

MASTER

Process mining in PeopleSoft

Ramesh, Adarsh

Award date:
2006

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

Process Mining in PeopleSoft

A. Ramesh

SUPERVISORS:

DR. M.H. JANSEN-VULLER (TM-IS)

DR. P.A.P. THIRAN (W&I-AIS)

PROF. DR. IR. W.M.P. VAN DER AALST (TM-IS)

Eindhoven, January 2005

Abstract

This Master's Thesis describes the efforts made contributing to the Process Mining research of the Eindhoven University of Technology. Process Mining is a methodology to distil a process definition from a set of real process executions, using enactment logs from process-aware information systems. Process Mining has successfully been applied to some of these systems (Staffware, FLOWer). However, attempts to apply this technology to the SAP R/3 ERP system, have been less successful. Hence, to test the application of Process Mining to other ERP systems, this research focuses on Process Mining in PeopleSoft. The efforts made to investigate the Process Mining possibilities for one PeopleSoft system, PeopleSoft Financials, are described. Furthermore, the problems encountered when applying Process Mining are explained.

Summary

Problem definition

The awareness of Workflow Management technology in the field of Information Systems (IS) is greater than it was ten years ago. Several Workflow Management Systems (WfMSs) now exist that offer generic modeling and enactment capabilities for structured business processes. Other systems besides WfMSs have adopted workflow technology.

There are many problems when applying workflow technology to WfMSs. Most common problem is that the workflow model does not reflect the actual work. Managers have no (or little) overview into their business processes and thereby, are not able to improve or manage their processes. From this necessity, in [3], the authors suggest a methodology (Process Mining) to discover a process description based on a running Workflow management system implementation. For this methodology, we need to obtain transactional data that is registered in the system. Process Mining requires specific information from any process-driven system to reconstruct a process model. These requirements are [7]:

- Case ID
- Task ID
- Tasks are logically ordered

If we are able to find this information for executed processes within a system, we have to extract and convert to a format suitable for the Process Mining tool. The format to store process logs, i.e. logs that contain events (which is the combination of Case ID, Task ID, and that the tasks are logically ordered), is defined in Appendix A. By converting the logs from the system we are mining to this process log format, we can use the Process Mining tool (ProM) to generate actual process models.

For WfMSs (Staffware, and FLOWer), Process Mining has been successfully applied, but for the ERP system (SAP R/3) Process Mining could not be applied in a highly generic and automated fashion. Therefore, the goal of this research is to explore the possibilities of applying Process Mining to other ERP system, namely, PeopleSoft Financials. The research question can be formulated as follows:

“Can we create an adapter to convert the logs of the PeopleSoft Financials ERP system to the common log format defined by the Process Mining tool?”

In other words, the aim of this research is to identify the Case ID, Task ID in the PeopleSoft Financials process log(s), then extract this information, convert it into a

common log format by building a suitable adapter and then generate models using the ProM tool.

Attempted approaches

The general approach taken to apply Process Mining to PeopleSoft Financials is

1. finding the Process Mining requirements;
2. extracting this information;
3. building an adapter that converts the extracted information to the common format for storing process logs;
4. generating process models of the converted logs using the ProM tool.

We only progress to the second step, provided that we successfully have finished the first step. From previous mining efforts we know that the first step is the most important step.

We look for the Case ID, Task ID of executed processes systematically in the system architecture of PeopleSoft Financials. PeopleSoft Financials is split into four main areas, namely, (a) web client, (b) web server, (c) application server, and (d) database.

The approaches attempted, to find the Process Mining information, are categorized into three sets of approaches.

The first set looked at the application server (c). The application server is where all applications within PeopleSoft Financials are executed. For the approaches regarding the application of Process Mining to PeopleSoft Financials, we were not able to find which application are executed for a process.

Therefore, in the next set of approaches, we decided to look at the database (d) of PeopleSoft Financials. Using techniques such as database foreign keys and SQL triggers we tried to obtain information, regarding processes, purely from the database. However, in the database of PeopleSoft Financials, we cannot find process flow information, we only find data. We conclude from this approach that processes are not controlled by the database.

The final set of approaches covers the web client (a) and web server (b) part of the system architecture. We notice that users work with PeopleSoft Financials using their web browsers, and the applications they work with are called pages and components. These components can be classified as business process tasks (Task ID). However, we cannot find the Case ID for which a component has been executed. Therefore, we are not able to continue with this approach. Since, we have looked at all areas of PeopleSoft Financials, we draw our conclusions.

Conclusions

For the application of Process Mining we need the information about the executed processes and the order in which activities for a process are executed. The efforts taken to answer the questions lead us to many different types of Process Mining

attempts. Unfortunately, we were not able to find the process mining requirements in all the approaches.

We conclude that PeopleSoft Financials does not enforce workflow through process models. For any process log of PeopleSoft Financials that describes the behaviour of the system, it is complex to find the common data element, which we call Case ID. From our efforts, and the generalisation we make to other similar PeopleSoft systems, it is possible to conclude that there is no such information available. Therefore, we are able to conclude that we are not able to create an adapter that converts the process logs of PeopleSoft Financials to the common format for storing Process Mining logs.

We recommend that for the application to other systems, the same strategy should be used. Furthermore, for the Process Mining research, it may be wise to focus on Process Mining techniques that do not need Case IDs to mine processes.

Preface

The Master's final research assignment was conducted at the Faculty of Technology Management, department of Information Systems and this Master's thesis is the result of many months of research.

Now the moment has come that the student life that I have enjoyed for more than seven years must come to an end. I, as a Bachelor in Computer Science from Fontys, am technology minded and the decision to do my Master's at the Eindhoven University of Technology was based on this fact. I wanted to be able to think more abstractly, and be less focused about the actual details of technology. It has not been easy to adapt my way of thinking, but I am confident now that I possess both ways of thinking and this makes me more versatile.

This research assignment has certainly been the most challenging assignment I have ever attempted. At the beginning of this research assignment I did not have any experience with Workflow Management, ERP systems and Process Mining. Therefore, during the research there have been occasions where I constantly have asked myself if I am a fish out of water, but I felt that, as an academic, one should be able to handle all kinds of environments. This is what kept me going.

I would like to take this opportunity to thank a number of people that have supported me during the period of my research assignment. First of all, I would like to thank Monique Jansen-Vullers who, not only was my supervisor, but also acted as a teacher during the research. Specially coming from a very technical background, she helped me in lifting my work to an academic level with her sharp remarks and feedback. I would also like to thank my other supervisors Wil van der Aalst, and Philippe Thiran for their advice.

Furthermore, I would also like to thank people from the Fontys: Frans Mouws, Frans Rouwhorst and especially the experts at the PeopleSoft Competence Center (PCC) for their support, because without them I would have never come so far with PeopleSoft. They taught me so much about the PeopleSoft ERP system in such a short period of time and they were always there to answer my questions.

Last but not least, I would also like to thank my room mate Ronny Mans for the help and support he has given me in writing, Christian Günther for his ideas regarding experimental implementation work and overall mental support and last but not least, my parents who have always been there for me in more ways than I can ever express.

Adarsh Ramesh

November, 2005

Contents

1	Research assignment	7
1.1	Introduction	7
1.2	Background	7
1.3	Concepts of Process Mining	8
1.4	Process mining research	10
1.5	Research goal	12
1.6	Approach	12
1.7	Report outline	14
2	Process Mining	15
2.1	Introduction	15
2.2	Process Mining	15
2.3	Process Mining example	16
2.4	Process Mining requirements	18
2.5	Process Mining perspectives	18
2.6	ProM Framework	20
2.7	Available converters	23
2.8	Conclusion	24
3	PeopleSoft ERP	25
3.1	Introduction	25
3.2	History	25
3.3	Process Aware Information System	26
3.4	WfMS vs ERP	26
3.5	PeopleSoft ERP	28
3.6	PeopleSoft Financials	28
3.7	Conclusion	29
4	Overview of approaches	31
4.1	Introduction	31
4.2	Application mining approaches	32
4.2.1	Application Engine mining	33
4.2.2	Process scheduler	37
4.2.3	Application mining results	38
4.3	Database mining approaches	38
4.3.1	Database foreign keys	38
4.3.2	SQL triggers	39

4.3.3	Table mining	40
4.3.4	Database mining results	40
4.4	User oriented approach	41
4.4.1	PeopleSoft pages	41
4.4.2	PeopleSoft components	47
4.4.3	User oriented results	48
4.5	Conclusion	48
5	Generic vs. Specific	50
5.1	Introduction	50
5.2	Generic	50
5.2.1	Other versions	50
5.2.2	Other implementations	51
5.2.3	Other systems	51
5.3	Specific	52
5.3.1	PeopleSoft Business maps	52
5.3.2	Customized mining	52
5.4	Conclusions	53
6	Conclusions and Recommendations	54
6.1	Conclusions	54
6.2	Recommendations	55
6.2.1	Process mining research	55
6.2.2	Mining PeopleSoft CRM system	55
6.2.3	Fontys	56
A	Common format for process mining logs	60
A.1	Introduction	60
B	Table mining explained	62
C	PeopleSoft Security	66
C.1	Users and Roles	66
D	Process mining explained	69
D.1	Introduction	69
D.2	History of process mining	69
D.3	Workflow mining	70
D.4	Process mining example	72
D.5	Conclusion	73
E	Inventory of logs	74
E.1	Introduction	74
E.2	User trace	74
E.2.1	SQL trace settings	75
E.2.2	PeopleCode trace settings	75
E.2.3	Component Processor trace settings	75
E.3	Application Server trace	77
E.4	Conclusion	77

Chapter 1

Research assignment

1.1 Introduction

The subject of this Master's Thesis is Process Mining and more precisely, visualising processes based on data taken from PeopleSoft Financials ERP system. In this chapter, we introduce the subject of Process Mining and the research for this Master's Thesis. The background for this research is given in Section 1.2. In Section 1.3, the concepts of Process Mining together with other preliminaries are given. We then briefly describe the existing Process Mining research at Eindhoven University of Technology in Section 1.4. The aim of this research, to extend the existing Process Mining research, is given in Section 1.5. Finally, the approach to achieve the research aim is given in Section 1.6 and in Section 1.7 we give the report structure.

1.2 Background

The existence of Workflow Management technology in the field of Information Systems is greater than it was ten years ago. A variety of Workflow Management systems (WfMS) now exist that offer generic modeling and enactment capabilities for structured business processes [3]. These business processes are supported by Workflow Management systems by describing the process of all kinds of cases. Process descriptions are typically graphical process descriptions, and are executable. Other software systems, besides Workflow Management systems, have adopted workflow technology (e.g. Windchill in Product Data Management systems¹ (PDM) and Adobe LiveCycle Workflow in Document management systems² (DMS)).

There are many problems when applying workflow technology to Workflow Management systems. Most common problem is that the workflow model does not reflect the actual work. These models describe only what should be done rather than what is done. Managers have no (or little) overview into their business processes and thereby, are not able to improve or manage their processes. From this necessity, in [3], the authors suggest a methodology to discover a process description based on a running Workflow management system implementation. For this methodology, we need to obtain transactional data that is registered in the system. Transactional data can be log files of the Workflow management systems that has recorded process

¹<http://www.ptc.com/appserver/mkt/products/home.jsp?k=352>

²<http://www.adobe.com/products/server/workflowserver/main.html>

activities. This methodology, called workflow mining, is then used as input for the (re)design of the system.

We aim to extend the workflow mining methodology to other kinds of information systems. Process mining extends workflow mining by looking at systems not only driven by workflow models but looking also to systems driven by process models. A process model can be a visual description of a process in isolation. A system driven by process models is called a Process Aware Information System (PAIS)[3].

Process models define the order of tasks that have to be executed for one process instance, e.g. an insurance claim. A task is a logical unit of work [4]. The order in which tasks of a claim are processed is predetermined by the process model, similar to a Workflow model. A process instance is the collection of executed tasks. For example, the process "File Insurance Claim" can have many process instances.

In order to use Process Mining for any process redesign, we need to obtain information regarding process instances.

Another category of PAISs, other than Workflow Management systems, are the Enterprise Resource Planning (ERP) systems. This research focuses on Process Mining of PeopleSoft's ERP systems. We explain more about ERP and Workflow management systems in the next chapter.

1.3 Concepts of Process Mining

Process mining research covers many concepts, and before we can focus on the goal of this research, we need to understand some of its concepts.

Process and process models

A *process* definition indicates in what order tasks must be performed [4]. These process descriptions can be given in a graphical notation. A graphical process description describing one process is called a *process model*. An example of a process model, represented in Petri net notation [22] is given in Figure 1.1. We do not explain the syntax of Petri nets here.

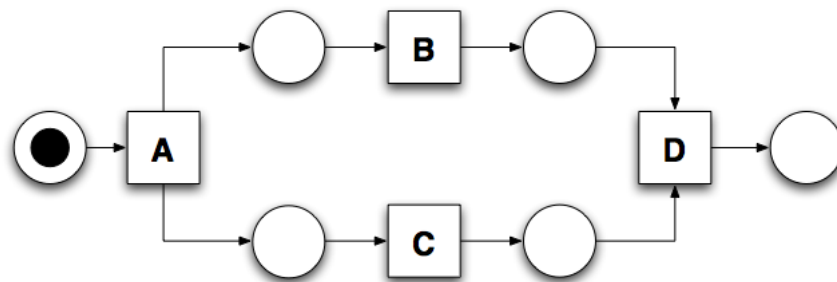


Figure 1.1: A process model example

In the process described in Figure 1.1 we can clearly see the start and the end of the process model. The process model starts with transition *A*, and allows transition *B* and transition *C* to execute in parallel. When both transactions have finished, transition *D* finishes the process.

Process instance

When a process model has been designed, it can be executed. Each execution of such a process is called a *process instance*, also known as case. Examples of process instances are the handling of an insurance claim, the delivery of a customer order or even the steps taken to handle a complaint registered by a call center.

Process Aware Information System

For this section, it is important to understand that there are information systems that may use process models to configure their information systems. These systems are more commonly classified as *Process Aware Information Systems* (PAIS). However, the level in which the process models are used to configure the information, differs per information system. Workflow management systems such as Staffware, COSA and FLOWer explicitly require a process model to define and configure their processes, while in ERP systems such as SAP R/3, PeopleSoft and Oracle a process model is less explicitly required. For some ERP systems no explicit workflow is enforced and in some systems the process model is hidden in the pre- and post-conditions of applications.

Process logs

Process logs or event logs are an essential concept of Process Mining. We assume that for an information system we can register which tasks have been executed and for which process. These process executions are what we know as process instances with information about events. An event is the combination of different information elements, such as the activity that has been executed, the case for which it has been executed and the time of the executed activity. We use this information for the mining of processes.

Process logs are usually stored as text files or as tables in a database. The idea of these recorded logs is that they register the behaviour of the PAIS. An example of a PAIS log is given in Figure 1.1.

Table 1.1: An example of a PAIS log

Case ID	Task ID	User	Timestamp
3	A	Steve	2005-10-02 09:30
2	B	Doug	2005-10-02 09:30
2	C	Doug	2005-10-02 09:32
1	D	Emily	2005-10-02 09:32
4	A	Steve	2005-10-02 09:34
3	C	Steve	2005-10-02 09:34
3	B	Doug	2005-10-02 09:36
5	A	Steve	2005-10-02 09:36
2	D	Emily	2005-10-02 09:38
3	D	Emily	2005-10-02 09:38
4	B	Steve	2005-10-02 09:40
5	C	Doug	2005-10-02 09:40
4	C	Doug	2005-10-02 09:42
5	B	Steve	2005-10-02 09:42
5	D	Emily	2005-10-02 09:44
4	D	Emily	2005-10-02 09:44
6	A	Doug	2005-10-02 09:46

Process mining

Process mining uses the information available in the process logs to reconstruct the order of activities in the form of a graphical model (i.e. process model). The model represents the executed processes based on the logs. The goal of Process Mining is to compare the actual behaviour of the PAIS with the configured model, based on the executed process instances. It is quite possible to have incorrect process models obtained from the system. Consider for example, the process model in Figure 1.1. When the process logs of the system only have recorded the tasks A, B, and D in this particular order, then, using Process Mining, we will only discover a part of the original process model. This results in a process model with tasks A, B and D in sequence. Therefore, we assume to have a sufficiently large set of logs. We discuss the concepts of Process Mining in more detail in Chapter 2.

1.4 Process mining research

Process mining research [24], at Eindhoven University of Technology, has provided a set of tools used for applying Process Mining to different information systems driven by process models. In general, the application of Process Mining to an information system consist of four steps. First, find the logs of executed processes in the PAIS and secondly, extract them into a convertible format. Third, we build an adapter to convert these PAIS logs into the correct format. Finally, use the existing Process Mining tools to generate process models of these converted logs. The Process Mining tool requires a common format for logs as input, as defined in [16].

The research group at Eindhoven University of Technology has applied Process Mining to the Workflow management systems: Staffware[29] and FLOWer[27], and the

ERP system SAP R/3[17].

To explain what the application of Process Mining means, see Figure 1.2. Unfortunately, we can not convert just any random data from the PAIS to the common format. The requirements for the common format for storing log files, that has to be extracted from the PAIS logs, are [7]:

- Case ID (which process instance is it?)
- Task ID (which activity has been executed?)
- Tasks are ordered

The first two data elements together, form one event of a process log. These elements are mandatory for mining. Furthermore, if the order in which these Case IDs, Task IDs occur in a log is ordered, then a Timestamp is not required. In Table 1.1 the mandatory fields are the first two columns, but if we have the *User* and the *Timestamp* information, we are also able to discover more than only the process model³.

If we are able to find more of these events, then we can extract the information, build an adapter that converts them into the common format for storing logs, and use the Process Mining tools to generate models. Only then, we can say that Process Mining has been applied to the system.

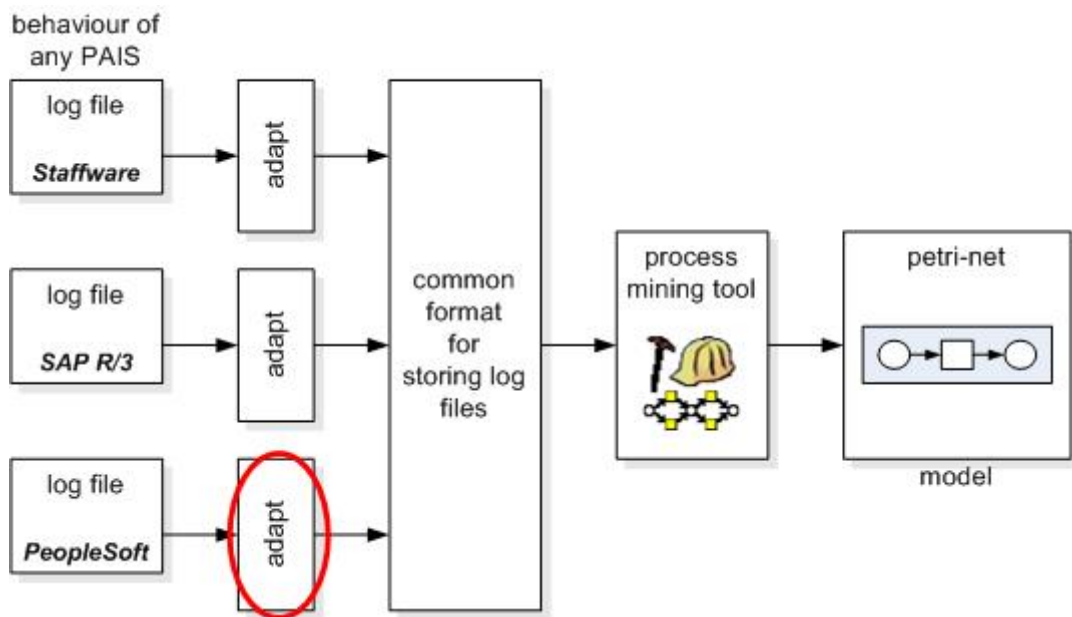


Figure 1.2: The genericity of Process Mining research

The goal of the Process Mining group is to provide support for as many PAISs as possible and thereby developing adapters as generic as possible. This means that once an adapter is built, for a PAIS, it should work for all other versions of this PAIS.

Process mining can not only be used for process discovery, but also for other mining

³In Chapter 2 we discuss other possibilities of Process Mining using these extra columns.

objectives. These are discussed in Section 2.5.

From the Process Mining experiences in Workflow management systems (Staffware and FLOWer), the adapters that were built are generic. An adapter is generic when it is reusable for different versions, implementations of that system with minor changes.

However, the adapter for the SAP R/3 ERP system, is specific and works for only one module and implementation. More details about the current research work is given in Section 2.7.

1.5 Research goal

The objective of this research is to contribute to the existing Process Mining research by adding another source system to the list of systems that can be mined. For the source system, we have chosen PeopleSoft's ERP system⁴. The PeopleSoft ERP system, is a large, well established, ERP system and offers support for many business fields. Due to the timeframe of this research, only one of the PeopleSoft systems is selected to be analysed. Ideally, any adapter built for this system is generic and thus applicable for other PeopleSoft systems.

In order to mine processes in PeopleSoft, one PeopleSoft ERP module is analyzed at a time. For this research, we need a PeopleSoft implementation. Fontys Hogescholen⁵ provided us a PeopleSoft system. This research environment houses the PeopleSoft Financials system. PeopleSoft Financials is selected, because any tests performed on the system will not affect PeopleSoft Financials users, and when a real mining has to be performed, the "live" environment can be used. Depending on the research results, we want to evaluate if the adapter is only restricted to one system. The research question for this research can be formulated as follows:

"Can we create an adapter to convert the logs of the PeopleSoft Financials ERP system to the common log format defined by the Process Mining tool?"

In other words, the aim of this research is to identify the Case ID, and Task ID in the PeopleSoft Financials process log(s), then extract this information, convert it by building a suitable adapter and then generate models using the Process Mining tools.

1.6 Approach

The approach for this research is derived from the aim of this research. It is not possible to answer the research question at once. Therefore, this has to be answered gradually during the research. The research is categorized into four areas: Process mining, PeopleSoft ERP, Adapter for PeopleSoft Financials and General applicability of the adapter. For each of these areas, a short approach is given.

1. Process mining

Before we start, we need to understand what the principles of Process Mining

⁴We discuss PeopleSoft in Chapter 3

⁵Fontys Hogescholen: <http://www.fontys.nl>

are first. Therefore, we have to ask ourselves: what is Process Mining? The idea of Process Mining is certainly not new. Many people have experimented with generating process models from actual events. Furthermore, Process Mining is not limited to process discovery alone, and therefore we discuss the other mining possibilities also. Then, by reading about these experiments and reading articles written about workflow mining, which is the mining of workflow models in Workflow Management Systems, we have a good place to start. It is important to know what is required from an information system for Process Mining to be applicable and to which systems it has already been applied to.

2. PeopleSoft ERP

Before we start looking at PeopleSoft Financials, it is logical to first start understanding how Enterprise Resource Planning systems work and then how they have evolved over the years. We focus our knowledge gathering about ERP systems to only what we require regarding PeopleSoft and this research. What first started out as small applications connected to each other to support production processes, now has grown to become one of the largest information systems packages in the world. Therefore, we must also know whether this large information system is “process aware”.

Furthermore, many ERP systems offer workflow management technology support. The support is however limited to certain modules and is not comparable to the level of support in WfMSs. So how does the ERP system differ from the WfM system when looking at the applicability of Process Mining and more importantly, how does PeopleSoft Financials work with processes? We try to answer these questions, to help us find the Process Mining information we are looking for.

3. Adapter for PeopleSoft Financials

Our goal is to find the common format requirements in PeopleSoft Financials logs. Once we know where to find this data in PeopleSoft Financials, it is a matter of extracting that data and then building an adapter that converts the data to the correct format.

ERP systems have very complex data structures and finding the data is a time-consuming task. Because of this time exhaustive task and time restrictions of the research, there are no guarantees for positive results. We therefore attempt to find the information in PeopleSoft Financials in a systematic way. Nonetheless, if the information is available, the design of the adapter should preferably be not only applicable to PeopleSoft Financials, but also to other systems/modules of PeopleSoft.

4. General applicability of the adapter

The generalisation phase is to use the knowledge gained from this research to evaluate the PeopleSoft Financials adapter for other versions, other implementations, other systems of PeopleSoft and also other PAISs in general.

The approach given is based on the information known about the assignment at the beginning of the research. Although the approach seems straightforward, finding the information in PeopleSoft Financials will be the most challenging. However, with the knowledge and support of others in the Process Mining group at the Eindhoven University of Technology, it should be possible to answer the research question.

1.7 Report outline

The remainder of this report is structured as follows. Chapter 2 discusses Process Mining. In this chapter we also explain about the different perspectives possible with Process Mining. In Chapter 3 we start by explaining what ERP systems are and how they have changed over the years. In addition, we explain how the PeopleSoft Financials ERP system works. The attempted approaches to identify the Process Mining requirements in PeopleSoft Financials are described in Chapter 4. The result of the mining approaches is evaluated and we give a generic answer and specific answer to our research goal in Chapter 5. This thesis ends by giving our conclusions and recommendations in Chapter 6.

Chapter 2

Process Mining

2.1 Introduction

The concept of Process Mining has already been briefly discussed in Section 1.3. In this chapter we explain the term Process Mining in Section 2.2. An example of Process Mining is given in Section 2.3. This example shows the requirements for Process Mining and these are listed in Section 2.4. Process Mining is not limited to discovering processes and in Section 2.5 other perspectives are discussed. These perspectives are supported in the Process Mining Framework (ProM), developed at the Eindhoven University of Technology. In Section 2.6, this framework is explained. Using this framework, several successful Process Mining efforts are explained in Section 2.7. Finally, in Section 2.8 we close this chapter with the conclusions.

2.2 Process Mining

The foundation upon which Process Mining is based on, is Business Process Management (BPM). Although often perceived otherwise, BPM is a very generic way to manage business processes[2]. BPM itself is not system bounded. However, when a BPM system (BPMS) is involved to support business processes, it becomes system bounded. The BPMS is the actual software system that is used to implement BPM for a specific organization. Examples of BPMSs are a Workflow Management System (WfMS), but also a Enterprise Resource Planning (ERP) system.

A general characteristic of BPMSs is that they are *process aware* and generic in the sense that it possible to modify the processes it supports.

Van der Aalst, for example, divides the BPM approach into four different phases, namely design, configuration, enactment, and diagnosis [3]. Section D.3, explains these phases in a WfMS setting in more detail.

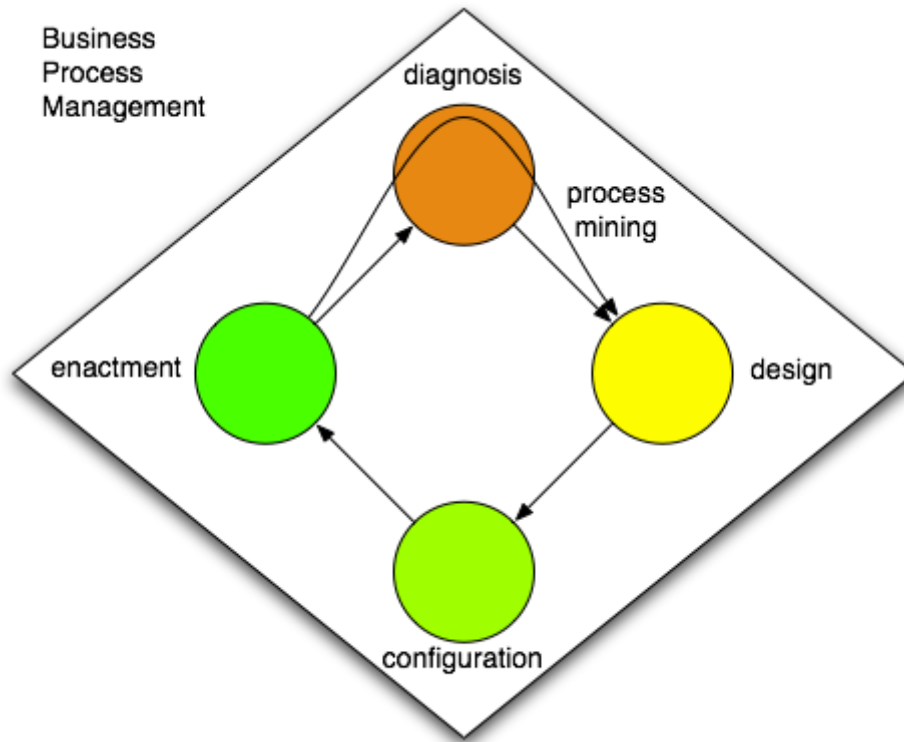


Figure 2.1: Business Process Management phases.

In Figure 2.1, an example of the four BPM phases are given. In the *design phase*, business processes that need to be supported by the BPMS are defined. These are then configured in the *configuration phase*. Remember that configuring BPMSs is implementation specific. For the configuration of, e.g. ERP systems, off-the-shelf applications are customized to support the business process at hand. In the *enactment phase* the configured BPMS is used to execute business processes. Unfortunately, many systems do not ensure that processes are executed in uniform. Therefore, the *diagnosis phase* is typically used to detect any deviations between process design and the execution of processes. In many cases, the diagnosis phase is skipped, because of the extra strain which is put on the BPMS to collect this information. Furthermore, the diagnosis phase is also used to gather information about all process executions (in the enactment and diagnosis phase) to aid the (re-)design of the process.

One form of diagnosis is Process Mining. In [7], the authors define the term *Process Mining* as follows, *a method of distilling a structured process description from a set of real executions*. The history of using executed process executions to discover process descriptions is given in Section D.2 of the Appendix. Workflow mining, the application of Process Mining to WfMSs, was first introduced in [30]. This initiated the application of Process Mining to other PAISs.

2.3 Process Mining example

To illustrate the principles of Process Mining, consider the log shown in Table 2.1.

Table 2.1: A Process Mining example

LOG LINE	Case ID	Task ID	User	Timestamp
001	3	A	Steve	2005-10-02 09:30
002	2	B	Doug	2005-10-02 09:30
003	2	C	Doug	2005-10-02 09:32
004	1	D	Emily	2005-10-02 09:32
005	4	A	Steve	2005-10-02 09:34
006	3	C	Steve	2005-10-02 09:34
007	3	B	Doug	2005-10-02 09:36
008	5	A	Steve	2005-10-02 09:36
009	2	D	Emily	2005-10-02 09:38
010	3	D	Emily	2005-10-02 09:38
011	4	B	Steve	2005-10-02 09:40
012	5	C	Doug	2005-10-02 09:40
013	4	C	Doug	2005-10-02 09:42
014	5	B	Steve	2005-10-02 09:42
015	5	D	Emily	2005-10-02 09:44
016	4	D	Emily	2005-10-02 09:44
017	6	A	Doug	2005-10-02 09:46

Assume that Table 2.1 is a table taken from a BPMS that has recorded the process executions. If we also assume that this log is complete¹, then we can use information to reconstruct the process model. Further details about this assumption is given in [3, 8].

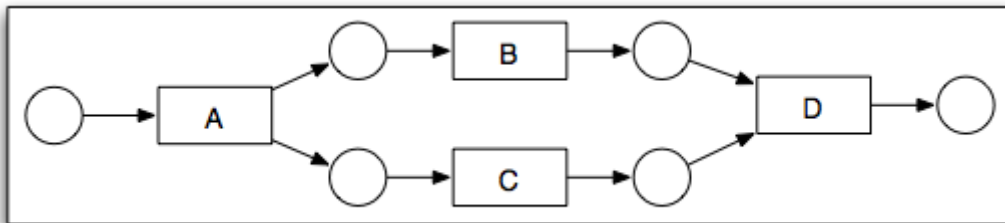


Figure 2.2: Discovered process model

The process model we obtain when we reconstruct the process logs, by looking at the Case ID and the Task ID, is given in Figure 2.2. Note that the Figure displays a visual model of the process in Petri-net language[1, 22]. From the log we can see that, for every complete case, the tasks *A*, *B*, *C* and *D* are executed. Furthermore, we see that the task *A*, for every case, is always executed first and task *D*, always executed as last. Therefore, they are the start and finish of this process. Finally, we see that task *B* and *C* are executed in any arbitrary order, we therefore may conclude that they are parallel tasks. Proof of the correctness of Process Mining is given in [30]. Also note that the cases 1, and 2 are not complete and these are skipped.

¹It represents enough process executions to distil an accurate process description

2.4 Process Mining requirements

From the Process Mining example, we understand that the information given in the logs is essential for mining. Therefore, an assumption is made that it is possible to record events such that we have information about the order in which the events of a case are executed [7]. The minimum requirements for the process logs, and indirectly for Process Mining, are:

1. Case ID
The Case ID indicates for which case the activities have been executed. Without the Case ID, we are not possible to identify from executed logs whether new processes are started, or finished.
2. Task ID
The Tasks ID indicates which activity has been executed. When a task is executed for a specific case, we refer to it as an activity [4]. Tasks are “atomic” and cannot be partially executed.
3. Tasks are ordered
The last requirement of Process Mining is that the tasks are ordered. In other words, we know that when an activity in the log appears before another activity in the log, we know that the first activity was executed first. It is also enough to have a totally ordered log file containing case and tasks, because the order is used as the logic in which the tasks are executed.

Without these requirements, we are not able to identify the process based on the executed process information. In Figure 2.1 these requirements are met with the columns *Case ID*, *Task ID* and *Timestamp* and therefore, it is possible to rediscover processes using Process Mining. When applying Process Mining, the objective is to identify these three requirements in the executed log of processes. Using this information we are able to mine with different perspectives.

2.5 Process Mining perspectives

Process Mining can be used for different perspectives and is not limited to process discovery. We distinguish six different perspectives for Process Mining. These are shown in Figure 2.3.

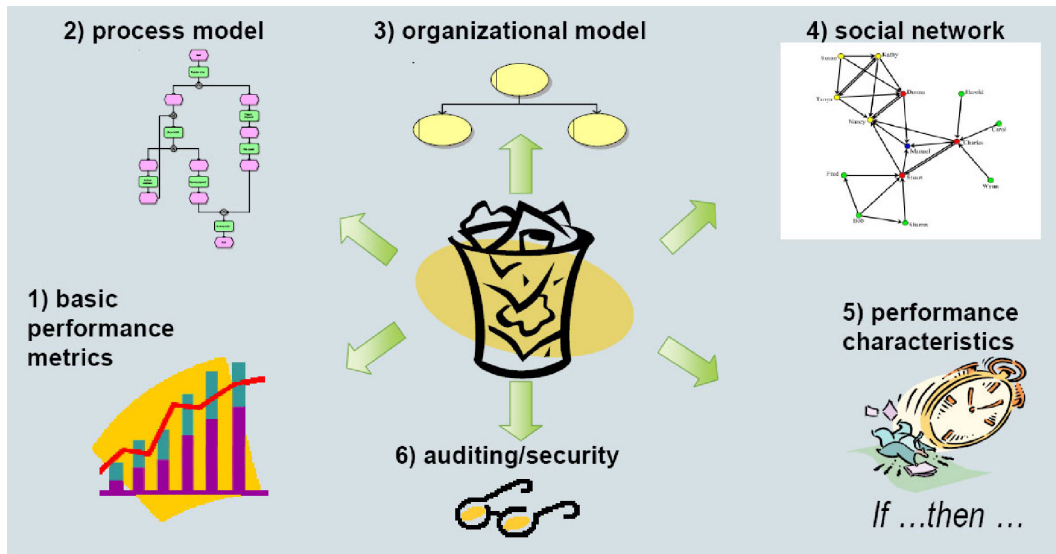


Figure 2.3: Different Process Mining perspectives

The dustbin in the middle represents the recorded transactional data taken from the information system. Usually, this dustbin is filled with data somewhere in a table of a database. Based on the information available in the database, it is possible to use different perspectives of Process Mining. A brief explanation is given for each mining perspective.

1. Basic performance metrics

These are standard performance indicators that originate from the world of logistics. Some of these indicators are the average throughput time, failure rate and capacity utilization. These indicators are important in measuring the performance of a concerned process, but they do not give an overview of how the process works.

2. Process model

Process discovery is the most common perspective of Process Mining. The example, given in Figure D.3, gives such a process discovery. A process model is discovered, based on a given information log. This model represents the actual routing of work within a system. As described before, the process model is discovered to identify differences between a process design and the actual process.

3. Organisational model

The third type of Process Mining is the mining of the organizational model. Organizations can have complex organizational structures and this mining type can analyze the roles people have in an organisation and generate organisational trees. Using the information regarding the users of an organisation in the process log, an organisational model can be discovered.

4. Social network

The relationship between people is important when looking at the social network. This type of Process Mining mines the flow of work that passes from

one person to another. It can detect that always the same group of people execute the same task and others are doing nothing. This, in particular, is important to indicate within organizations. When an employee is relieved of his duties, the social network can view the social network of that person and use it for the training of new employees. More information about the mining of social networks using Process Mining is described in [6].

5. Performance characteristics

This type of Process Mining is based on the if-then question. For example, if a request for product X and Y is produced (assuming this is unusual), then there is always a delay in the delivery, because product X and Y can use all their own systems in parallel except for one product and that one needs to be configured specially for the product. This is a very small example, but when production lines become more complex it can be useful to analyze the process based on its performance and relate it to a particular instance causing the problem.

6. Auditing/Security

The last of the Process Mining goals is the auditing and security. This Process Mining type is able to check whether there are security flaws in the design of a process model. An example of a security flaw is when person "A" can declare bills on the company expense account, anybody, but person "A" must approves of these expenses.

For each of these perspectives, some data elements need to be found in the transactional process logs (dustbin). We do not discuss these elements here, but refer to [16]. By default, these perspectives are supported in the ProM framework. We describe the ProM framework in the next section.

2.6 ProM Framework

ProM offers a framework to support different mining perspectives discussed in the previous section. It has been setup to provide a platform for future mining perspectives. Some of the mining perspectives have been added to the framework as plug-ins. A "plug-in" is a piece of software or functionality that can be added or removed without affecting the total framework functionality.

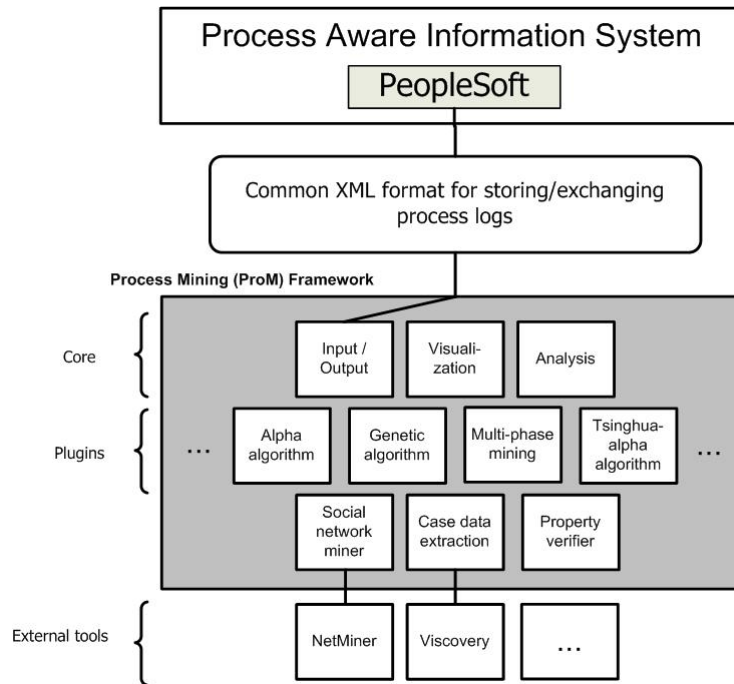


Figure 2.4: Overview of the ProM Framework

Figure 2.4 gives an overview of the ProM framework. We discuss the core of the ProM framework components individually [16]:

- Input/output component
This component handles the input and output of ProM framework files. As Figure 2.4 suggests, the input for the ProM framework are process logs stored in a common XML format. The structure of this XML file can be found in Appendix A. The output of the ProM framework is flexible. The ProM framework also supports different commercial product formats such as NetMiner² and Viscovery³). These tools are analytical tools used in the industry to improve business processes through providing statistical data and since they have not been implemented as a plugin within the ProM framework, having an export function, for these industrial standard tools is a plus.
- Visualisation
The framework provides a visualisation plug-in that handles the actual drawing of rectangles, squares, circles or any graphical two dimensional object for the Process Mining framework. This component uses the DOT⁴ language, which is the language that is used to describe the graphical elements [25].
- Analysis
The analysis component comprises of plug-ins capable of Petri net analysis, log replay analysis, and checking EPCs for their correctness using their relative

²www.netminer.com

³<http://www.eudaptics.com/loesungen/produkte.php?sprache=en>

⁴<http://www.graphviz.org/>

plug-ins. There are also plug-ins to compare a log and a model (i.e., conformance testing [23]) or a log and an Linear Temporal Logic (LTL) formula.

The plug-ins supported in the current ProM framework has evolved from smaller, deprecated, independent mining tools. Furthermore, the ProM framework supports different modelling languages (e.g., Petri nets, and EPCs) and other output formats such as Social Networks.

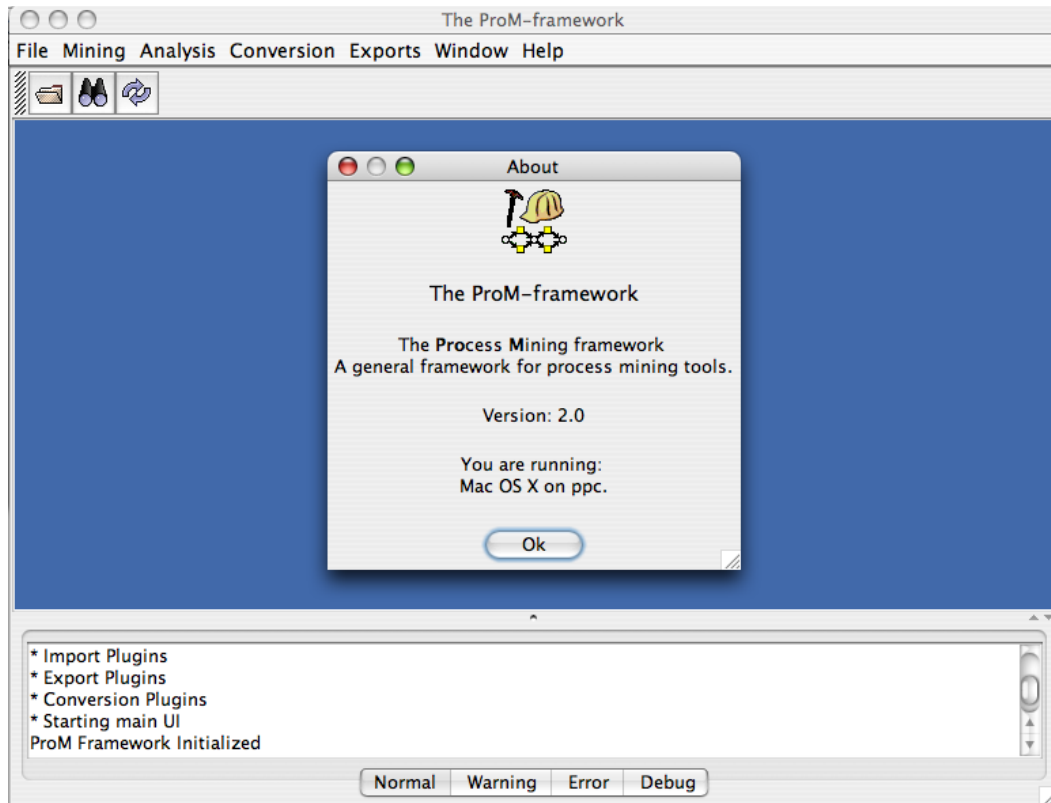


Figure 2.5: ProM tool screenshot

In Figure 2.5 a screenshot of the ProM tool is given. In addition, ProM has supplied interface description for the development of new plug-ins. To assist others that want to apply Process Mining to their information system, the Process Mining research group at the Eindhoven University of Technology has developed an interface that offers functionality to write the log in the correct XML format.



Figure 2.6: ProMimport tool screenshot

The ProMimport tool (Figure 2.6) is the universal interface for allowing people to develop their own converters to the common log format defined by the ProM framework⁵.

2.7 Available converters

In the introduction to Process Mining in the first chapter, we already mentioned that Process Mining has been applied to other systems. For each of these systems, a converter was written to transform the process log data into Process Mining XML data [24]. We briefly describe the most important aspects of the Process Mining to these systems.

1. Staffware

In TIBCO Staffware [3, 29], a Workflow Management System, finding the Process Mining information is straightforward. In the log files of Staffware, the case and the Task ID are explicitly given. Furthermore, for each executed task, there is a timestamp. The Staffware converter simply parses the Staffware logs and outputs the data in the Process Mining common format.

2. FLOWer

Similar to Staffware, the Workflow Management System FLOWer [27] also has Process Mining data such as Case ID, Task ID about the executed activities readily available. The data is stored in the database. The FLOWer converter

⁵Documentation about the ProMimport can be found at <http://www.processmining.org/>

extracts the data from a particular table and writes to the common Process Mining log format.

3. SAP R/3

Unlike any of the two Workflow Management Systems, in SAP Enterprise Resource Planning system R/3 [28], we encounter problems in finding Case IDs in the transactional log files. As described by [7], enterprise information systems are incredibly complex and consist of a large number of applications and components. These applications/components support fragments of a process and as a result the information required for Process Mining is scattered over the system. Obtaining the required Process Mining data is therefore far from trivial.

However, [17] describes an implementation specific solution to collect the Process Mining data, but the solution is not as generic as those in Workflow Management Systems.

For the application of Process Mining to Workflow Management Systems such as Staffware and FLOWer, the solution is generic. Unfortunately, for the SAP R/3 ERP system, we were only able to apply Process Mining specifically to one implementation and for one module. Future research into the application of Process Mining to other ERP systems should indicate whether other ERP systems also log their process executions in different places and thereby obstructing a generic Process Mining methodology.

2.8 Conclusion

In this chapter, we have discussed the fundamentals of Process Mining. The diagnosis of process executions used to aid process (re-)design. Process Mining is not limited to process discovery. As we have seen from the Process Mining perspectives, there are many possibilities to use Process Mining. In order to apply Process Mining to a system, we need to obtain process logs that describe the behaviour of a BPMS. Then, by matching the terminologies such as Case ID, Task ID of the BPMS log, with the same elements in the common format for storing Process Mining logs, we are able to mine processes. This research focuses on the application of Process Mining (i.e. the matching of terminology and building a converter to the common format) to other systems. From the application of Process Mining to SAP R/3 ERP system [17], M. van Giessel has indicated that finding a generic solution is difficult.

Chapter 3

PeopleSoft ERP

3.1 Introduction

This chapter describes the PeopleSoft ERP system that is used for conducting the research. To explain what an ERP system is, we start with a brief history of ERP systems in general, in Section 3.2. In Section 3.3 we explain the Process Aware Information System (PAIS) and indicate that ERP systems are also part of the PAISs. Another group of PAISs are the Workflow Management Systems (WfMS). When a comparison is made between WfMS and ERP systems, we see commonalities and differences. These are explained in Section 3.4. As this research is focused on one ERP package, PeopleSoft, we explain about the changes it has undergone over the years in Section 3.5. Highlighted for this research is the financial system, PeopleSoft Financials. For this reason, we give an overview of how PeopleSoft Financials is set up in Section 3.6. Finally, the chapter ends with the conclusions in Section 3.7.

3.2 History

Software applications in the seventies, if existent in organizations, were custom built. These applications were usually able to execute only one task repetitiously. Slowly, a trend emerged in making software that is easily adaptable to other organizations. Software developers now only had to make one software package, and sell the same package to different organizations with minor adjustments. Several administrative tasks were among the first to be automated, because many of them are similar in organizations, except for a few details. These details could then be manually changed. More of these standard applications have been developed to automate processes within organizations. Although many processes had been automated, there was no integration between applications. Information was printed using application and then re-entered into another automated process application. There was a necessity for a complete software package capable of integrating these small individual packages.

There are two answers for this problem, namely making them integrate using the latest technology or use one software package that contains the functionality of all these software packages. The first solution is called Enterprise Application Integration (EAI) and the latter solution is the Enterprise Resource Planning (ERP). ERP is the collective name for a large enterprise software solution capable of support-

ing different organization fields, such as financial operations, costs control, human resource management, project planning, administration, marketing, sales, and distribution.

ERP is the successor of MRP (Material Requirements Planning (MRP I), Manufacturing Resource Planning (MRP II)) systems and support processes broader than production processes [20]. Large ERP systems and the implementation of these system are expensive, and for smaller organizations the full functionality of the system is never used. Therefore, many ERP system developers have split their ERP functionality, for smaller organisations, into modules. Although the software modules for small organisations might be the same, they will differ in implementations. Now only the modules which are relevant for an organization are selected for customization.

3.3 Process Aware Information System

An ERP system belongs to a special group of information systems, namely the Process Aware Information System. PAISs are systems that are driven by process models, of which some are more explicit than others. Other PAISs are Workflow Management Systems (WfMS) and case-handling/ Customer Relationship Management (CRM) systems. For some of these systems, has been generically applied, but not for ERP systems. However, ERP systems, unlike WfMS and case-handling/CRM systems, do not work with an explicit process model. ERP systems usually work with a process model which is hidden within the software [9], or with a reference model which is not “real-time”. These process models in ERP systems are inflexible, when compared to models in WfMS, and case-handling/CRM systems. Using the technique, we try to obtain the process models of an implementation of an ERP system. For this research we use the PeopleSoft ERP system, and it is interesting to find out how process aware the system is. Does a PeopleSoft implementation work with explicit process models? The approaches given in Chapter 4 should yield an answer to this question.

When we look more closely at ERP and WfMS systems, we notice that they share the same goal: to ultimately improve business processes. Their approach in reaching that goal differs strongly. When managed correctly, a coherent picture of what is happening in an organisation can be captured in both systems. This supports better decision-making and planning, because data from departments are integrated and accessible throughout the system. The integration is especially important when legacy applications are involved, since very often they do not integrate with other systems.

3.4 WfMS vs ERP

When we compare WfM systems to ERP systems more closely, we notice that while both systems focus on business processes, the approach taken by each system is distinct. A WfMS, as described in Section D.3, uses a workflow model, which is then customised to accommodate specific business process structures. ERP systems are implemented around the idea of prefabricated applications. ERP manufacturers develop applications for particular sectors of the industry. Organizations acquire

the applications according to their needs. The workflow model is embedded in these applications. This means that many applications have pieces of the process model “hard-coded” in the software. The ERP software applications are built with parameters to allow some flexibility in configuring business processes. Each application includes the logic necessary to control organizational processes from a data and information point of view [9]. To achieve a better fit with the implementing organization, applications often need to be tailored by setting thousands of parameters. Figure 3.1 represents one of the key differences between WfMS and ERP systems.

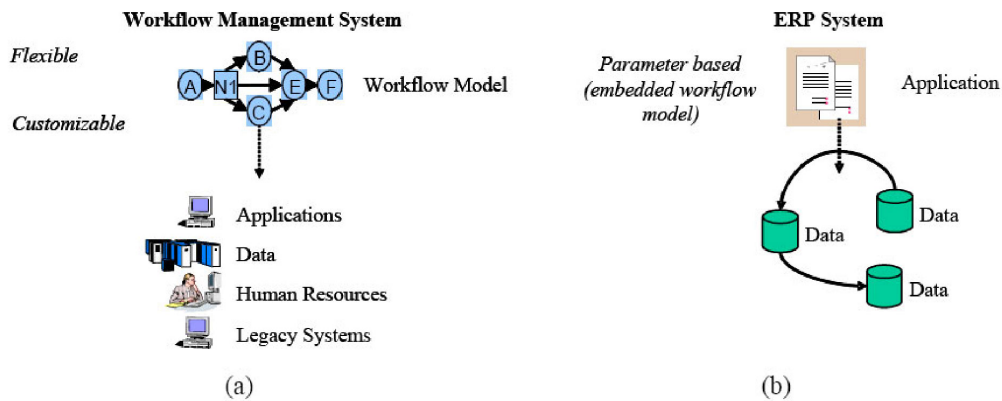


Figure 3.1: WfMSs and ERP systems

Using a workflow definition and model, the system implementers can develop applications to support the flow of work described in the workflow model (see Figure 3.1a). These applications work together to form a WfMS. Workflow systems are process-centric, focusing on the management of flow logic [9]. For ERP systems, the workflow model is usually embedded in the application software (see Figure 3.1b). System implementers work at configuring the system by setting parameters that are accessed by the applications.

To compete with the clarity of process modelling used in Workflow management system configuration, ERP systems introduce the notion of reference models. These models are best practice process models that can be used for the configuration of an ERP implementation using parameters [10]. However, this is not how things are being configured in ERP systems at the moment, but this would be the ideal solution. The more parameters an ERP system has, the more the workflow process can be configured. The final configured ERP workflow model is therefore in many cases not visible because it is stored in the application logic and parameters tables. Logically, with WfMSs, developers tend to have more flexibility since they are developing the workflow model directly, being limited only by the tools provided in the WfMSs development platform. The ERP systems are data-centric, providing a common homogeneous data infrastructure across the organisation [9]. The question is: “Is PeopleSoft also data-centric as well as process-centric?” If we are able to mine processes from PeopleSoft, this means that we are able to identify processes from a running implementation and therefore, can conclude that it is also process centric.

The recent trend from ERP vendors is to integrate Workflow Management Technology into ERP systems. For example, in order to facilitate the customization and

deployment of ERP systems, some vendors are making their systems workflow-driven with the integration of workflow components. This workflow integration is however not equivalent to workflows in a Workflow Management System, and only interface existing ERP components with some kind of workflow functionality. In other words, in workflow systems, a workflow is required before any work can be executed, while in ERP workflow, the work can start without workflow functionality. Unfortunately, the integration of this workflow interface to components is not throughout the whole ERP system and therefore only available on particular modules.

3.5 PeopleSoft ERP

When PeopleSoft company first started, they built their ERP systems to support financial and human resource management (HRM). With their major release in the beginning of 1998, they have increased their functionality rapidly. These new functionalities were production and distribution oriented. The acquisitions of Red Pepper Software (1996) and Vantive (2000) helped in adding supply chain planning (SCP) and e-Business application to PeopleSoft. By doing so, it is suitable for distribution oriented discrete production (make-to-stock). Relating this to the previous Section, we see that other ERP applications are acquired and integrated into larger ERP systems like PeopleSoft. Furthermore, although PeopleSoft have increased their functionality for finite capacity planning, it is less suitable for shop-floor control and warehousing. PeopleSoft is currently very popular for their ERP solutions for universities and schools. PeopleSoft Campus Solutions allows universities to integrate their student financial data with their student progress data. PeopleSoft has recently been taken over by ORACLE, a company well known for their Database Management System (DBMS) software and ERP solutions. DBMS software is a suite of programs responsible of managing the database containing the data. Although ERP systems like SAP and BaaN have developed their own database and database management software, PeopleSoft supports different databases and thus different DBMS software. PeopleSoft is one of the first ERP suppliers to allow ERP applications to be accessed using a standard web browser [21]. Users no longer needed to install ERP client software, and are not dependent on the operating system. This technology is called the PeopleSoft Pure Internet Architecture (PIA). Unfortunately, for customizing applications embedded in PeopleSoft, one can not use the web browser. You need the PeopleSoft developing tool: PeopleTools[26]. This is a Windows application that is directly connected to the database of the ERP system and is capable of configuring the ERP system.

3.6 PeopleSoft Financials

For the research at the Fontys University of Professional Education we have access to their PeopleSoft Financials test environment. Because the system is for testing purposes only and is identical to the live environment, we are able to test if our requirements, discussed in Section 2.4, can be met. The financial system is primarily used for the billing of external people that follow courses offered by Fontys, for example, members of a software development company following a course on object oriented programming. Normally, students are billed from the Fontys Student

System (PeopleSoft Campus Software). This system also supports the billing of students for their annual tuition fees. We now have a look at the PeopleSoft ERP Systems internal architecture. An overview of the architecture, as developed by PeopleSoft, can be seen in Figure 3.2. Note that this figure is taken from PeopleSoft documentation [15] and is universal to all PeopleSoft ERP systems with Version 8.

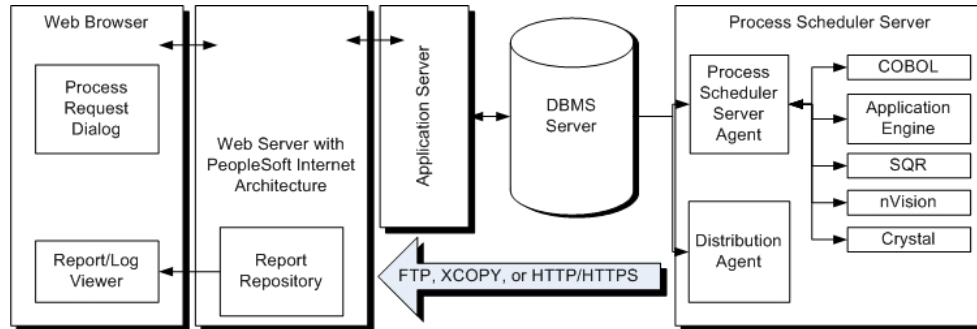


Figure 3.2: PeopleSoft Financials system architecture

At Fontys, they use different versions of PeopleSoft, but for this mining effort the focus is only on PeopleSoft Financials which runs on Version 8. Applications in PeopleSoft Version 8 and later are started from the web browser using the Pure Internet Architecture (PIA). This is depicted in the most left part of the picture. The PIA is introduced with PeopleTools 8 and is completely focused on the internet environment. Its main purpose is to provide internet-based ERP access and integration. Using the PIA, it is possible to access the ERP information anywhere in the world. The web server displays web applications which are defined in the application server and is responsible for the look and feel of the application. In the application server the content of the application is defined. There are many applications in an ERP system like PeopleSoft. These applications are stored, in this case, in an Oracle database. The database management system (DBMS) is the core of the system and is responsible for maintaining the database. ERP applications are prefabricated and the settings for the execution of these applications are stored in the database. When an application is started from the web browser, a processing request is placed in a table of the system database. This table contains a list of applications that have not been processed. The process scheduler server agent then reads the table containing all applications (COBOL, SQR, Application Engine, etc.) processing requests and schedules applications for processing. This is done by the Process Scheduler Server Agent. One application started from the web interface can start multiple processes. These kinds of applications are called jobs. Users of the PeopleSoft ERP system work from the left (the web browser) and system administrators operate from the right (the process scheduler server). On the left we see web browsers opening web pages and starting applications, and on the right we have different types of processes that are started, caused by the web browser. After a web application starts, the system can execute smaller applications related to the initial application.

3.7 Conclusion

Enterprise Resource Planning (ERP) is a common name for a large integrated software package capable of supporting business processes through process automation. Although it shares the same goal as a Workflow Management System (WfMS), it has a different approach, compared to a WfMS. The process model that drives the ERP system is not clearly visible. It is usually embedded in the logic of ERP applications, and therefore not easy to alter. ERP systems provide many parameters to change settings of the ERP application, but not the model itself. This is in contrast to WfMS systems. The collective name for systems that are driven by processes are called PAISs. For Workflow Management Systems it is easy to say whether it is process driven, but does the same hold for ERP systems and PeopleSoft in particular. Extracting the business process model from an ERP system, like PeopleSoft, is what is all about. Thereby indicating that there is an explicit process model that drives the system (process aware). The ERP system that is being analysed in this research is PeopleSoft Financials. Once a small ERP system known for their HRM module, it has grown by acquiring other smaller ERP packages (later on to be added as modules in PeopleSoft).

In 2005 Oracle took over PeopleSoft and hopes to compete with the larger ERP vendors. Analysing the PeopleSoft architecture, we can see how applications are started in PeopleSoft. We can use this as a starting point for our hunt for data.

Chapter 4

Overview of approaches

4.1 Introduction

We are interested in how we can generically add PeopleSoft Financials to the list of information systems for which we can mine processes. The attempts made to realise this, are called mining approaches.

From the Process Mining chapter, Chapter 2 we know that there are requirements for Process Mining that we need to find in PeopleSoft Financials in order to mine any processes. Then, we can export the information taken from the system and eventually build an adapter that transforms the data. Finally, we can generate process models, based on the process logs taken from the PeopleSoft Financials system, and evaluate the result of the mining. We are then also able to assess if the result of the mining is valid.

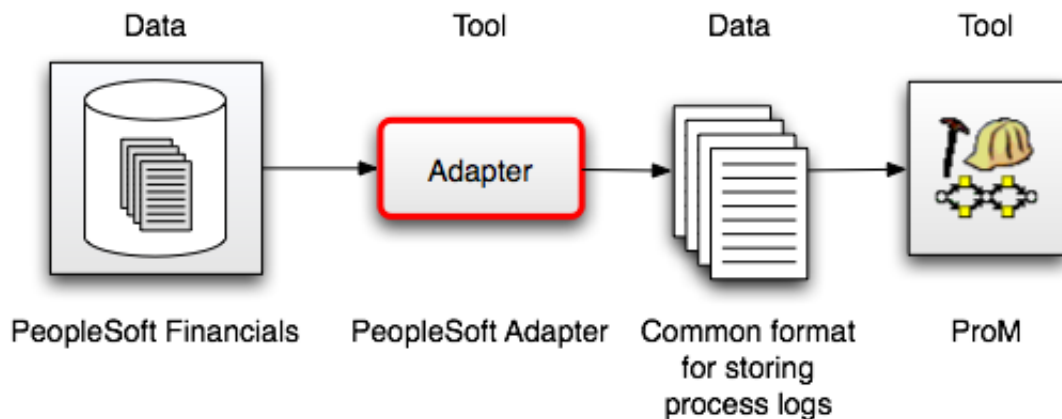


Figure 4.1: The PeopleSoft adapter.

Figure 4.1 displays the research aim visually. Since there is no assurance that the common format requirements, discussed in Section 2.4 exists, we have to look deep into the PeopleSoft Financials ERP system to find the required information.

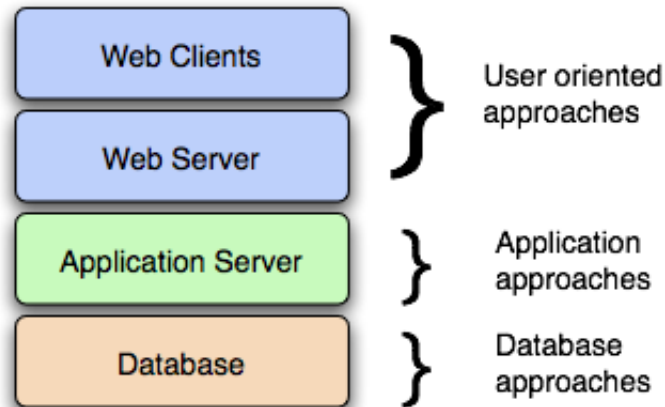


Figure 4.2: An abstract overview of the type of mining approaches applied to PeopleSoft Financials to find the data in the system.

We have attempted to find the Process Mining requirements in PeopleSoft Financials using several approaches. Some in more detail than others. The approach taken to systematically look for Process Mining information is based on the system architecture of the PeopleSoft Financials ERP system (see Figure 4.2). We have split them into three categories:

1. Application mining (Section 4.2),
2. Database mining (Section 4.3), and
3. User oriented approaches (Section 4.4).

For each of these approaches, we describe the four steps: (1) what our motivation was, (2) what actions were taken to realise the approach, (3) we describe the result of the approach, and (4) we evaluate the approach.

For two approaches, the *Application Engine mining* approach given in Section 4.2.1 and the *PeopleSoft pages* approach in Section 4.4.1, we were able to build a converter. Therefore, these approaches are more elaborately described than the others.

In Section 4.5 we give our conclusions and answer some of the questions raised in the beginning of the research.

4.2 Application mining approaches

The first type of approaches are the PeopleSoft application related approaches. The assumption that we made for these approaches is that PeopleSoft Financials only consists of applications that, in turn, modify data. In Section 4.2.1, we look at one type of application, and from there, try to gather information about which other applications are executed for the same process. The approach that followed, described in Section 4.2.2, focusses on multiple applications and the process scheduling server, which is the core of the PeopleSoft application scheduling system. In Section 4.2.3 we see what the results of the efforts are.

4.2.1 Application Engine mining

Motivation:

In an ERP system like PeopleSoft Financials, there are many different applications. Confusingly, these applications are internally called processes in PeopleSoft. Applications are executables started from the user interface of PeopleSoft Financials. These applications are capable of modifying data within the system in one way or another. In PeopleSoft Financials we distinguish five major application types, namely Application Engine, COBOL, SQR, nVision and Crystal.

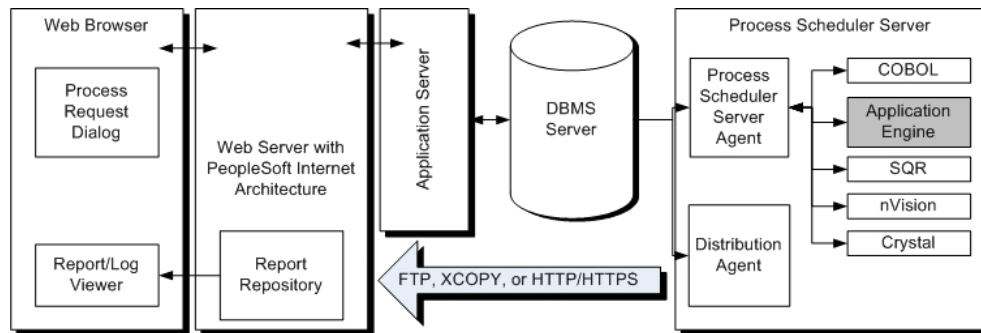


Figure 4.3: Application Engine in the PeopleSoft system architecture.

In Figure 4.3 we see these applications on the right extreme part of the PeopleSoft Financials system architecture. For this approach we assume that an application is process and every application execution is a process instance. We choose to look at one type of PeopleSoft Financials applications, namely the Application Engine (AE) applications. The AE applications are highlighted in Figure 4.3. We notice that for every AE application, it is possible to log the application. Therefore, we analyze this log. The reasons for selecting this AE log are:

- It contains information that we can use for Process Mining;
- It is reasonable easy to enable the logging;
- There is trend to shift the functionality of non PeopleSoft proprietary applications to AE applications;
- AE supports the largest amount of applications within PeopleSoft Financials.

The log file of the AE is called a trace file or trace. The AE trace (AET) is a text file registered at the time of execution. By default, no AET log is created, because logging the AE applications is an extra load on the PeopleSoft Financials system. The logging of these AE applications is only enabled, when errors occur in the execution of applications.

Action:

In the action phase we distinguish two steps. Firstly, we need to know how the AET file is constructed that we eventually use to convert, and then we build a converter

that actually converts the AET log in the common XML format for storing logs.

Application Engine Trace (AET)

All applications, also that of type Application Engine, are executed in the Application Server. When the logging option for an AE application is enabled, then for every execution of this application a separate log is created. The format in which the log is stored in PeopleSoft Financials gives us useful information about the executed application. The AET logs have the following format:

```
“application_type”_“application_name”_“application_instance”.“log_type”
```

For example, the trace file: “AE_AP_APGLOPEN.441515.AET” has “AE” as the type of application namely, Application Engine abbreviated. Furthermore, the file-name contains traces of the Application Engine application: “AP_APGLOPEN” and the number “441515” is the instantiation of the application. Every execution of this application can be uniquely identified by its number, the application instance. Finally, the “AET” is the default extension of an Application Engine Trace file. An AET file contains lines of events. These events are recorded activities in the AE application. The first few lines of the trace file are:

```
1 -- PeopleTools 8.42.15 Application Engine
2 -- Copyright (c) 1998-2004 PeopleSoft, Inc.
3 -- All Rights Reserved
4 -- Database: FFEZ (Oracle)
```

These lines contain:

- The PeopleTools version: 8.42.15;
- The process type: Application Engine;
- The name of the database: FFEZ;
- The type of database: Oracle.

After these first lines, the trace starts with the following line:

```
5 -- HH.mm.ss YYYY-MM-DD Tracing Application Engine program <process name> <description>
```

The application name has the same value as the name obtainable from the filename. It is a small application in the Application Engine. The description of the application can only be obtained from line 5. From this line also the start date of the application is obtained: *HH.mm.ss YYYY-MM-DD*.

Other lines usually have the following format:

```
6 -- HH.mm.ss LVL (PROCESS.SECTION.STEP) (DESCRIPTION)
```

The first two “_” indicates that this line is the start of an event, and then followed by the timestamp “HH” for hours, “mm” for minutes and “ss” for seconds. The “LVL” indicates at which level the event took place. An AE application is split into sections and each section has several steps. A trace can contain subsections which are executed from the main section. For instance the line:

```
7 -- 15.04.06 .(AP_APGLOPEN.MAIN.000-1) (Call Section AP_APGLOPEN.A000)
```

In turn causes the following event in the file:

```
8 -- 15.04.06 ..(AP_APGLOPEN.A000.A000-1) (SQL)
```

Note that the LVL was “.” and after the call the level is “..”. After all the steps of section “A000” have been executed the LVL returns to “.” and continues with the next step in the “main” section. The rest of the event contains SQL code as indicated in the description and every event finishes with a slash “/” at the end. An example of an event is:

```
9 -- 14.44.23 ..(AP_APGLOPEN.A000.A000-2) (Do When)
10 %Select(TEMP_FLD) SELECT 'X' FROM PS_APGL_OPEN_RQST WHERE ' ' = 'Y'
11 /
```

Line 10 is a piece of SQL code. It is not required to understand the query.

The AET trace file ends usually with:

```
12 -- HH.mm.ss Application Engine program <program name> ended normally
13 -- HH.mm.ss Application Engine ended normally
```

If the trace file contains line 5 in the beginning of the trace, then line 12 will be there at the end of the file. In other words if an application is traced, then a beginning and ending of the trace exists, unless the program has errors.

AET Converter

The converter has as input a PeopleSoft AET file, and converts this to a common format for process logs. This common log is described in Appendix A. The AET converter starts by reading the file, taking the application name and the application instance from the filename. This is stored as the process name and process instance in the common XML format. Then, every line that begins with “- -” and a timestamp is converted to an AuditTrailEntry and the WorkflowModelElement is the application name concatenated with the section and step.

```
14 -- 15.04.06 ..(AP_APGLOPEN.A000.A000-1) (SQL)
```

Line number 14 is another example of an event in an AET file. The whole line is considered as an AuditTrailEntry, and the WorkflowModelElement is “AP_APGLOPEN.A000.A000-1”. The application name is “AP_APGLOPEN”, section “A000”, and the step is “A000-1”. The EventType is in this case always “complete” because the line is written after that WorkflowModelElement has been executed. For the Timestamp the date which is taken from the beginning of the file and the time from the beginning of every line is used. As in line 5 we have the date, and in line 7 we have the time: “15.04.06”. For the “ORIGINATOR” we use unknown, but the information is available in the trace file. The value of the originator is stored in the OPRID variable in the AET file.

The general idea behind this converter is first read only the lines that start with “- -” and then when there is a timestamp in the form of “HH:mm:ss” it is stored, and the parsing of the line continues. The next thing that is stored is the value between the first parentheses and the description (e.g. line 14 the description is “SQL”) is not stored. Once this is finished, the converting process has been completed. Further details about the AET converter, can be found in Appendix ??.

Result:

For the analyzing of AET logs, we have built a converter. Now, we use the ProM tool to generate models of the converted AET logs.

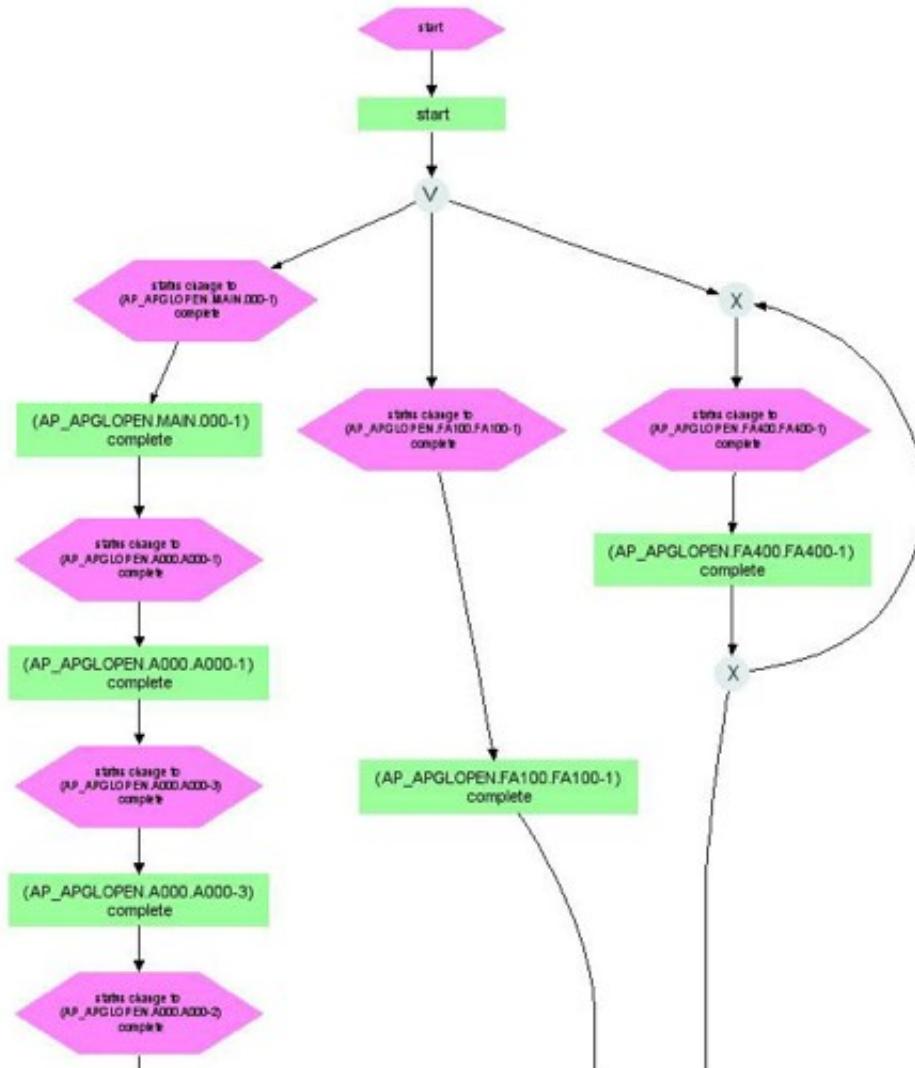


Figure 4.4: Aggregated EPC of AP_APGLOPEN application executions.

Figure 4.4 is a large diagram, and we only show the start of the trace. At this moment we have obtained a process model in the form of an Event-driven Process Chain (EPC)[15]. The result of the mining based on the converter we built was not interesting, because the log file we analyzed from an AE application only describes which steps were taken for that particular application execution. The process model is a sequential model with each branch displaying the different application executions. These steps are the same for every application execution of an application.

Evaluation:

With this approach we are able to visualize PeopleSoft Financials application. However, this AE application is of a too low level and does not cover a business process. Furthermore, from a single application we are not able to find relations to other applications. Therefore, the next step is to investigate what caused the application to be executed. In other words, we need to look at a higher level (see 4.2.2).

4.2.2 Process scheduler**Motivation:**

In PeopleSoft Financials, applications are centrally scheduled for processing in the Process Scheduler Server¹. Therefore, the assumption is that this server is the place where information regarding which applications are executed for which process instances, could be stored.

Whenever an application has to be executed, a request is done to the process scheduler for execution. For that application a time and/or date is set when the application is to be executed. This is stored in a table of the database and the process scheduler server uses this table to eventually execute the applications in the PeopleSoft Application Server. When we look more closely at the scheduling of applications, we notice that a unique number is assigned to each application before execution. This unique identifier is issued by the process scheduler and also stored in the scheduling table. Therefore, using the unique identifiers, it might be possible to collect information about which applications are executed for a particular case.

Action:

When we investigate the assumption, we notice that once the execution of an application is complete, the unique execution identifier is reset to zero. Thereby, we lose information regarding which applications are scheduled for execution of a particular process. We are not able to use the unique identifiers as information for case id's. Therefore we are not able to build any converter, because we are not able to find the information we are looking for using this approach.

Result:

The Process Scheduler Server might contain information about the PeopleSoft applications that are executed for a particular process. Unfortunately, this information is lost at run-time. By default, no log is generated by this process scheduler server. When such a log file needs to be created, all the 512 AE applications in the PeopleSoft Financials system need to be altered, because it is coded in every application that a request to the process scheduler server must take place before being executed.

Evaluation:

Using the approach given here, we are not able to obtain information for Process Mining, but if we were to store this unique case id manually, then we would have to

¹PeopleSoft uses the term "server" because the scheduling of these applications can be very strenuous and in such cases a different physical server is used for the process scheduling.

change all applications in PeopleSoft Financials. Changing the applications would not be feasible and the solution would be too implementation specific. Applications might be customized for a PeopleSoft implementation, and if we want to apply this same approach to other implementation we would still have to change the applications individually.

4.2.3 Application mining results

The conclusion we can draw from the application mining approaches is that in an ERP system, like PeopleSoft Financials, there are a lot of applications. Using the information available from a single application execution and the process scheduler server, we miss information on the business level. By business level we mean information about the business process that ultimately caused applications to be executed.

4.3 Database mining approaches

The database mining approaches look at the database of the PeopleSoft Financials system. Fontys has opted to use Oracle DBMS software for their database. PeopleSoft is quite unique in this sense, because many of the larger ERP systems supply their own database.

The following approaches focus on what can be found when analyzing purely the database of PeopleSoft Financials. We start with the foreign keys approach in Section 4.3.1, and the information that can be found using SQL Triggers is described in Section 4.3.2 and finally we try something more specific, namely, table mining in Section 4.3.3. Once again, we summarize the results in Section 4.3.4.

4.3.1 Database foreign keys

Motivation:

PeopleSoft stores all the data in tables of the database. The database tables used for processes can be easily found when we have an overview of which tables are linked to each other, using foreign keys². Once such an overview is obtained, we know exactly when and where tables are accessed, and data is stored.

We make the assumption here that the activities that we are looking for, change the data in the database tables. Therefore, we have to look at which tables were altered for an activity.

Before we can identify the tables that are changed by an activity, we try to create a metamodel of the PeopleSoft Financials database. This metamodel should provide an overview of which activities use which tables and hopefully, also which processes. A database metamodel describes more than only the relation between tables. According to [19], a database metamodel is a description of those database properties that are not dependent on a particular database state. So, by reconstructing a metamodel of PeopleSoft Financials, we know what is required for a PeopleSoft

²A foreign key is a field or group of fields in a database table that point to a key field or group of fields forming a key of another database table in some (usually different) table. Source: http://en.wikipedia.org/wiki/Foreign_key

database. Combine that with the knowledge we get from the links between tables and we might be able to have an idea of how processes are supported in PeopleSoft Financials.

Action:

In order to find links between tables we require foreign key support for the database. Unfortunately, foreign keys have not been implemented in the Oracle database. Therefore, no overview of all the tables can be made. No actions towards Process Mining were performed.

Result:

No result, because foreign keys are not implemented.

Evaluation:

Information regarding how PeopleSoft Financials database tables are linked to each other cannot be found, so we look for other alternatives for finding interesting tables for mining.

4.3.2 SQL triggers

Motivation:

Similar to the Database foreign key approach, this approach uses a different technique to see which tables are modified during activities. SQL triggers are procedures (i.e. stored SQL applications) that are “triggered” when basic table operations are performed: Insert, Update, or Delete. Triggers are similar to stored procedures. A trigger stored in the database can include SQL, PL/SQL, or even JAVA statements, and can invoke stored procedures. A procedure is explicitly run by a user, application, or trigger, whilst triggers are implicitly fired by ORACLE when a triggering event occurs, no matter which user is connected or which application is being used.

Action:

The underlying database of PeopleSoft Financials at Fontys supports the use of SQL triggers to see which tables are modified during some activities. To use this technique, we need to place triggers to tables. However, we do not know to which tables we have to add these triggers. PeopleSoft Financials has in excess of 13.000 tables and adding SQL triggers to each of these tables would cause the system to slow down, because every database access would execute the SQL trigger. Furthermore, with the database alone, we cannot get information about the applications causing to trip the trigger. So, no further implementation steps were taken for this approach.

Result:

No result, because nothing was implemented.

Evaluation:

SQL triggers are supported in the PeopleSoft Financials database, we have to add triggers to all the tables before we might find the information we are looking for. Therefore, we suggest the concept of table mining.

4.3.3 Table mining**Motivation:**

Another approach is to find out which tables belong to a certain process and then monitor those tables using triggers. By recording which database tables are accessed, we can place triggers to these tables.

Action:

The approach starts by looking at one process of PeopleSoft Financials, the billing of customers. The concept of this approach is applicable to other processes. For this section we take this one as an example. Whenever an application requires data from these billing related tables, the triggers are fired. The limitation of this approach is that we define to which tables triggers are placed beforehand. Therefore, if we already know which tables are used for an activity, we do not need to obtain it from the system using triggers.

Result:

Therefore, the result is only limited to the processes that use tables to which triggers have been placed and not all the business processes. We have performed tests to find out which tables belong to which activities. No actual implementation took place. Example of this manual table hunt is described in Appendix B.

Evaluation:

Table mining is specific and requires knowledge about the processes before any triggers can be placed. This makes the approach subjective. Although the idea behind this is applicable to all PeopleSoft Financials implementations and requires triggers to be supported, we describe in Section 4.4 an approach that is objective without the use of triggers and also applicable to all implementations, including Fontys.

4.3.4 Database mining results

What we can conclude from these database approaches is that in PeopleSoft Financials the business logic is not stored in the database, but stored in the applications. Hence, the type of database is totally irrelevant for PeopleSoft Financials to work. Information regarding business processes and PeopleSoft Financials applications are not stored in the database. Only the configuration of these processes and/or applications are stored and retrieved. Table mining attempts to find out which tables are important for a particular business process, but in order to find the tables we need to “cheat” (i.e. have prior knowledge of the processes) and that is not what we had in mind before we started.

4.4 User oriented approach

Another category of attempted Process Mining approaches, are the user oriented approaches. From the user perspective, we define that tasks are what users of the PeopleSoft Financials system actually do from their web browsers. We start by looking at PeopleSoft pages in Section 4.4.1 and continue with PeopleSoft components in Section 4.4.2. In Section 4.4.3 we summarize the user oriented approach results.

4.4.1 PeopleSoft pages

Motivation:

In WfMSs it is important to realize that workflows consist of tasks that are executed by one or more resources [4]. Compared to an ERP system, in this case PeopleSoft Financials, activities executed by the user can also be classified as instances of tasks. However, the difference between Workflow management systems and ERP systems like PeopleSoft is that usually, all the activities occur within PeopleSoft. There are however activities that are executed manually, thus without using the PeopleSoft Financials system. These actions are not in any way registered in the system and therefore obtaining these activities is obviously not possible.

Activities within PeopleSoft Financials, as for all the other different PeopleSoft systems, can be abstractly classified as mutations of data. In other words, users who work with the financial system use the system to execute business process steps by accessing and working with PeopleSoft Financials. They work by executing small graphical user interface (GUI) applications that handle underlying database tables, and thus data, transparently. Once the user finishes with an application, other applications can be started or the same can be restarted, by the user. The execution of one application is what Workflow Management technology calls an activity, the actual performance of work [4].

As the system is responsible for the interaction between users, let us look more closely to the user interface of PeopleSoft Financials and how that relates to tasks.

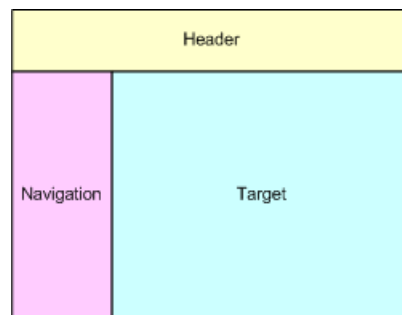


Figure 4.5: PeopleSoft user interface areas.

Users of the PeopleSoft Financials system connect to PeopleSoft Financials through an internet browser. Therefore, the applications that are executed by users are web pages. The user interface of PeopleSoft displayed in Figure 4.5 can be split into three areas: header information, navigation, and target. The structure of the screen is similar to websites, where the header information displays information that is valid

for all PeopleSoft web pages. Examples of header information are the organizational logo, logout option and a link to the homepage. The homepage is the view the user gets after a successful login into the system. The navigation area of the screen contains hyperlinks to web pages that are visible in the target area.

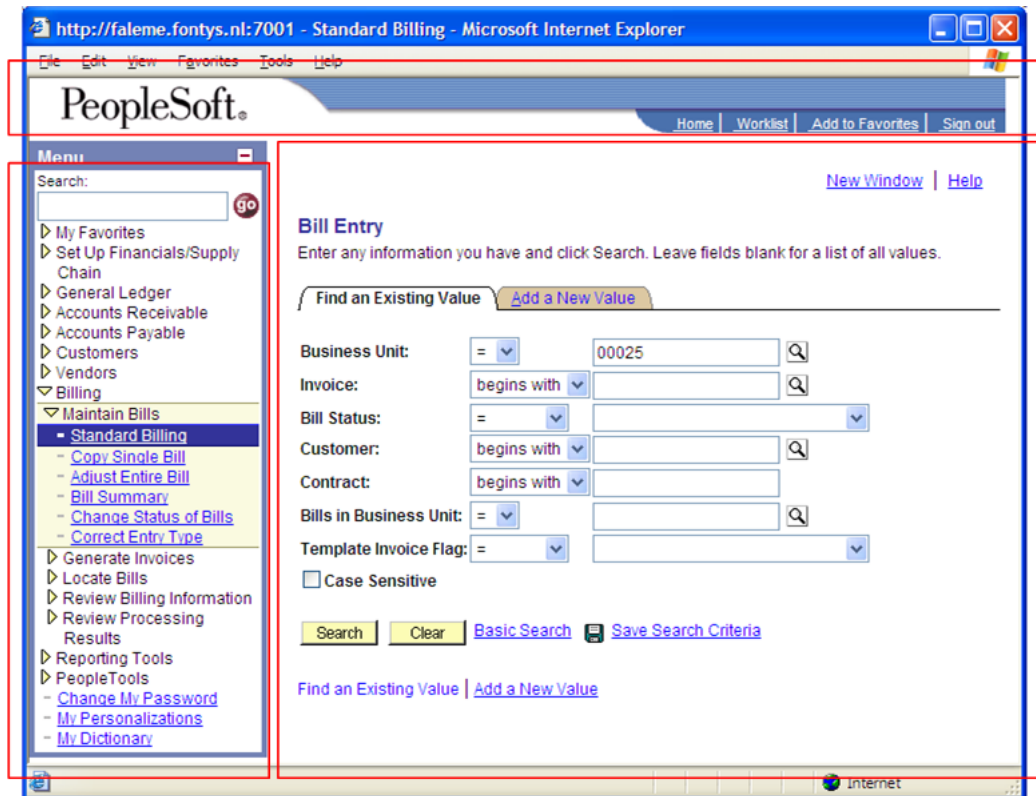


Figure 4.6: PeopleSoft user interface example.

Figure 4.6 is the view of PeopleSoft Financials after successful login. From the three rectangles we can see the three areas clearly. In this example, the user has selected the hyperlink “Standard Billing” beneath the “Billing” and “Maintain Bills” user interface option.

From PeopleSoft documentation³ we learn that the hyperlinks can open PeopleSoft components in the target area. Components are PeopleSoft’s web applications.

³<http://eiswbpapb.admin.unt.edu/PSOL/htmldoc/eng/psbooks/index.htm>

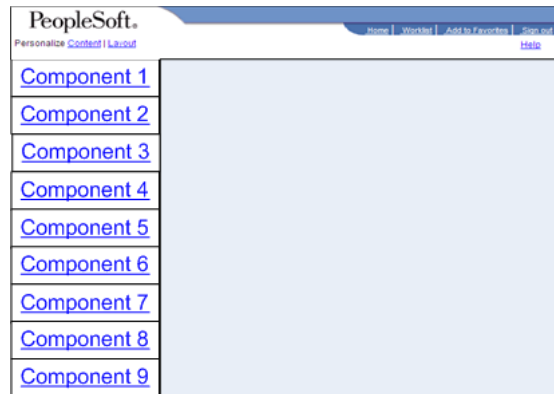


Figure 4.7: PeopleSoft component hyperlinks.

The user in Figure 4.7 has the possibility to choose nine different hyperlinks. In this case, a component can be “Standard Billing”, or “Copy Single Bill” in Figure 4.8. A more formal definition of a PeopleSoft component is, a PeopleSoft term for a set of applications that is capable of modifying data. Components in turn can contain one or more PeopleSoft pages, which are smaller applications that can also modify data. Note that a web page is not the same as a PeopleSoft page.

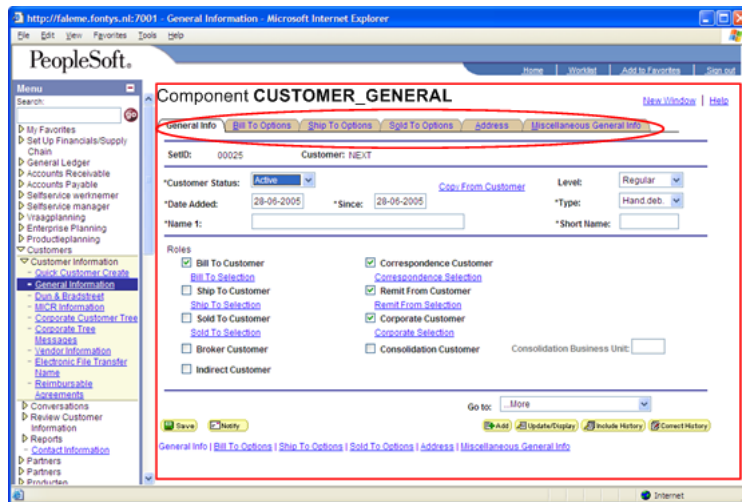


Figure 4.8: PeopleSoft component and pages.

The PeopleSoft component that has been selected, is in this case “Customer General”, and has the PeopleSoft pages “General Info” (page currently selected), “Bill To Options”, etc. A page in Figure 4.8 is one part of a component. PeopleSoft pages contain User Interface fields such as edit boxes, scroll lists and check boxes. User Interface fields can be used by a PeopleSoft Financials user. Transparently, these components, pages and fields are mapped to the PeopleSoft database. When a user accesses a page and starts entering data, he can start different components or can be automatically forwarded to other components. The automatically forwarding of pages can happen by defining on the page level, what the next step

is. At this point we make the assumption that each page corresponds to a task.

According to [4], a task is logical unit of work that is carried out as a single whole by one resource. The tasks of that user will consist of components, and pages that were carried out after logging into the system. The collection and the order in which these are executed will be our process model.

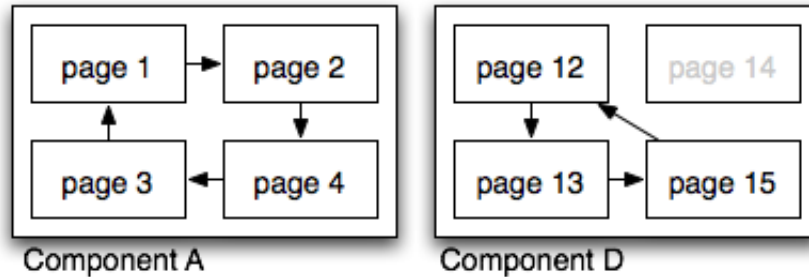


Figure 4.9: Pages within components

Figure 4.9 visualises PeopleSoft pages within PeopleSoft components. Component *A* has page *1*, *2*, *3*, and *4* in use, while component *D* only has some of its pages in use. During the configuration of the ERP system, these pages are either enabled or like page *14*, disabled. What we also notice is that in component *A* and *D* there is routing between pages. This routing is what we wish to mine for this approach, because we assume that a page corresponds to a task.

Action:

The actions taken for this approach are divided into two steps. First, we have to generate the logs containing information about the pages accessed by users. Secondly, we have to build a converter that can parse the log files. Then we have to parse the logs files.

Generating log files

In PeopleSoft Financials, there are many ways to log users. Furthermore, there are many things that can be logged. We know which options we must have enabled before users work with the system and start generating log files. In Appendix E, we describe each setting of the logs in PeopleSoft separately. These settings are:

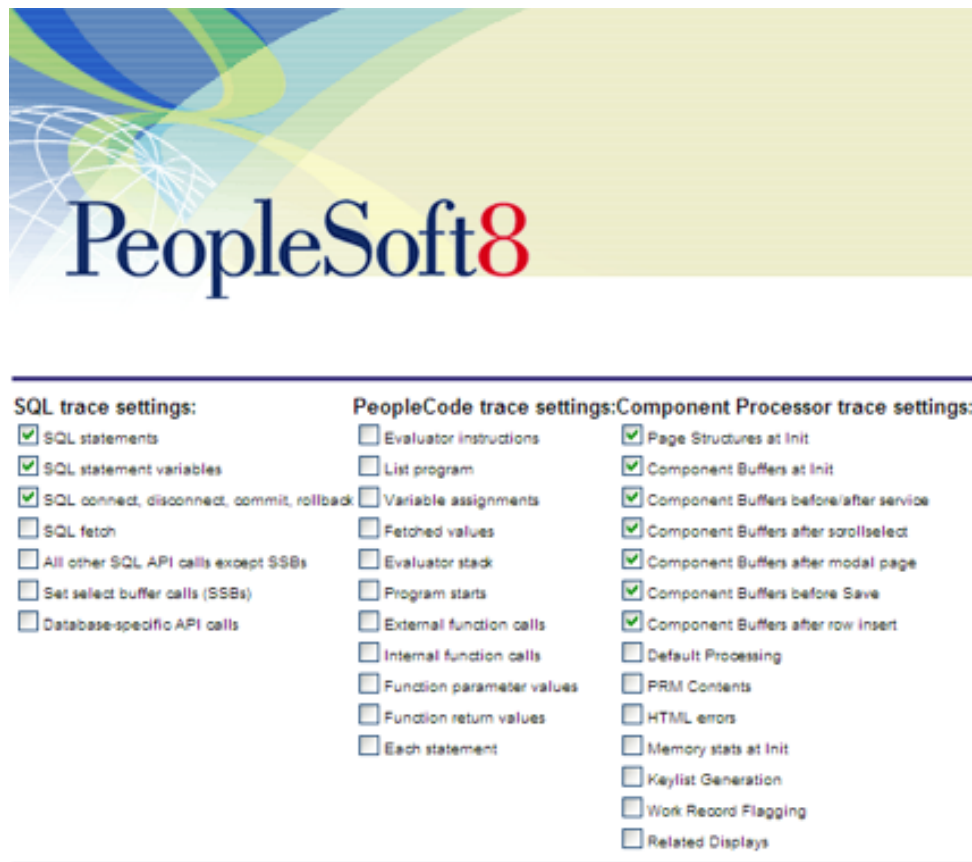


Figure 4.10: PeopleSoft logging options.

Note that the settings in Figure 4.10 are the possible settings for PeopleSoft Financials, and hold for all versions after Version 8.0 of PeopleSoft. These settings make it possible to record information that can be used to construct the Case ID, Task ID and Timestamp (required for Process Mining). The log files are stored on the server of PeopleSoft Financials. Now that we know which options need to be set in order to have the right information in the log files, we are now ready to log users.

The logging the users can be done in two different ways: manually, or automatically. Logging users manually means that we ask users to select the logging options in Figure 4.10 before they login into the system. Once they log out of the system, or they are inactive with the system for one hour, they are logged out. By default, no logging options are set, and users might forget to enable the logging.

Automatically logging users means that the logging is enabled on the server. When users login normally, automatically the correct logging options are enabled. The more users that are logged, the slower the system performs, because for every user a separate log file has to be created.

However, when we trace all users, we get a more accurate process model. We therefore need to find the optimal solution. So, in order to get an accurate view of the business process, we could select only those users that work, for example, in a certain area (location) or select users with same tasks, based on their job description. Another interesting thing about the log files is that for each individual user, a log

file is created. This means, that for every log file you already have the user that has executed that log of activities. At Fontys, we enabled the server logging during a period when not many people were available.

Unfortunately, this automated logging also has drawbacks. The automated logging causes the system to work slowly, because for every user a log file is created that records all the pages that has been accessed by the user. The size of an user log obtained from PeopleSoft Financials is approximately 650 MegaByte per day.

Converter

The log files generated are in PeopleSoft format (*.tracesql), and they have to be converted to process logs which are in common XML format (Appendix A). It is not straightforward to convert the PeopleSoft Financials logs to Process Mining logs, because they are in a different format and do not directly contain Process Mining information. They have to be generated based on the information found in the log file. In other words, we have to process each log file to obtain the information we need to write the common process log format.

By default, there is no clear Case ID available in the logs. Therefore, we are not able to say that a particular page has been executed for a particular case. However, we can work around this problem when we look more closely at the pages of one component. When a page is accessed, then all the pages, and its corresponding data is loaded into a buffer. The value of the data within this buffer can be used as a Case ID, because for a different process, the value of the buffer changes.



Figure 4.11: ProMimport screenshot with PeopleSoft filter.

The PeopleSoft pages converter, depicted in Figure 4.11, reads the individual

PeopleSoft logs and writes them to a common format log in XML format. The converter uses the ProMImport architecture [24]. This architecture offers log writing functionality, database connectivity and a graphical user interface for extracting logs from databases. The converter only uses the log files for conversion. The PeopleSoft converter is a compiled Java Archive (JAR) file⁴ and can be imported into the ProMimport tool. The tool also allows different converters to be imported into the basic ProMimport interface. Converters are called filters in the ProMimport tool. For every filter, there are general features such as start the conversion process, abort and reset the filter settings. The setting of the individual filter is specific to the filter.

Result:

The prototype converter for the approach was built to test whether the approach is correct. The assumption made for this approach is not correct, because the result was not what we expected. A page is not on the same level as the business process task, and therefore we do not consider this approach to be correct.

Evaluation:

This approach works only for pages within one component. A page is the smallest logic unit of work and therefore we assumed as one task. For the adapter that was built we use the data between pages within one component as the case ID. The result is a process model of pages which is limited to one component and therefore not interesting, because pages within a component are predefined and only cover a single step in a business process. We wish to obtain process models of business processes. Therefore, the next approach tries to look at the information available between components (see Section 4.4.2).

4.4.2 PeopleSoft components

Motivation:

From the pages approach, we observe that pages within components also are accessed by other pages belonging to other components. As the pages within a component are a number of steps in the business process, the different steps together must form the business process. In other words, a PeopleSoft page is a small task, a PeopleSoft component is a business process task.

For this approach, we assume that one component is a task and the case is the order in which components are accessed.

Action:

When we try to obtain this logic between components for a process, we notice that there is no generic information shared. Unlike pages within a component, the data shared between two components does not match, and generically, we cannot identify which components belong to the same process instance using the data.

When we look more closely to the components and how users are instructed to

⁴<http://java.sun.com>

work with PeopleSoft Financials components, we are surprised that there is no logic defined in the system. In other words, the process model that defines how users should work with components is not defined in the system. Users are unrestricted in the components that they can access. Furthermore, the security model of PeopleSoft Financials only defines which users are permitted to access certain components, but not the logic. The process model, and moreover the case id information, is inside the minds of the users and thus not very helpful for mining. We are not able to obtain a process model based on the components accessed by users, because the data shared between accessed components does not provide enough information about the process (i.e, the Process Mining requirements can not be met).

Result:

Unfortunately, the approach that we used for PeopleSoft pages cannot be used for PeopleSoft components. This is because, the data shared between two components is not fixed beforehand, and are not able to generalize how two components communicate. Based on this approach, we are not able to identify the case id in the PeopleSoft Financials system.

Evaluation:

We can conclude from this approach that components within PeopleSoft Financials are smaller processes responsible for one business process task. In order to execute a business process, a user hops from components to other components. A group of pages fulfills a business process task. By default, no process model is defined in the system that restricts which components have to be followed by others to complete a process. Therefore, we cannot identify the process based on the logs, because it is not available.

4.4.3 User oriented results

Although the approach for mining PeopleSoft pages seemed promising, the process model that resulted from the approach is a small process step. Mining the components is the real solution. However, the approach used for the pages, cannot be reused, because the data shared between components is not generically the same. Therefore, we cannot use it to identify the Process Mining requirements.

4.5 Conclusion

The goal of this research is to answer the question: *“Can we create an adapter to convert the logs of the PeopleSoft Financials ERP system to the common log format defined by the Process Mining tool?”* For the application of Process Mining we have already mentioned that we need the information about the executed processes and the order in which activities for a process are executed.

The approach taken to answer the questions leads us to many different types of approaches. Unfortunately, we were not able to find the Process Mining requirements in all the approaches. The attempted approaches with respect to parts of the PeopleSoft Financials system architecture is given in Table 4.1.

If there is an “X” in one of the columns, then the row with that approach has focused on that part of the system architecture to find the Process Mining data.

Table 4.1: Approaches vs System architecture

Approaches	Web Browser	Web Server	Application Server	Database
Application mining			X	
Process scheduler			X	
Foreign keys				X
SQL triggers				X
Table mining				X
Page mining	X	X		
Component mining	X	X		

Finding the case id information is troublesome in PeopleSoft Financials. However, for the page mining approach, we were able to identify the Process Mining requirements. The process model that we obtained is a single step in the business process. Although, these PeopleSoft pages describe the smallest logical unit of work, i.e. a workflow task [4], the tasks only cover a part of a business process task. Example of pages that together form one business process task is the entering of a new invoice in the system.

Based on the system architecture of PeopleSoft Financials and cross-referenced with the attempted approaches, we have looked at a significant number of places within the system and are quite sure that obtaining a process model generically is not possible. The reason we are not able to build a converter for PeopleSoft Financials is because, component mining is not possible. This approach is close to the generic approach we seek, but because we lack common information between all components, it is not possible.

Therefore, the assumption made by [7] that for any PAIS that is subjected to Process Mining, there is a process log available, containing the order of events taken place, is not valid. Neither can we give a generic approach to apply Process Mining to PeopleSoft Financials at Fontys. We conclude that, based on the attempted approaches, we are not able to create an adapter to convert the logs of the PeopleSoft Financials at Fontys to the common Process Mining log format. In the next chapter we discuss whether the answer holds to all PeopleSoft Systems or to only PeopleSoft Financials.

Chapter 5

Generic vs. Specific

5.1 Introduction

In this chapter, we take the opportunity to evaluate how generic the answer given in the previous chapter is. *Is Process Mining in PeopleSoft generically possible?* Therefore, in Section 5.2, we discuss whether some of the attempted approaches would yield the result we are looking for, and why not. From these approaches, the *component mining* approach related to the “user end” of PeopleSoft Financials seemed closest to what we want. In that respect we describe, whether component mining is possible, in general, for other versions of PeopleSoft Financials, different PeopleSoft systems and even different implementations of Financials.

In Section 5.3, we discuss if there are possibilities to obtain process models alternatively from the knowledge we have gained about the PeopleSoft Financials system at Fontys. These alternative solutions are specific for the implementation of PeopleSoft Financials at Fontys, but interesting from the Process Mining perspective.

5.2 Generic

We take this opportunity to reflect on the component mining approach discussed in the previous chapter. We were not able to implement this approach for PeopleSoft Financials version 8.4, but how about other systems? We give some remarks based on the knowledge gained from analyzing PeopleSoft Financials at Fontys.

5.2.1 Other versions

The approaches attempted are based on PeopleSoft Financials version 8.4. According to PeopleSoft themselves, all versions after 8.0 are based on the same system architecture, as discussed in Section 3.6, and of which currently the latest version of PeopleSoft Financials is version 8.9. Versions of PeopleSoft Financials before 8.0 have a totally different architecture. In version 8.0 and above, users can connect and use PeopleSoft Financials with their web browsers, whereas, in versions before 8.0, the users have to install client software that gives them access to the PeopleSoft system. The order and access in which they use the system is defined not only by the menu structure, but also by the implementation of authorisations in the system. If we wish to apply Process Mining to PeopleSoft Financials systems before

version 8.0, by using the idea of the component mining approach, we need to focus on the menu structure of users. PeopleSoft pages and components are not available in versions before 8.0, because there is no web server to host these applications. So, once the client software is installed, authorisations forbid certain menu options to users. These menu options are the “pages and components” in the later versions of PeopleSoft. Once again, these menu structures do not enforce a workflow. Therefore, a generic approach to PeopleSoft Financials versions before 8.0, shows similar problems to that of versions after 8.0 and no solution could be found.

5.2.2 Other implementations

For other implementations of PeopleSoft Financials where for example, the underlying database supports foreign keys, it is possible to find relationships between tables in the database. This could be done by using the foreign key approach, described in Section 4.3.1. However, when we create a database model that describes which tables are used for PeopleSoft Financials and how they are linked together, we cannot abstract to tasks that triggered these database transactions. From the system alone, we cannot identify which activities (and database transactions) are part of a process. Finding a generic solution to PeopleSoft Financials that is possible in all implementations is doubtful, because if there were such a solution, we would have found it. Therefore, we can conclude that a generic method to identify the Process Mining requirements in other PeopleSoft Financials implementations cannot be found.

5.2.3 Other systems

PeopleSoft has over the years acquired many other systems from best-of-breed companies¹. Their systems do not always have the same system architecture. Recall that the system architecture in PeopleSoft Financials is abstractly divided into four pieces: Web clients, Web server, Application Server, and Database. Therefore, we are not able to foresee whether Process Mining can be applied “generically” to other systems.

Currently at Fontys, they have three different PeopleSoft systems in use. PeopleSoft Campus Solutions, PeopleSoft Human Resources and PeopleSoft Financials (F). From the investigations to these other systems, we are not able to process mine generically. When we compared the Financials system architecture with that of Campus Solutions, and Human Resource, we did not find any major differences. The difference between these systems are the components. Different systems offer different components, which in turn, have different database tables. A group of components that can “work together” are called an ERP module or system. Note that when we say “work together”, we only mean that the required database tables are shared for the different components.

As for the users of the other systems, the user is once again not forced to work in a particular way. From the data used by components, we are not able to identify when a component is executed for a case.

However, there are PeopleSoft systems that claim to offer an explicit workflow design

¹PeopleSoft themselves have been recently taken over by ORACLE and this acquisition makes ORACLE the largest competitor to SAP R/3, the ERP system market leader.

for supporting their business processes. PeopleSoft Customer Relationship Management (CRM) is such a PeopleSoft system with an explicit process definition. The processes that will be discovered, using Process Mining, are limited to those processes with case id information. We were not able to analyse the PeopleSoft CRM system for Process Mining, because there was no system available for research at Fontys at this time. Future work can be done to investigate the Process Mining possibilities in PeopleSoft CRM. As for other (future) PeopleSoft systems, we are not able to assess the possibility of applying Process Mining.

5.3 Specific

The goal of Process Mining is to obtain the process models of executed processes. Now that we know that Process Mining in PeopleSoft Financials based on the approaches attempted is not possible. There are specific approaches for Fontys to visualize their business processes. These options are implementation specific, and will not be able to work for other systems, but it is a comprehensible step for the understanding of their processes.

5.3.1 PeopleSoft Business maps

PeopleSoft Financials as for other of PeopleSoft systems² in versions above 8.0 have *business process maps*³. support. Currently, this functionality in PeopleSoft is not used at Fontys. PeopleSoft documentation[14] describes that it is possible to define “workflows” for a business process. A workflow in PeopleSoft terminology is not comparable with the workflow as we know from Workflow Management Systems such as Staffware. The PeopleSoft business maps consist of routing of PeopleSoft pages, and indirectly components. Thereby, it is also possible to add items in a worklist. At design time, the pages that are part of a process are directly linked using hyperlinks. A worklist is a user specific table with work items that have yet to be completed. These work items are not indicated by case id's, but by their actual data. However, this data is not structurally the same and therefore cannot be used as a case id. The actual routing is specified at the PeopleSoft page level in PeopleSoft code. Additionally, triggers can also be added to events that are fired when conditions are met. Therefore, using PeopleSoft business process maps, it is possible to define how processes have to be executed by linking pages and components together. The workflow of the processes is stored in the database and is encrypted. Process mining is complicated for such specific implementations, because there is no case id information available and any logic between components, the workflow, is encrypted in the database.

5.3.2 Customized mining

The assumption that a task id is a page within a component that has been accessed by a user, and the timestamp being when that page was accessed is plausible, but the missing piece of information is the case id. This means that when a page within a component has been executed, we do not know for which case this is.

²Business process maps are also available in PeopleSoft Campus Solutions and Human Resource

³Business process map documentation: <http://eiswbpapb.admin.unt.edu/PSOL/html/doc/eng/psbooks/twfl/book.htm>

An alternative solution is to “manually” create cases by using data values. We make changes in the PeopleSoft system that will help to *create* a process definition. Then add information, at design time, what is required later on to mine.

The problem of component mining is that we are not able to identify how components share data for a process. In other words, we cannot identify whether a component *A* and component *B* have been accessed for a single process instance.

This approach assumes that a business process consists of components linked together to realize a process. By default, when a PeopleSoft component is accessed, it loads all the database fields, tables, and pages belonging to that component. For every component, there is a set of data fields that uniquely identify a component. Assume that we add a field called “case id” to every component. Then, we make sure that when the component is accessed by an user, the component writes a file to a certain location containing information about that component. This information is the name of the component, the artificially created case id, and the timestamp.

Then, define a process, for example billing, and walk through all the components that is used and record the component names. Then, using the PeopleSoft development tools (PeopleTools), check what data is important for each component accessed along this process path. Afterwards, assign a unique case id value at the beginning of the process and pass it along from one component to another.

Then, let users work with the system for a period of time to generate information. Using this generated information, we create our own adapter that converts the output information of components to the common process log format. Then we are able to visualise this using the Process Mining tool.

5.4 Conclusions

For the PeopleSoft systems so far, no generic mining approach could be found. Other versions of PeopleSoft Financials have the same system architecture and do not supply us with a case id or anything similar. As for the database of PeopleSoft Financials, it is not possible to use just the database to mine processes. PeopleSoft applications store some logic within, and the database is solely used to store application configurations.

There are specific approaches that are capable of defining the business process model of the system, such as business process maps. To mine these PeopleSoft business process maps is completely a different challenge. Furthermore, customizing PeopleSoft in order to create an artificial case id at design time, is a starting point. These customizations are specific for an implementation.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

One of the underlying goals of this research is the extension of Workflow mining research, called Process mining, to other systems driven by process models. Process mining is a generic methodology to distil a process description based on a set of real process executions.

We have attempted different approaches for the PeopleSoft Financials ERP system. From all of the approaches discussed, we notice that from the process executions obtainable, it was not possible to find the business process model. Unlike in Workflow management systems, it is not clear how the business process flow is. For this research, no adapter for Process Mining in PeopleSoft Financials could be made, because the set of process executions provided by the system does not contain enough information about the processes required for Process Mining.

Furthermore, in PeopleSoft Financials, we notice that a business process consists of smaller processes. These smaller processes are what we classify as business process tasks. Only when these smaller processes are connected together and executed, can we refer to it as business process execution. Unfortunately, in PeopleSoft Financials the connection between small processes is not defined in the PeopleSoft Financials system. Therefore, we conclude that the default logic, that is the logic that is available in PeopleSoft Financials when it is purchased, only describes how these small processes should operate.

However, this does not imply that there is no process model at all. It just means that business processes are not explicitly defined in the system using (graphical) process descriptions at Fontys. It is up to the organisation that use PeopleSoft Financials to educate users how to work with the system. In this case, Fontys.

Furthermore, PeopleSoft Financials does not enforce workflow through process models, and it is also not visible in the log files. For any process log of PeopleSoft Financials that describes the behaviour of the system, it is complex to find the common data element, which we call case. From our efforts, and the generalisation we make to other similar PeopleSoft systems, it is quite possible to conclude that there is no such information.

Nonetheless, there are possibilities to develop solutions to obtain these process mod-

els, but they are implementation specific and not interesting for the Process Mining research at first. There was not enough time to implement these specific approaches, therefore these different ideas were just introduced.

To conclude, the assumption for Process Mining is that it is possible to record events such that we have information about the order in which the events of a case are executed [7], based on the approaches discussed, does not hold for this system. Therefore, we can answer our research goal. We cannot create an adapter to convert the logs of the PeopleSoft Financials ERP system to the common log format defined by the Process mining tool, because we lack information about the order in which events of a case are executed.

We can also conclude that the “process awareness” of this PeopleSoft Financials system is low at business process level. Nevertheless, software in PeopleSoft ERP system can be reused for different implementations, because at business process level nothing is predefined. It seems that the basic functionality of PeopleSoft, that resides within components, is enough to claim flexibility on a business process level and call itself a PAIS.

6.2 Recommendations

In this section we give some recommendations for the Process Mining research, other research efforts in the application of Process Mining to non case-driven systems. Finally, we give our recommendations to Fontys.

6.2.1 Process mining research

The problem of not being able to discover processes without Case IDs is beyond the scope of this research. However, in [18], they introduce techniques to mine processes without case ids. These techniques use clustering techniques, known from data mining to detect workflows. Specially for ERP systems such as PeopleSoft, where no process information is available, any research work, in obtaining process models without case-driven workflow logs is useful. Some work towards implementing this into the Process Mining framework is useful. Currently, these workflow logs cannot be stored in the common Process Mining log format, because the log files require case ID's. Since the idea of mining without case ids is an alternative approach, it is not available in the Process Mining architecture.

6.2.2 Mining PeopleSoft CRM system

As described, PeopleSoft also offers a CRM system. Future work can be done in analysing the Process Mining possibilities for that system. In any case, the approach towards implementing Process Mining to a system remains the same: (1) find the information you want, (2) extract this information, (3) build a converter that converts the extracted information in the correct format and (4) generate process models using the available ProM tool.

6.2.3 Fontys

For the research, Fontys has provided us an implementation of their PeopleSoft Financials system. Unfortunately, for the “out-of-the-box” system implemented by Fontys, there is no overview of how the processes are supported. Managers have little knowledge about the business processes supported in the system, and have no knowledge about the system. By implementing PeopleSoft business process maps, visual models are given to the manager as feedback, and the system administrators have less work on their hands, because users are restricted in their actions and it leaves a smaller room for errors.

Bibliography

- [1] W.M.P van der Aalst. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 8(1):21–66, 1998.
- [2] W.M.P. van der Aalst. Business Process Management: A Personal View. Business Process Management Journal, 10(2):135–139, 2004.
- [3] 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. Knowledge Engineering, 47(2):237–267, 2003.
- [4] W.M.P van der Aalst and K.M. van Hee. Workflow Manangement: Models, Methods, and Systems. MIT Press, 2002.
- [5] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewki, and A.P. Barros. Workflow Patters. QUT Technical report, FIT-TR-2002-02, Queensland University of Technology Brisbane., 2002.
- [6] W.M.P. van der Aalst and M. Song. Mining social networks: Uncovering interaction patterns in business processes. In International Conference on Business Process Management (BPM 2004), volume LNCS 3080 Springer-Verlag Berlin Heidelberg, pages 244–260, 2004.
- [7] W.M.P van der Aalst and A.J.M.M. Weijters. Process Mining: A Research Agenda. Computers in Industry, 53(3):231–244, 2004.
- [8] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In Sixth International Conference on Extending Database Technology, pages 469–483, 1998.
- [9] J. Cardoso, R. P. Bostrom, and A. Sheth. Workflow management systems and erp systems: Differences, commonalities, and applications. Information Technology and Management Journal, 4(3/4):319–338, July/October 2004.
- [10] M. Chiang, A. Dreiling, and M. Rosemann. Towards an understanding of model-driven process configuration and its support at large. In Submission to ECIS 2005, 2004.
- [11] J.E. Cook and A.L. Wolf. Discovering Models of Software Process from Event-Based Data. ACM Transactions on Software Engineering and Methodology, 7(3):215–249, 1998.

-
- [12] J.E. Cook and A.L. Wolf. Event-Based Detection of Concurrency. In Proceedings of the Sixth International Symposium on the Foundations of Software Engineering, pages 35–45, 1998.
- [13] J.E. Cook and A.L. Wolf. Software Process Validation: Quantitatively Measuring the Correspondence of a Process to a Model. ACM Transactions of Software Engineering and Methodology, 8(2):147–176, 1999.
- [14] PeopleSoft Product Documentation. PeopleTools 8.4 PeopleSoft Workflow, chapter 1, pages 1–9. PeopleSoft Inc., 2002.
- [15] B.F. van Dongen and M.H. Jansen-Vullers. EPC Verification in the ARIS for MySAP reference model database. BETA Working Paper Series WP 142, Eindhoven University of Technology, 2005.
- [16] B.F. van Dongen, A.K.A. 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 (ICATPN 2005), LNCS 3536:444–454, 2005.
- [17] M. van Giessel. Process mining in SAP R/3. Master's thesis, Technische Universiteit Eindhoven, 2004.
- [18] G. Greco, A. Guzzo, L. Pontieri, and D. Saccà. Mining Expressive Process Models by Clustering Workflow Traces. Proceedings of the 8th PAKDD, 2004.
- [19] P. Habela, M. Roantree, and K. Subieta. Flattening the Metamodel for Object Databases. In ADBIS '02: Proceedings of the 6th East European Conference on Advances in Databases and Information Systems, pages 263–276, London, UK, 2002. Springer-Verlag.
- [20] A. Koedijk and A. Verstelle. ERP in bedrijf, chapter 1, page 17. Tutein, Nolthenhuis, 1999.
- [21] A. M Rashid, L. Hossain, and J.D. Patrick. The Evolution of ERP Systems: A Historical Perspective. In Idea Group Publishing, 2002.
- [22] W. Reisig and G. Rozenberg. Lectures on Petri Nets I: Basic Models. Springer-Verslag: Lecture Notes in Computer Science, 1998.
- [23] A. Rozinat and W.M.P. van der Aalst. Conformance Testing: Measuring the Alignment Between Event Logs and Process Models. BETA Working Paper Series WP 142, Eindhoven University of Technology, 2005.
- [24] Website. <http://is.tm.tue.nl/research/processmining>.
- [25] Website. <http://www.graphviz.org/pub/scm/graphviz2/doc/info/lang.html>.
- [26] Website. http://www.oracle.com/applications/peoplesoft/tools_tech/ent/pia/index.html.
- [27] Website. <http://www.pallas-athena.com/>.
- [28] Website. <http://www.sap.com/solutions/business-suite/erp/index.epx>.

- [29] Website. <http://www.staffware.com>.
- [30] A.J.J.M. Weijters and W.M.P. van der Aalst. Workflow mining: Discovering Workflow Models from Event-Based Data. In Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), 2001.

Appendix A

Common format for process mining logs

A.1 Introduction

This appendix chapter describes the format in which the process mining logs have to fulfil in order to be used by the ProM tool. The common format for storing logs is in the *eXtensible Markup Language (XML)* file format.

The “WorkflowLog” element indicates that it is a workflow log with processes. Each “Process” can contain process instances. A “ProcessInstance” has audit trail entries, a process instance is what process mining defines as a case. The “AuditTrailEntry” holds every step of process. The audit trail entry consists of the “WorkflowModelElement” (task id), “EventType”, “Timestamp” and “Originator”. The workflow model element is the name of the task taking place. An event type can hold different values. The most common event type is complete, indicating that the task has been completed. Note that for systems like Staffware other event types have been used [6]. The originator is the resource that starts the audit trail entry. The main idea behind the converter is matching terminologies. Knowing what is required and in which format and finding the data in the system being analyzed. Process mining is primarily concerned with the process flow and not concerned about the data. But the workflow log as defined in Figure A.1 suggests also room for data. By using the available process mining tools, it is possible to export data for other analytical tools as described in Section 2.6.

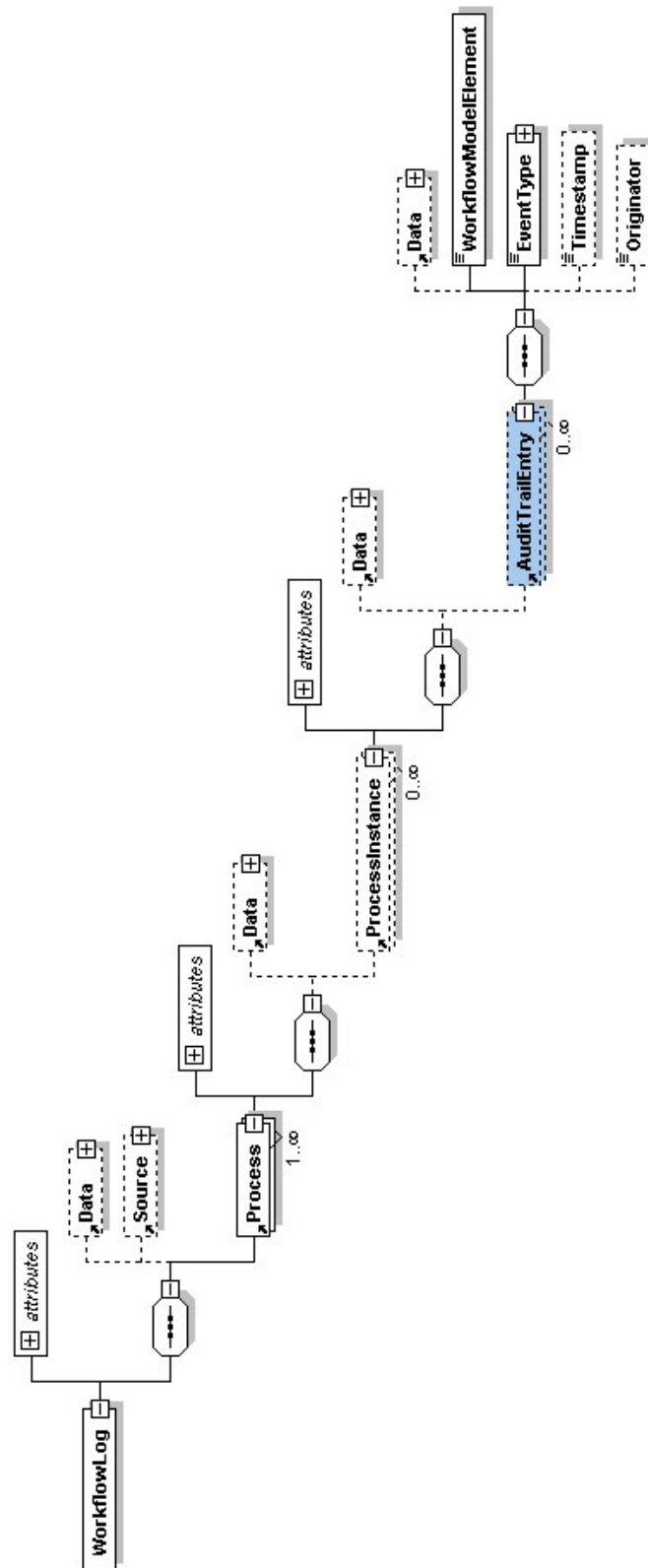


Figure A.1: Common format for process mining logs.

Appendix B

Table mining explained

When we take one of the business processes of Fontys, e.g., the billing of customers, we see that they have to create an invoice. This invoice is created in from the web page, and the entered data is stored in the database. To find out where exactly the invoice is stored, and how, we enable a user trace. User trace is the possibility of tracking changes made to the system by the user. Figure B.1 shows the login screen of PeopleSoft Financials with trace options.

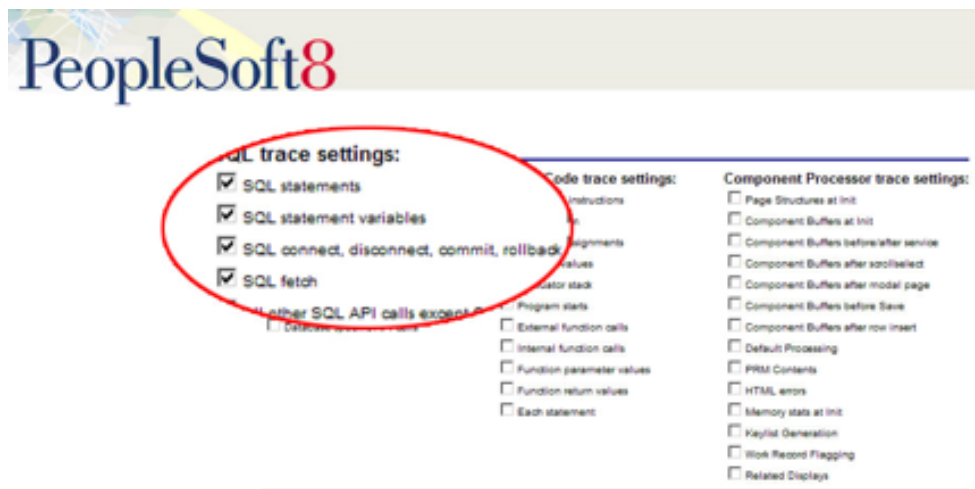


Figure B.1: PeopleSoft login with trace functionality.

The circle in the figure above zooms on the Structured Query Language (SQL) within PeopleSoft applications to modify the underlying database. The user trace settings we have checked are capable of tracing SQL statements, statement variables, SQL connect, disconnect, commit, rollback operations and SQL fetch commands. We first:

1. login into the system with these trace options active;
2. and then enter a new bill into the system;
3. then view the corresponding tracesql file.

From the trace file we know which tables store the bill information. Table B.1 shows the part of the trace file where the tables concerning billing are updated. Note that the first few columns are not been displayed because they are not relevant information.

Table B.1: Table inserts after a bill entry

Nr	Statement	Description
1	COM Stmt=INSERT INTO PS_BI_HDR_VAT(...)	Table for tax
2	COM Stmt=INSERT INTO PS_BI_HDR_AR(...)	Accounts receivable (debiteuren)
3	COM Stmt=INSERT INTO PS_BI_LINE_TAX(...)	Tax scale per item (line)
4	COM Stmt=INSERT INTO PS_BI_LINE(...)	Every item in the bill
5	COM Stmt=INSERT INTO PS_BI_HDR(...)	Billing record
6	COM Stmt=INSERT INTO PS_BI_HDR_CRCARD(...)	Creditcard billing information

As can be seen from Table B.1, a bill is stored in six different tables. The primary key for these six tables are the same. The most important table is the PS_BI_HDR. This table stores the heading information of the bill. When we use PeopleSofts Application Designer to view the definitions of the table we get Figure B.2.

The image shows two screenshots of the PeopleSoft database table definition for BI_HDR. The top screenshot shows fields 1 through 26, with the first three fields (BUSINESS_UNIT, INVOICE, and BILL_TO_CUST_ID) circled in red. The bottom screenshot shows fields 170 through 192, with the PROCESS_INSTANCE field circled in red.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	BUSINESS_UNIT	Char	5	Upper	Unit	Business Unit
2	INVOICE	Char	22	Upper	Invoice	Invoice
3	BILL_TO_CUST_ID	Char	15	Upper	Bill To	Customer
4	BILL_STATUS	Char	3	Upper	Status	Bill Status
5	CONTRACT_NUM	Char	25	Mixed	Contract	Contract
6	INVOICE_TYPE	Char	3	Upper	Inv Type	Invoice Type
7	CONSOL_HDR	Char	1	Upper	Consol Hdr	Consolidated Hdr?
8	CONSOL_KEY	Char	22	Upper	Consol Key	Invoice Consolidation Key
9	CONSOL_SETID	Char	5	Upper	Setid	Consolidation Customer SetID
10	CONSOL_CUST_ID	Char	15	Upper	Cust ID	Consolidation Customer ID
11	CONSOL_BUS_UNIT	Char	5	Upper	Unit	Consolidation Business Unit
12	CONSOL_INVOICE	Char	22	Upper	Consol Inv	Consolidated Invoice
13	ORIGINAL_INVOICE	Char	22	Upper	Original	Original Invoice
14	PRIOR_ADJ_INVOICE	Char	22	Upper	Adjusted	Adjusted Invoice
15	NEXT_ADJ_INVOICE	Char	22	Upper	Adjusting	Adjusting Invoice
16	LATEST_INVOICE	Char	22	Upper	Latest	Latest Invoice
17	ADJUSTED_FLAG	Char	1	Upper	Adjusted	Adjusted Flag
18	BILL_TYPE_ID	Char	3	Upper	Bill Type	Bill Type Identifier
19	BILL_SOURCE_ID	Char	10	Upper	Source	Bill Source
20	BILL_CYCLE_ID	Char	10	Upper	Cycle ID	Billing Cycle Identifier
21	BILL_BY_ID	Char	10	Upper	Bill By ID	Bill By Identifier
22	HDR_FIELDS_KEY	Char	30	Upper	Hdr Fields Key	Header Fields Key
23	BILLING_FREQUENCY	Char	3	Upper	Frequency	Billing Frequency
24	TEMPLATE_IVC_FLG	Char	1	Upper	Template	Template Invoice Flag
25	TEMPLATE_INVOICE	Char	22	Upper	Template	Template Invoice ID
26	RECURRING_START_DT	Date	10		Start	Starting Date
	RECURRING_END_DT	Date	10			Ending Date
	RECURRING_NUM	Char	1			Recurring Numbers

170	ENTRY	Upper		Lang Cd	Language	Entry Event
171	REIMB_AGREEMENT_NUM	Char	25	Mixed	Reimb Agr Num	Reimbursable Agreement Number
172	USER_AMT1	Sign	23.3		Amount 1	User Amt 1
173	USER_AMT2	Sign	23.3		Amount 2	User Amt 2
174	USER_AMT1_BSE	Sign	23.3		Amount 1	User Amt 1
175	USER_AMT2_BSE	Sign	23.3		Amount 2	User Amt 2
176	USER_AMT1_XEU	Sign	23.3		Amount 1	User Amt 1
177	USER_AMT2_XEU	Sign	23.3		Amount 2	User Amt 2
178	USER_DT1	Date	10		Date 1	User Date 1
179	USER_DT2	Date	10		Date 2	User Date 2
180	USER1	Char	1	Upper	User 1	User 1
181	USER2	Char	1	Upper	User 2	User 2
182	USER3	Char	1	Upper	User 3	User 3
183	USER4	Char	1	Upper	User 4	User 4
184	USER5	Char	1	Upper	User 5	User 5
185	USER6	Char	1	Upper	User 6	User 6
186	USER7	Char	1	Upper	User 7	User 7
187	USER8	Char	1	Upper	User 8	User 8
188	USER9	Char	1	Upper	User 9	User 9
189	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance
190	ADD_DTTM	DtTm	26		Dtme Add	Date Time Added
191	LAST_UPDATE_DTTM	DtTm	26		Last Updt	Last Update Timestamp
192	LAST_MAINT_OPRID	Char	30	Upper	Last OPRID	Last Maintained By Operator ID

Figure B.2: Billing Header table.

The table definition describes the fields of the table BI_HDR. Note that in PeopleSoft tables are known as records. The table has in total of 192 fields which corresponds to columns in the database. The first two fields: BUSINESS_UNIT and INVOICE are the primary keys for this table. The other fields are not that interesting except the PROCESS_INSTANCE field. Remember that this field is the primary key for another approached we already discussed namely, the Process Scheduler. When we visualize the link between the two tables we obtain the following picture (Figure B.3).

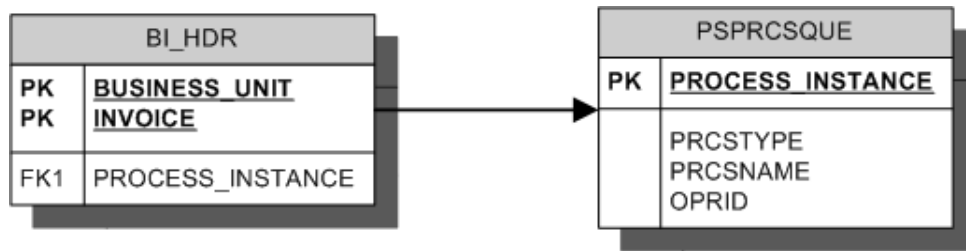


Figure B.3: PS_BIHDR and PSPRCSQUE link.

The foreign key link between the billing table and the Process Scheduler table is an application link and not as the figure is indicates, a foreign key. The PeopleSofts database does not support foreign keys, only primary keys and alternate search keys. The link between these two tables is “hardcoded” in the applications. When an application want to perform changes on a bill, the application concerned, issues a PROCESS_INSTANCE number to that tuple of the table and schedules the tuple for processing in the Process Scheduler table. After the application has executed the PROCESS_INSTANCE field in the BI_HDR table is reset to zero. In other words, information about which application has just executed on a certain bill is lost. Therefore we still can not find out which applications mutate the bill.

Appendix C

PeopleSoft Security

C.1 Users and Roles

The PeopleSoft components and pages available on the internet browser are different per user. Users access the system with their own user profile and password. The system checks the user profile and password using the PeopleSoft Financials database. After a successful login the components that the user is permitted to use are displayed on the screen. To each user profile, one or more PeopleSoft roles can be defined (e.g. employee, manager) and based on the role a user has, a list of permitted components are defined. In the other words, the access to PeopleSoft Financials is based on permission lists, roles and user profiles¹.

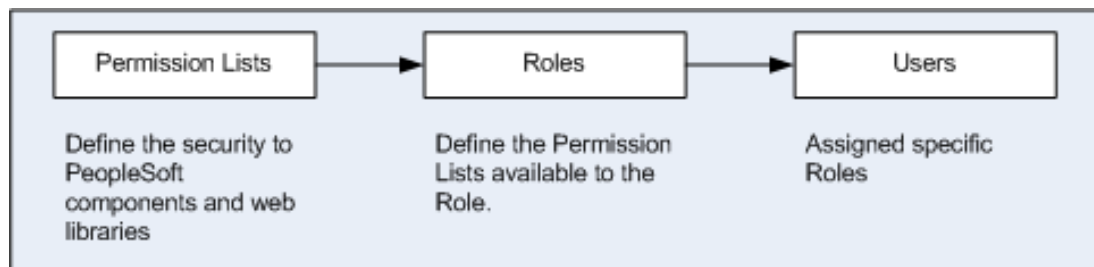


Figure C.1: PeopleSoft roles and permissions.

Figure C.1 describes the roles and permission structure. The user code (i.e. the user profile in PeopleSoft), is unique to one user. An example of the security within PeopleSoft is given in Figure C.2 PeopleSoft Security.

¹<http://eiswbpapb.admin.unt.edu/PSOL/htmldoc/eng/psbooks/tsec/book.htm>

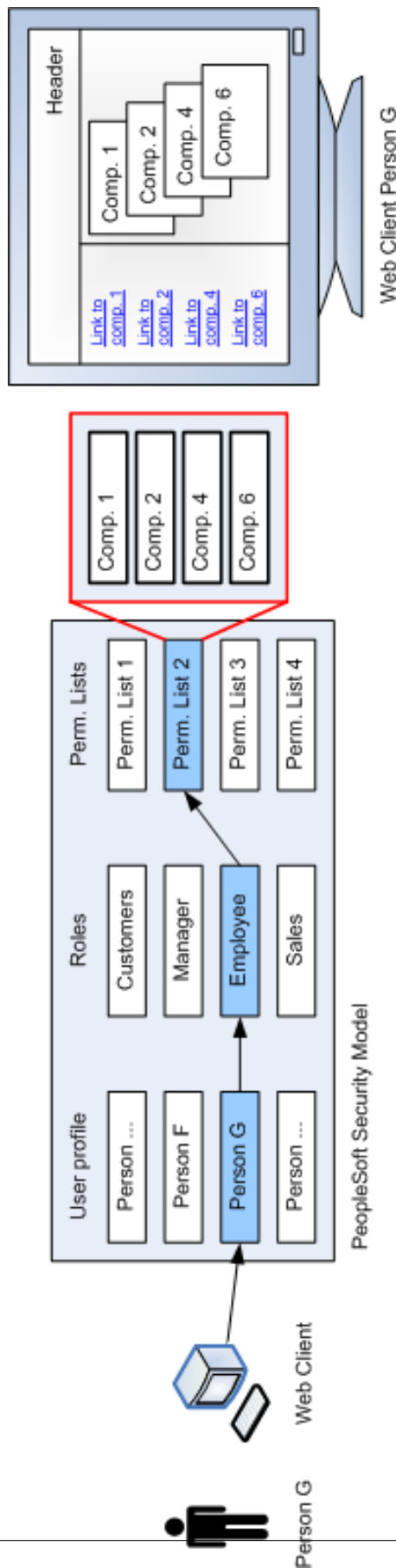


Figure C.2: PeopleSoft security.

Person G in Figure C.2 is an Employee of Organization X. When he enters his user name and password, the roles that belong to that user are obtained from the system. In this case, the user's role is "Employee" and the Permission list belonging to that Role is queried from the database. Note that a user, like Person G can have different roles, which also means he has different permission lists. Eventually, he is also allowed to access different components. Also note that when the list of components that are accessible is known, the PeopleSoft system ensures that the content of person G (which we represent as the monitor on the right extreme), is accessible.

In general, a Process Mining task is one page of a component in PeopleSoft Financials. Users navigate through their applications by clicking on hyperlinks to components and pages. These components and pages eventually modify data in the database. Using the components and pages structure available per user and the role that belongs to that user, we have a collection of possible tasks of users. If the logs files generated by the PeopleSoft system contain this information, and the link between the tasks, we can use it to mine processes.

Appendix D

Process mining explained

D.1 Introduction

To clarify what process mining means, we start with the history of process mining in Section D.2. The initial research done by Cook and Wolf was expanded to Workflow management systems and was called workflow mining, which is described in Section D.3. In Section D.4 we continue with an example of the actual process mining and explain some basic concepts of the mining process. In Section D.5 we conclude the chapter with the conclusions.

D.2 History of process mining

Process mining is certainly not new, Cook and Wolf have investigated process mining issues in the context of software engineering processes [11]. They describe three methods for process discovery: one using neural networks, one using a purely algorithmic approach, and one Markovian approach. The latter two approaches seem the most promising according to Cook and Wolf. The purely algorithmic approach builds a finite state machine. The states are merged when their paths lead to the same state in the future. The Markovian approach uses a mixture of algorithmic and statistical methods and is able to deal with noise. Noise, in this case, can be data that is not complete, or, data of other processes that is being analyzed. Therefore, this noise can be skipped while processing.

The approach given in [11] is initially limited to sequential process behaviour. Cook and Wolf then extended their work for concurrent processes by proposing specific metrics (entropy, event type counts, periodicity, and causality) in [12]. Using these metrics they had discovered models (using Petri net notation) from event streams, but no approach to generate the actual process models was given.

Furthermore, Cook and Wolf introduce a measure to quantify discrepancies between a process model and the actual behavior as registered, using event-based data [13]. These event-based data are similar to runtime data collected for process mining. The idea of applying process mining in the context of workflow management was first introduced in 1998 [8]. This application, called workflow mining, is described in section D.3.

D.3 Workflow mining

For the past ten years there has been a growth in the application of workflow management technology in the field of information systems (IS). Usually, when organizations use information systems, their focus is on applications and less attention is paid to the underlying business process. By using workflow management technology, hidden business processes are highlighted, and the accompanying workflow management tools also support the business process. Workflow management technology is driven by workflow models, and without them, it is not possible to operate. Workflow management systems such as Staffware, IBM MQSeries, COSA, offer generic modeling and enactment capabilities for structured business processes. By making graphical process definitions, i.e. models describing the life-cycle of a typical case (workflow instance) in isolation, one can configure these systems to support business processes [30]. Many other software systems, besides workflow management systems, have adopted workflow technology. Contrary to claims of workflow systems being flexible, many problems are encountered when applying workflow technology. As indicated by many authors, workflow management systems are inflexible and have problems dealing with change [30]. One of the problems is that these systems require a workflow design, i.e., a designer has to construct a detailed model accurately describing the routing of work [3]. Modeling a workflow requires deep knowledge of the business process at hand. The traditional approach towards workflow design and enactment, compared with workflow mining, is shown in the workflow life cycle in Figure D.1.

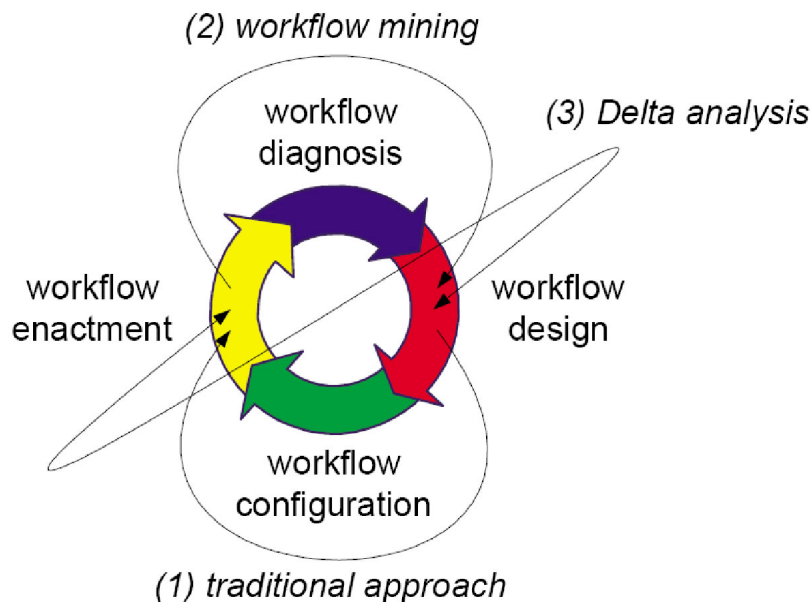


Figure D.1: The workflow life cycle used to illustrate workflow mining and delta analysis in relation to traditional workflow design

The workflow life cycle, in Figure D.1, consists of four phases:

A. Design phase

In the traditional approach the design phase is used for constructing a workflow

model. A workflow model is a diagram that comprises cases, using a graphical representation language (e.g., Petri net [22]). Typically, this is done by a business consultant and is driven by ideas of management on improving the business processes.

B. Configuration phase

Once the design is finished, the workflow management system needs to be configured as specified in the design. This happens in the configuration phase. Note that one has to deal with the limitations and particularities of the workflow management system being used [5].

C. Enactment phase

In the enactment phase, cases (i.e., workflow instances) are handled by the workflow system as specified in the design and configured in the configuration phase.

D. Diagnosis phase

It is possible to collect diagnostic information from a running workflow which is analyzed in the diagnosis phase. This diagnostic information serves as input for the (re)design phase and is given by the diagnosis phase, thus completing the workflow life cycle. In process mining terminology, the diagnostic information is called workflow log.

In the traditional approach the focus is on the design and configuration phases. Not many organizations pay attention to the enactment phase and only few organizations systematically collect runtime data which is analyzed as input for redesign (i.e., the diagnosis phase is typically missing). This diagnosis information is usually used by the organization for solving design problems within the system.

The goal of workflow mining is to use the data collected at runtime to support the design and analysis. It is possible to use run-time collected data (i.e., workflow logs) to derive a model explaining the events recorded. The model can then be used for the diagnosis and the (re)design phase. Workflow mining, in contrast to normal workflow design is objective because it is based on the actual things that have taken place and unlike normal design which is to a larger extent based on perceptions, opinions and is therefore more subjective.

Delta analysis is now possible because one can closely monitor the events taking place and we are capable of detecting differences between the constructed design in the design phase and the action execution registered in the enactment phase. Workflow mining results in a post-design process model that can be compared with the pre-design model, moving the workflow technology into the direction of more operational flexibility [3].

The topic of workflow mining is related to many management trends such as Business Process Reengineering (BPR), Business Intelligence (BI), Business Process Analysis (BPA), Continuous Process Improvement (CPI), and Knowledge Management (KM). Workflow mining can be seen as part of BI, BPA, and KM trends. Moreover, workflow mining can be used as input for evolutionary change of activities. Workflow mining seems to be more appropriate for BPR than for CPI. Also note that workflow mining is not a tool to (re)design processes.

The concept of workflow mining is to understand what is really going on in an organization and finding differences between pre-designed model and actual enactment

model. Despite the fact that workflow mining is not a tool for designing processes, it is evident that a good understanding of the existing processes is vital for any redesign effort [3]. In other words, workflow mining is a technique to assist Business Process Redesign (BPR) activities by using transactional data of events, recorded during the enactment phase of a common workflow lifecycle.

Workflow mining is the mining of workflows from Workflow Management Systems (WfMS) and process mining is the mining of systems that are driven by process models. Process mining does not focus on the mining of workflow systems alone, i.e. systems that require a workflow design to operate, but all systems that work with a process model.

D.4 Process mining example

To illustrate the principles of process mining, consider the log shown in Figure D.2. Note that the example is taken from [3].

case identifier	task identifier	timestamp (date:time)
case 1	A	08-05-2002 : 08:15
case 2	A	08-05-2002 : 08:24
case 3	A	08-05-2002 : 09:30
case 2	B	08-05-2002 : 10:24
case 5	A	08-05-2002 : 10:24
case 4	A	08-05-2002 : 10:25
case 3	B	08-05-2002 : 10:26
case 1	F	08-05-2002 : 11:45
case 4	B	08-05-2002 : 11:46
case 2	C	08-05-2002 : 12:23
case 2	D	08-05-2002 : 15:14
case 5	F	08-05-2002 : 15:17
case 3	D	08-05-2002 : 15:19
case 1	G	08-05-2002 : 16:26
case 4	C	08-05-2002 : 16:29
case 5	G	08-05-2002 : 16:43
case 3	C	09-05-2002 : 08:22
case 4	D	09-05-2002 : 08:45
case 3	E	09-05-2002 : 09:10
case 4	E	09-05-2002 : 10:05
case 2	E	09-05-2002 : 10:12
case 2	G	09-05-2002 : 10:46
case 3	G	09-05-2002 : 11:23
case 4	G	09-05-2002 : 11:25

Figure D.2: A process log

The log in Figure D.2 represents data taken from an arbitrary PAIS. Each line of the log represents an event (i.e. the combination of case id, task id and timestamp). Furthermore, the log contains information about five executed process instances (cases). The log shows that for two cases (1 and 5), the tasks A, F and G have been executed. For case 2 and case 4 the tasks A, B, C, D, E and G have been executed. For case 3, the same tasks have been executed as case 2. However, the tasks C and D are swapped. Each case starts with the execution of A and ends with the execution of G. Task C and D occur in parallel.

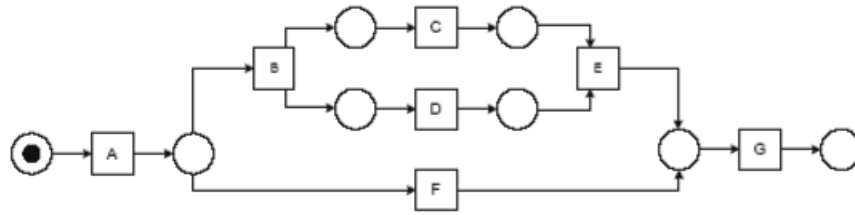


Figure D.3: A process model

Based on the information shown in Figure D.3, and by making some assumptions about the completeness of the log (i.e., assuming that the cases are representative and a sufficient large subset of possible behaviors is observed), one can deduce for example the process model shown in Figure D.3. The model is represented in terms of a Petri net. The Petri net starts with task A and finishes with task G. These tasks are represented by transitions. More detailed explanation about what the boxes and circles mean is given in [1, 22]. After executing A, there is a choice between either executing B, or executing F. After executing B, tasks C and D are executed in parallel, followed by E. Finally, after executing either E or F, G can be executed. To execute C and D in parallel, task B corresponds a so-called AND-split and task E corresponds a so-called AND-join¹. Note that for this example it is assumed that if two tasks appear in any order then they are in parallel. By distinguishing between start events and end events for tasks it is possible to explicitly detect parallelism. The proof of this given in [30].

D.5 Conclusion

In this chapter we have discussed the history of Process Mining. Cook and Wolf were one the first to experiment with obtaining process models from executed processes. Van der Aalst et al., applied the idea of Process Mining in the field of BPM. Soon after, the term Workflow Mining was used to apply Process Mining techniques to WfMS such as Staffware and FLOWER. Finally, in the example, the basics of Process Mining are explained.

¹AND-split and AND-join are not standard Petri-net notation, and have been taken from [4].

Appendix E

Inventory of logs

This section describes the results gathered from analyzing PeopleSofts trace settings. After the introduction to PeopleSoft logs, we continue with the two different types of logging and what kind of information these traces provide us. We conclude the chapter with our results and conclusions.

E.1 Introduction

Within PeopleSoft Financials version 8, there are different types of log files. Log files are called trace files in PeopleSoft. The PeopleTools version we are working with is 8.4, running on an Oracle 8i database. Traces are used to monitor the system and when the system behaves undesirably, the trace file can be used to locate the error. Database administrators use the trace options regularly to solve errors. Although trace files record the activity taken place between the ERP application and the database, it does not contain information on business process level. It simply describes data that is used by the server or user. The next section describes the traces of a user. The other section describes the trace information of the server.

E.2 User trace

It is possible to log different things of a PeopleSoft user. These are divided into three categories: SQL trace, PeopleCode trace and Component Processor trace settings. Any of these trace options can be enabled manually. Figure E.1 shows these trace options.

SQL trace settings:	PeopleCode trace settings:	Component Processor trace settings:
<input type="checkbox"/> SQL statements	<input type="checkbox"/> Evaluator instructions	<input type="checkbox"/> Page Structures at Init
<input type="checkbox"/> SQL statement variables	<input type="checkbox"/> List program	<input type="checkbox"/> Component Buffers at Init
<input type="checkbox"/> SQL connect, disconnect, commit, rollback	<input type="checkbox"/> Variable assignments	<input type="checkbox"/> Component Buffers before/after service
<input type="checkbox"/> SQL fetch	<input type="checkbox"/> Fetched values	<input type="checkbox"/> Component Buffers after scrollselect
<input type="checkbox"/> All other SQL API calls except SSBs	<input type="checkbox"/> Evaluator stack	<input type="checkbox"/> Component Buffers after modal page
<input type="checkbox"/> Set select buffer calls (SSBs)	<input type="checkbox"/> Program starts	<input type="checkbox"/> Component Buffers before Save
<input type="checkbox"/> Database-specific API calls	<input type="checkbox"/> External function calls	<input type="checkbox"/> Component Buffers after row insert
	<input type="checkbox"/> Internal function calls	<input type="checkbox"/> Default Processing
	<input type="checkbox"/> Function parameter values	<input type="checkbox"/> PRM Contents
	<input type="checkbox"/> Function return values	<input type="checkbox"/> HTML errors
	<input type="checkbox"/> Each statement	<input type="checkbox"/> Memory stats at Init
		<input type="checkbox"/> Keylist Generation
		<input type="checkbox"/> Work Record Flagging
		<input type="checkbox"/> Related Displays

Figure E.1: PeopleSoft 8 trace options

These options are available before a user has logged into the system (part of login). In the subsections that follow we describe each category of trace options. We are not bothered about the technical value but the process mining value.

E.2.1 SQL trace settings

As the trace category says, the SQL trace settings option logs everything that has to do with the communication between the ERP system and the database. Applications within ERP systems usually are embedded with SQL code to query, or modify data. The information we find when analyzing the trace file with these trace settings is on the database level. From the SQL trace data we can not identify process tasks. However, we can see the executed queries and the values used for the queries. The SQL trace setting is used for the logging of SQL code that is invoked while using PeopleSoft Financials. However, these trace options do not provide us with information regarding the business process. The trace file contains database operations and with this information we can not obtain the activity that caused the SQL code to be executed. We are interested in those activities and the order in which those activities are performed.

E.2.2 PeopleCode trace settings

PeopleCode is the proprietary language used by PeopleSoft applications. It resembles other programming languages. However, many aspects are unique to the language and the PeopleTools environment. The trace settings of the PeopleCode provide information about the application, but not to what started the application. Usually, a set of applications are started by an activity.

E.2.3 Component Processor trace settings

Since version 8 of PeopleSoft, there is no client software required to use PeopleSoft. Users access the system using their web browsers. The web browser displays components which comprise many pages. Pages are in turn views of records of the database.

With the component processor trace settings we can identify which components and

pages have been accessed by the end-user. This is done by checking the “Page structures at Init” checkbox, from the Component Processor trace settings. The results are the following lines that have been added in the trace file:

```
1      Get : Remote PRM(BI_ENTRY/GBL/ENG)u1.p0.v274
2      Page Constructed
3      Component=BI_ENTRY; Market=GBL; Page=BI_PNLSERIES_WRK;
Menu=ENTER_BILLING_INFORMATION Process flags=50103000;
Mode=0001
4      Keylist
5          BILL_SOURCE_ID='TEAM_A'
6          BILL_TO_CUST_ID='10043'
7          BILL_TYPE_ID='ALG'
8          BUSINESS_UNIT='00025'
9          INVOICE='NEXT'
10     Begin Component
11     Page BI_PNLSERIES_WRK IdxCount=30
12     Page COMBO_EDIT_WRK IdxCount=18
13     Page BI_HDR IdxCount=213
14     Page BI_HDR2 IdxCount=112
15     Page BI_HDR_ADDR IdxCount=102
16     Page BI_HDR_CC IdxCount=109
17     Page BI_HDR_OM IdxCount=62
18     Page BI_HDR_PC IdxCount=57
19     Page BI_HDR_SVC IdxCount=49
20     Page BI_AR_DST IdxCount=96
21     Page BI_LINE IdxCount=96
22     Page BI_LINE2 IdxCount=83
23     Page BI_LINE_TAX IdxCount=90
24     Page BI_LINE_VAT IdxCount=123
25     Page BI_LINE_OM IdxCount=115
26     Page BI_LINE_PC IdxCount=75
27     Page BI_LINE_SVC IdxCount=61
28     Page BI_LINE_MISC IdxCount=65
29     Page BI_LINE_WKS IdxCount=88
30     Page BI_LINE_DST IdxCount=105
31     Page BI_LINE_DFR IdxCount=98
32     Page BI_LINE_AR IdxCount=91
33     Page BI_LINE_UAR IdxCount=99
34     Page BI_LINE_DS IdxCount=94
35     Page BI_LINE_DS_DTL IdxCount=86
36     Page BI_LINE_DS_DFR IdxCount=83
37     Page BI_LINE_IU_EXP IdxCount=91
38     Page BI_LINE_IU_PAY IdxCount=91
39     Page BI_LINE_DST_STA IdxCount=86
40     Page BI_HDR_NOTE IdxCount=53
41     Page BI_LINE_NOTE IdxCount=71
42     Page WF_FUNCTIONS_01 IdxCount=29
```

43 End Component

The line that interests us is line number 3. This line clearly states the component that been accessed by the user. Although there is no indication which user has accessed the component `BIENTRY`, we can obtain the user by looking at the login steps of the trace file which user has logged in. Furthermore, the PeopleSoft Application Server registers all traces per user and the IP address of that user.

What does this component line mean? As you know, components contain pages and pages can activate application, and also SQL. We have a link between user actions and system applications. So this seems useful for the process mining research.

E.3 Application Server trace

The other type of tracing is the tracing of application engine which runs on the application server. We have already analyzed the traces of an application engine and found them to be not useful for the mining of processes. The application server also traces many other things, but these are related to optimization of the application server and not interesting for this research.

E.4 Conclusion

As we are interested how the end user accesses a certain component, we can look at the trace file which has been generated with the “Page structures at Init” checkbox selected to find out which components have been accessed by the user.