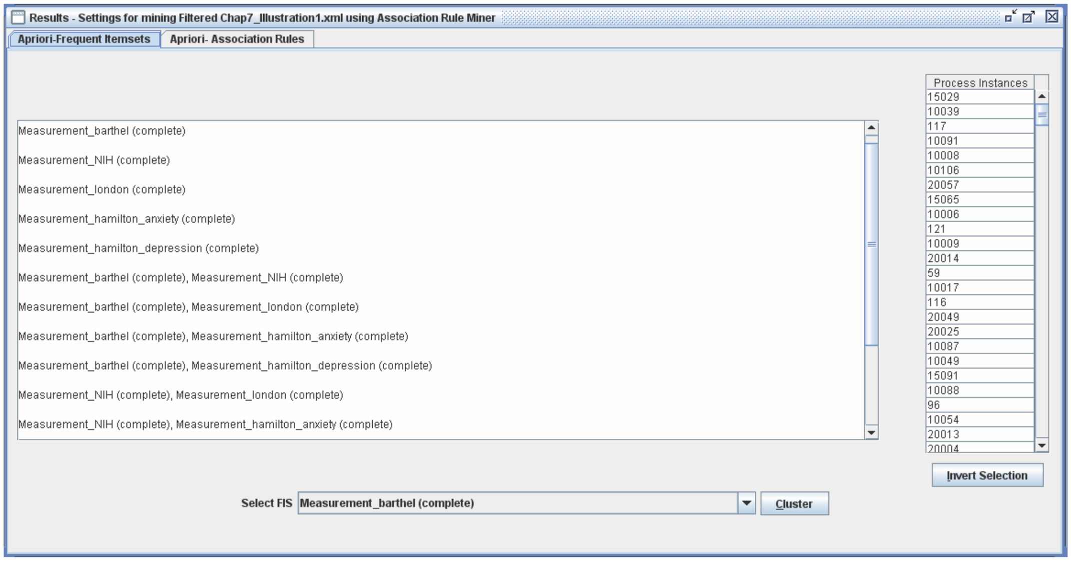
☑ care about Event Types?

☑ Save intermediate input ARFF file

Browse...

Figure 52: The start screen of ARM.

The following figures are results in ARM.



Results - Settings for mining Filtered Chap7_Illustration1.xml using Association Rule Miner

Measurement_barthel (complete)

Measurement_NIH (complete)

Measurement_london (complete)

Measurement_hamilton_anxiety (complete)

Measurement_hamilton_depression (complete)

Measurement_barthel (complete), Measurement_NIH (complete)

Measurement_barthel (complete), Measurement_london (complete)

Measurement_barthel (complete), Measurement_hamilton_anxiety (complete)

Measurement_barthel (complete), Measurement_hamilton_depression (complete)

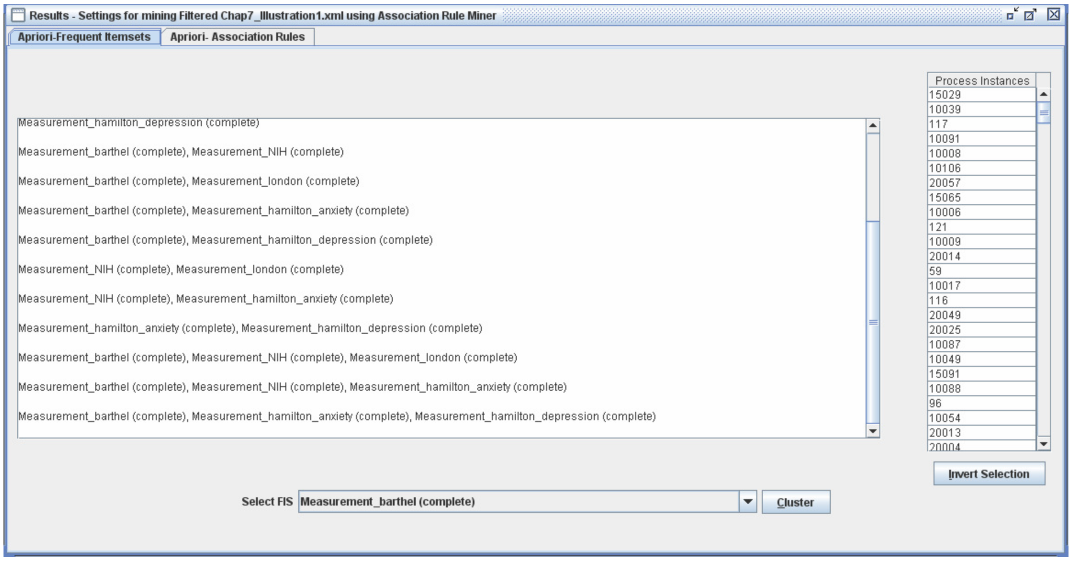
Measurement_NIH (complete), Measurement_london (complete)

Measurement_NIH (complete), Measurement_hamilton_anxiety (complete)

Process Instances
15029
10039
117
10091
10008
10106
20057
15065
10006
121
10009
20014
59
10017
116
20049
20025
10087
10049
15091
10088
96
10054
20013
20004

Select FIS Measurement_barthel (complete)

Figure 53: The Apriori frequent itemsets results of case study (part I).



Results - Settings for mining Filtered Chap7_Illustration1.xml using Association Rule Miner

Measurement_hamilton_depression (complete)

Measurement_barthel (complete), Measurement_NIH (complete)

Measurement_barthel (complete), Measurement_london (complete)

Measurement_barthel (complete), Measurement_hamilton_anxiety (complete)

Measurement_barthel (complete), Measurement_hamilton_depression (complete)

Measurement_NIH (complete), Measurement_london (complete)

Measurement_NIH (complete), Measurement_hamilton_anxiety (complete)

Measurement_hamilton_anxiety (complete), Measurement_hamilton_depression (complete)

Measurement_barthel (complete), Measurement_NIH (complete), Measurement_london (complete)

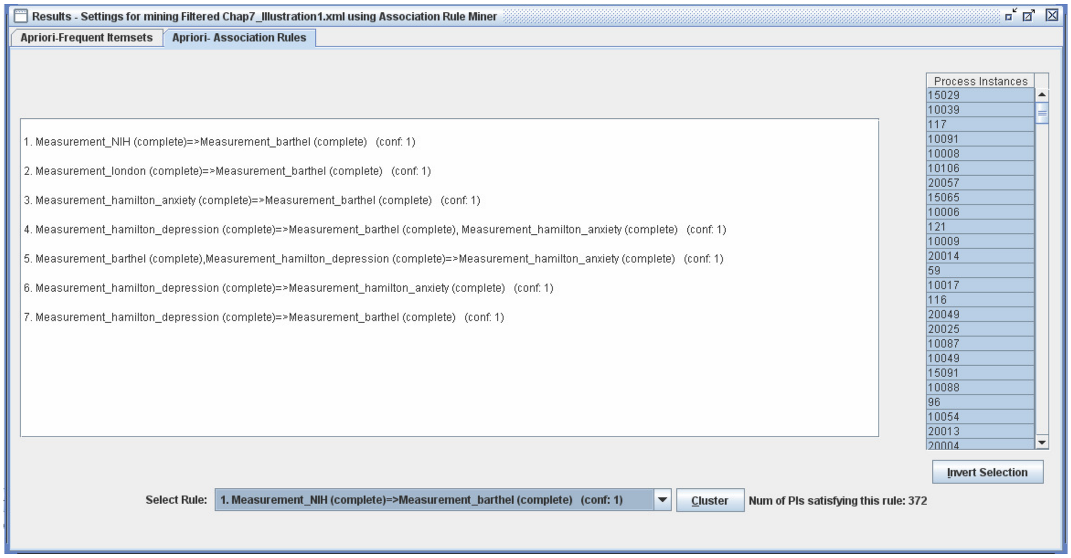
Measurement_barthel (complete), Measurement_NIH (complete), Measurement_hamilton_anxiety (complete)

Measurement_barthel (complete), Measurement_hamilton_anxiety (complete), Measurement_hamilton_depression (complete)

Process Instances
15029
10039
117
10091
10008
10106
20057
15065
10006
121
10009
20014
59
10017
116
20049
20025
10087
10049
15091
10088
96
10054
20013
20004

Select FIS: Measurement_barthel (complete)

Figure 54: The Apriori frequent itemsets results of case study 2(part II).



Results - Settings for mining Filtered Chap7_Illustration1.xml using Association Rule Miner

1. Measurement_NIH (complete)=>Measurement_barthel (complete)   (conf: 1)

2. Measurement_london (complete)=>Measurement_barthel (complete)   (conf: 1)

3. Measurement_hamilton_anxiety (complete)=>Measurement_barthel (complete)   (conf: 1)

4. Measurement_hamilton_depression (complete)=>Measurement_barthel (complete), Measurement_hamilton_anxiety (complete)   (conf: 1)

5. Measurement_barthel (complete),Measurement_hamilton_depression (complete)=>Measurement_hamilton_anxiety (complete)   (conf: 1)

6. Measurement_hamilton_depression (complete)=>Measurement_hamilton_anxiety (complete)   (conf: 1)

7. Measurement_hamilton_depression (complete)=>Measurement_barthel (complete)   (conf: 1)

Process Instances
15029
10039
117
10091
10008
10106
20057
15065
10006
121
10009
20014
59
10017
116
20049
20025
10087
10049
15091
10088
96
10054
20013
20004

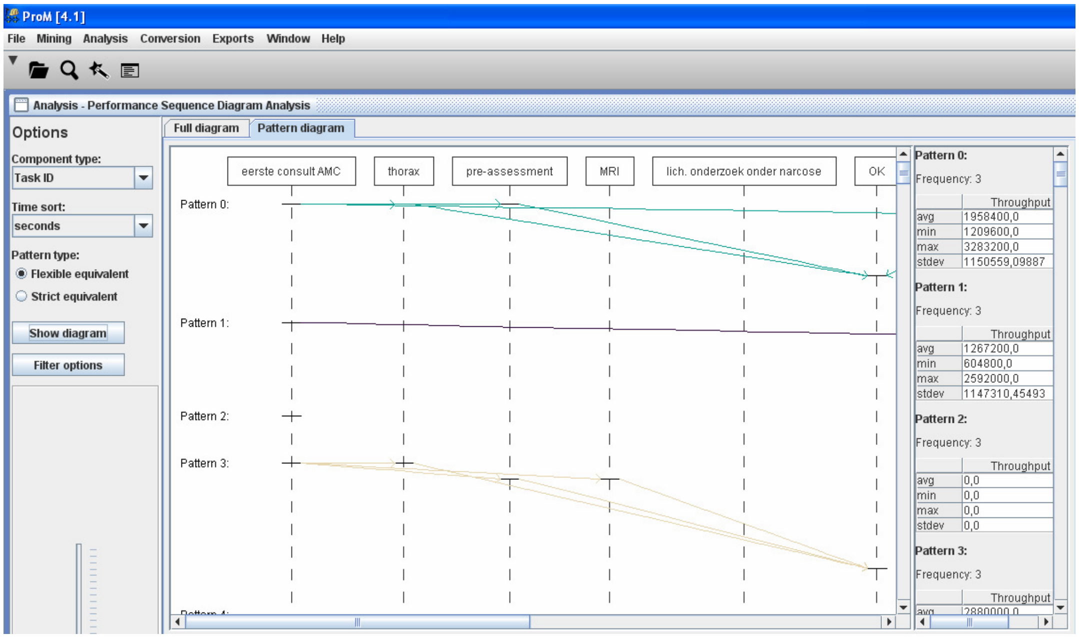
Select Rule: 1. Measurement_NIH (complete)=>Measurement_barthel (complete)   (conf: 1)

Num of PIs satisfying this rule: 372

Figure 55: The Apriori Association rules results of case study 2.

A.20 Performance Sequence Diagram Analysis

This section illustrates the Performance Sequence Diagram Analysis (PSDA). It is an option within the HM. It shows the performance time between the related activities, and gives also some statistical information about average, minimum, maximum and standard deviation of the throughput time. Below illustrates a screenshot of the PSDA of the first consult at AMC.



Options

Component type: Task ID

Time sort: seconds

Pattern type:
● Flexible equivalent
○ Strict equivalent

Pattern 0:

Frequency: 3

	Throughput
avg	1958400,0
min	1209600,0
max	3283200,0
stdev	1150559,09887

Pattern 1:

Frequency: 3

	Throughput
avg	1267200,0
min	604800,0
max	2592000,0
stdev	1147310,45493

Pattern 2:

Frequency: 3

	Throughput
avg	0,0
min	0,0
max	0,0
stdev	0,0

Pattern 3:

Frequency: 3

	Throughput
avg	2880000,0

Figure 56: PSDA of case study 1.