Preprocessing Support for Large Scale Process Mining of SAP Transactions

Jon Espen Ingvaldsen and Jon Atle Gulla

Norwegian University of Science and Technology,
Department of Computer and Information Science,
Sem Saelands vei 7-9,
NO-7491 Trondheim, Norway
{jonespi, jag}@idi.ntnu.no

Abstract. Since ERP systems, like SAP, support the backbone operations of companies, their transaction logs provide valuable insight into the companies' business processes. In SAP every transaction is stored and linked to relevant documents, organizational structures and other process-relevant information. However, the complexities and size of SAP logs make it hard to analyze the business processes directly with current process mining tools. This paper describes an ERP log analysis system that allows the users to define at a meta level how events, resources and their inter-relations are stored and transformed for use in process mining. We show how the system is applied to extract and transform related SAP transaction data for the ProM process mining tool.

1 Introduction

SAP is the most widely used ERP system for backbone operations. SAP implementations are configured according to the SAP Reference Model or customized for specific requirements. Even though there may be blue print models defined for how the the systems should support organizational business processes, there are often gaps between how the systems are planned to be used and how the employees actually carry out the operations. To identify these gaps, we need models that reveal how the actual business processes are carried out.

Static blue print models, like the SAP Reference Model, also ignore information about load distributions. This means that a system might be modeled correctly, but for a person reading the model it is not possible to say which parts of the process flows are carried out frequently and which parts are hardly carried out at all. And we cannot know whether the process have been finished within acceptable time limits. To access such information we need to collect historical information about executed process instances.

The functional richness of SAP systems makes the whole SAP Reference Model extensive and complex. Recent research has revealed modeling errors in 5.6% (lower bound) of the Event Process Chains in the SAP Reference Model [1]. As a consequence, the EPC models in the SAP Reference Model are not

A. ter Hofstede, B. Benatallah, and H.-Y. Paik (Eds.): BPM 2007 Workshops, LNCS 4928, pp. 30–41, 2008.

completely representational for how standard SAP systems are implemented on the system level.

Process mining aims at extracting descriptive models from event logs in Enterprise Resource Planning (ERP) or Workflow Management Systems (WfMS) to reconstruct the underlying business process flows[2]. As process mining models are built upon instance data, we are also capable of enriching the models with key performance indicators, load distribution information and other more detailed analyses of the business flows[3][4].

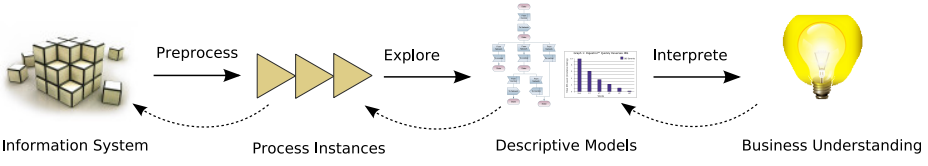

Fig. 1. Process mining project methodology

Figure 1 shows the steps involved in a process mining project. The basis for carrying out a project is a data material that contains event related information fragments. To make use of the raw data material, pre-processing activities are often required before process mining algorithms can be applied. The goal of the pre-processing phase is to extract normalized event logs, a job which involves activities such as data cleansing, feature selection, and merging of distinct data sources.

The output of the pre-processing phase is process or event instances that can be explored through graphs and process- and data mining models. The goal of the exploration phase is to give the user a deeper understanding of his business, which again can be exploited to improve organizational structures and policies. To gain a proper business understanding users typically have to extract several models that describe different perspectives in the process analyses, i.e. control flow, social networks, load distributions, etc.

An important aspect of the process mining projects is that for each phase it might be necessary to return to the previous phase to make improvements or perform additional activities. This makes the nature of process mining projects iterative and interactive.

Some information systems produce event logs that can be fed into process mining algorithms directly with little pre-processing involved. For other systems, the pre-processing is the most time-consuming and work intensive phase.

In this paper, we will describe the Enterprise Visualization Suite (EVS)[1] Model Builder; an application that is designed to support the pre-processing phase of ERP related process mining projects. We will use data from procurement and logistics as examples to show SAP related pre-processing challenges

[1] EVS is a visualization, process- and data mining framework, developed by Businesscape AS (www.businesscape.no).

and show the EVS Model Builder approach to support the work. The output of the EVS Model Builder is identified process instances that are stored as MXML, which is the input format to the open source process mining application ProM[2].

Many SAP tables can be viewed as an event log that individually has some potential for process mining. In this paper, we focus on process mining where process instances are constructed from tracing resource dependencies between executed transactions.

Section 2 describes characteristics of SAP transaction data, and how they create challenges for larger process mining projects. The architecture of the EVS Model Builder and its approach for supporting the pre-processing phase are given in Section 3. Section 4 follows with an discussion of results and alternatives. Related work is described in Section 5, followed by some concluding remarks in Section 6.

2 SAP Transaction Data

SAP contains more than 10000 transaction. Transactions are sub applications of SAP that are accessed by their unique transaction code or through menu hierarchies. It is important to note that transactions do not have a one-to-one mapping to tasks, which means that tasks can be carried out through more than one transaction, which again can incorporate a set of task. A task (also known as function) is an atomic entity describing "what is to be done" in the SAP Reference Model[5].

Transactions carried out in the SAP system store and change data elements in master data and transaction tables. Transaction tables are the largest, since they contain the daily operations data, such as sales orders and invoices. Master data files describe sets of basic business entities such as customers, vendors and users [6]. Most transactions do operations on some resource, typically a document. Documents are most often represented by two tables, one describing header properties and one describing properties for each involved item.

Most dependiencies between SAP documents are stored at the item-level. This makes it possible for one document to depend on a collection of other document resources.

In SAP, each resource operation is logged, also change operations. Most creation and change events on documents are stored in the CDHDR (shown with example data in Table 1) and CDPOS (Change Document Positions) tables.

Master data tables are valuable for accessing textual descriptions of involved business entities and relationships between them. In process mining, events are assigned with a user (originator in MXML). While users in the transaction tables are referred to by their unique and often cryptical username, their full name and department relationship are found in master data sources.

The SAP database also contains ontological information that helps us interpreting the transactions. The two database tables TSTC and TSTCT (shown in Table 2) describe data related to every SAP transtaction. The TEXT attribute

[2] http://sourceforge.net/projects/prom

Table 1. Sample data from the CDHDR (Change Document Header) table

OBJECTID	OBJECTCLAS	CHANGENR	USERNAME	TCODE	UDATE	UTIME
10764301	BANF	33255224	HANSEN	ME51	20030802	144344
00411544	EINKBELEG	33255226	BJARMAN	ME21	20030802	151531
00367081	EINKBELEG	33255227	BJARMAN	ME21	20030803	114030
10445894	BANF	33255243	ANNE	ME51	20030804	090311
00411544	EINKBELEG	33255221	BJARMAN	ME22N	20030804	091740
00411544	EINKBELEG	33255340	TOR	ME22	20030804	123041
00367092	EINKBELEG	33255261	ANNE	ME21N	20030804	130401
04516134	VERKBELEG	33256062	BATMAN	VA02	20030804	135643
74002003	BELEG	33255265	BATMAN	FB02	20030804	150945

Table 2. Sample data from the TSTCT (Transaction Descriptions) table

TCODE	TEXT
MR1M	Post Invoice Document
ME21N	Create Purchase Order
ME57	Assign and Process Requisitions
ME22N	Change Purchase Order
ME52N	Change Purchase Requisition
MIRO	Enter Invoice

in TSTCT gives a full textual description for each transaction. Whenever we see the transaction code ME21N in the logs, we know that this code stands for 'Create Purchase Order'.

A main challenge for doing process mining on the resource-flow between SAP transactions is that there is no defined consistency for how all documents, change events and resource dependencies are stored. Numerous data tables have to be merged, and the data attributes that are interesting for process mining have to be explicitly located in each data table. In the SAP tables there are also missing and faulty data values that must be handled in the pre-processing phase.

To extract process chains that incorporate the creation and changing of purchase requisitions, purchase orders and invoice receipts, and how these events are related to each other, users, departments and transaction descriptions, we have to pre-process information from the EBAN (Purchase Requisitions), EKKO (Purchase Order Header), RBKP (Invoice Receipts), RSEG (Invoice Items), CDHDR (Change Document Header), TSTCT (Textual Transaction Descriptions), and USR03 (User) tables. In larger process mining projects, potentially incorporating elements such as goods movement, goods receipts, sales and delivery orders, shipping documents etc., the pre-processing job would be even more extensive and difficult.

3 EVS ModelBuilder

The EVS Model Builder supports the pre-processing phase of process mining projects by extracting the data elements that are interesting for process mining analysis, constructing process instances and store them as MXML files. As shown in figure 2, the process of constructing process instances consists of three steps:

1. Extraction of business objects and their inter-relations.
2. Extraction of events and their relationships to business objects.
3. Identification of process instances by tracing dependency relationships between events.

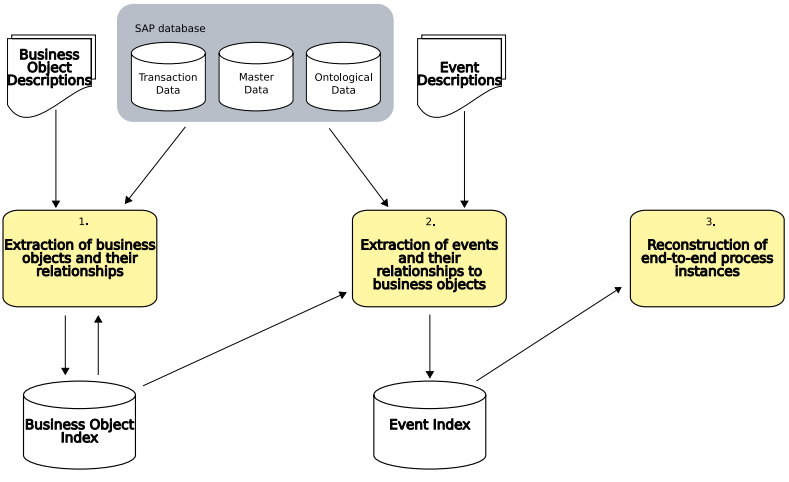

Fig. 2. The three-step-process for constructing process instances. The data flow is represented with arrows.

Business objects are entities that have a valuable (with respect to out process mining analyses) relationship to the business flow. A business object can be a user, department, transaction, document, and other more project specific entities. Business objects are defined by a unique identifier and a textual description (optional).

Events are happenings at a point of time where a set of business objects co-occur. An example of an event is the alteration of a purchase order. Such an event has a certain timestamp and relates a set of business objects, like a user, a purchase order, and a transaction.

Relationships are labeled such that the process instance construction can treat them differently. To trace the dependencies between the events and to reconstruct process instances, we need to know which business objects an event consumes (inputs) and which it produces (outputs). For an alter purchase order event, a purchase order is consumed, altered, and provided as output. Alteration events

consume typically the same resource as they produce. Creation events, on the other hand, uses the consumption resources to produce a new resource. The creation of a purchase order consumes (most often) a purchase requisition, and a purchase order is produced and provided as output.

The EVS Model Builder requires three types of data sources:

1. Transactional data - Information source for constructing event.
2. Master data - The information source for constructing business objects with meaningful names.
3. Ontological data - Information source for interpreting events at the system level.

Together with meta-descriptions, provided by the user, EVS Model Builder knows how to combine these data sources and extract meaningful process intances.

3.1 Defining Element Descriptions

Figure 3 shows a UML class diagram model of how the meta-descriptions are defined. Event and business object descriptions tell us where the information needed to extract their instances exists. Figure 4 shows an meta level description for how business objects, events and their relations can be defined for process mining projects analysing processes related to the creation and chaning of purchase requisitions, purchase orders, and invoice receipts.

In some SAP tables, several types of elements are stored in the same table. CDHDR is one such table, where creation and change events for several document types are stored. Discriminators are used to separate out specific rows of a database table that are of interest to a given element description. A discriminator contains an attribute, an operator (like equals) and a value. For an event description that are only specifying events on purchase requisitions, we can add the discriminator (CDHDR.objectclas = 'BANF').

EVS Model Builder has two ways of describing relationships between event and business object descriptions. Explicit relation descriptions are used to define a connection between two element descriptions, where mapping attributes for both are stated in a single database table. An example of such a relation is between transactions, described in TSTCT, and events in the CDHDR table. Here, we can define an explicit relation description where we map the CDHDR.tcode attribute to the transactions key attribute.

Implicit relations are used to define connection paths that involves several linked tables in the SAP database. An example of such a relation is between purchase orders and creation of invoice receipt events. Here the events are located in the RBKP tables, which have a produce-relation to an invoice receipt (also found in RBKP). These invoice receipts have explicit relations to invoice items found in RSEG, which further points to a set of purchase orders. To create invoice receipt events with a consumption-relation to a set of purchase order, an implicit relation description containing the path of involved element descriptions can be defined.

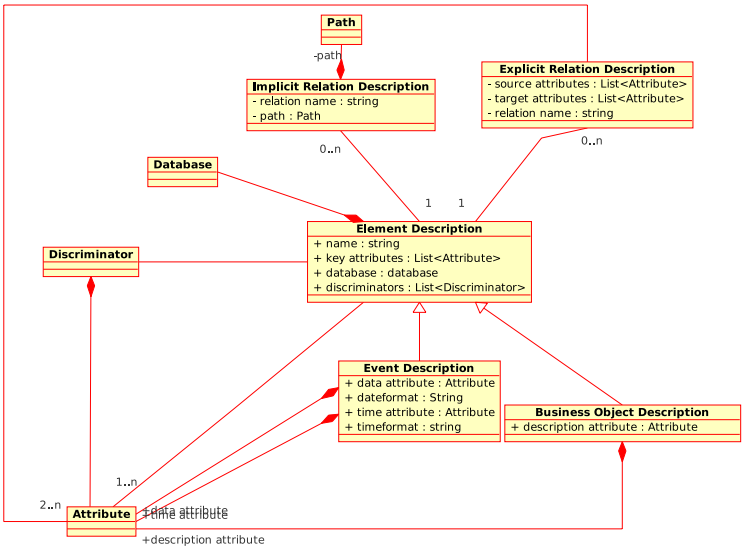

Fig. 3. UML class diagram defining description constructs

3.2 Extracting Business Objects

The first phase in the construction of process instances is to extract all business objects from the set of business objects descriptions that the user has provided. The extraction operation iterates over each business object description and constructs an SQL based on the information they contain.

The SQL statements are executed on the underlying database, and instances from the result sets are stored in a business object index (based on the Lucene Search Engine[3]). This index enables fast lookup of the business objects based on the values of their key and relation mapping attributes. In the index, the business objects refer to each other by id values (loose referencing).

3.3 Extracting Events

The extraction of events creates and executes SQL statements similarly to the extraction of business objects. When the result sets are processed, the date and time values are parsed according to the date and time formats provided in the event descriptions.

To construct relationships to business objects that occur in an event, the business object index is queried. If an event description contains implicit relations to some business objects, these are located by following the relation path and stepwise querying the business object index. All extracted events are stored in a separate event index.

[3] http://lucene.apache.org/

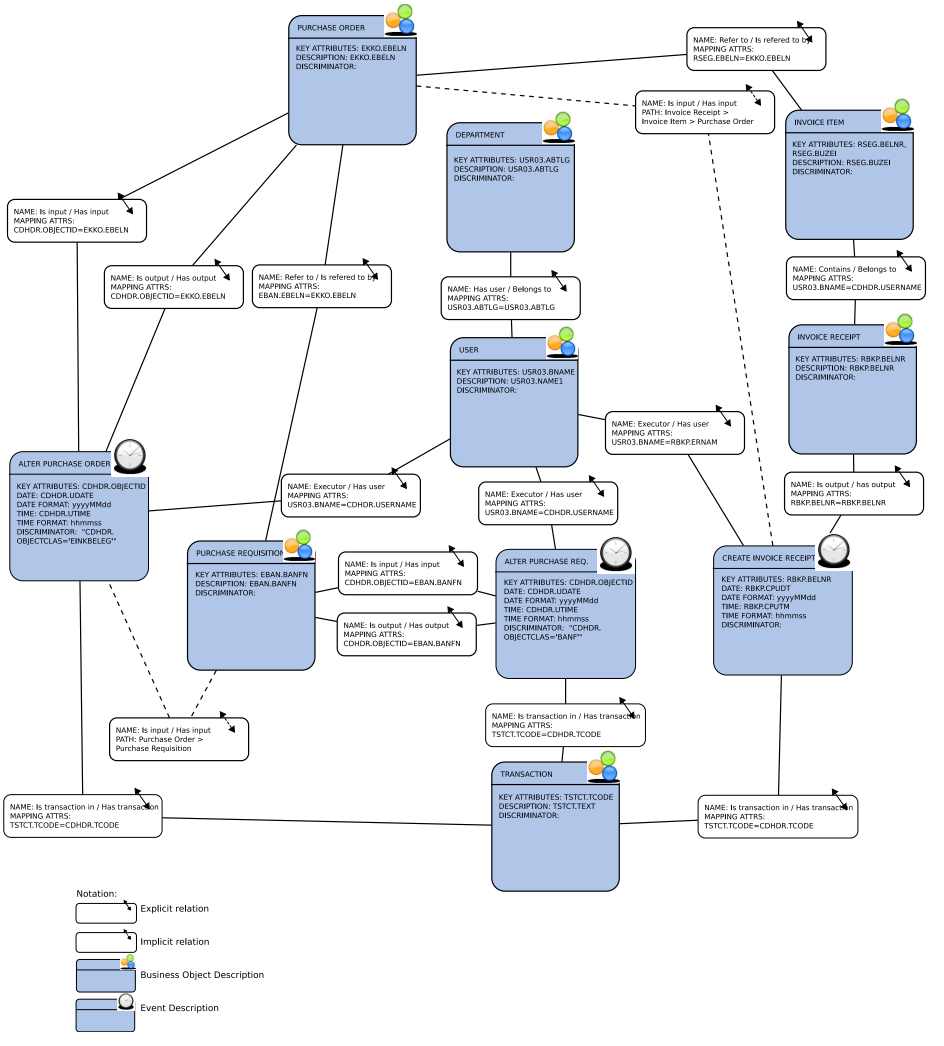

Fig. 4. Example of business object, event and relationship descriptions

As there is no one-to-one mapping between tasks in the SAP Reference Model and transactions, we cannot map executed transactions to the defined business processes. Although this mapping would be preferable, we can still extract meaningful end-to-end process chains of subsequent events. The EVS Model Builder construct process chains by identifying those events that produce and consume the same set of resources.

3.4 MXML Export

The output list of process instances from the EVS Model Builder can be converted to and serialized as MXML. MXML assumes that it is possible to record

events such that (i) each event refers to an activity (i.e., a well-defined step in the process), (ii) each event (named AuditTrailEntry) refers to a process instance, (iii) each event can have a performer (the person executing or initiating the activity), and (iv) events have a timestamp and are totally ordered. The structure also incorporates flexibility for other data requirements by having an additional data elemement at each level [4][7].

To store the process instances as MXML, the user must point out which business object type that represents the originator and activity (named WorkflowModElelement in MXML). For our example in Figure 4, user business objects take the role as originators, while transactions are set as WorkFlowModelElements.

MXML is the event log format for the ProM framework, which is a plug-in based architecture where the kernal offer event log information to its components. Plug-ins within five categories are developed (i) Mining plug-ins (e.g., extraction of Petri nets, social networks, frequency abstraction models, etc.), (ii) Export plug-ins (implementation of "save-as" functionality), (iii) Import plug-ins (e.g., import of instance-EPCs from ARIS PPM), (iv) Analysis plug-ins (implementation of property analysis on som mining results), (v) Conversion plug-ins (implementation of conversion between different data formats, e.g., from EPCs to Petri nets)[4][8].

In ProM the user can explore the data and potentially uncover unknown knowledge about the processes that have been executed. It may be that certain tasks in a process are unreasonably time-consuming compared to others, or that certain operations are under-staffed or over-staffed. More dramatically, the analysis may prove that the organization is not carrying out their business processes according to the policies given. If the uncovered knowledge is used to improve the business processes in an organization, new process mining projects can again be carried out to measure and monitor the new situation.

4 Discussion

The preprocessing support in the EVS Model Builder has been tested in two SAP process mining projects. The projects were carried out at the Norwegian Agricultural and Marketing Cooperative and Nidar (producer and distributor of cholocate and sweets). Both these project target processes related to purchase and logistics, and the analyses targeted vendors behavior specifically. The goals of the projects included:

1. Model discovery - How do the AS-IS models look like?
2. Delivery times - How long delivery time do different vendors have? How do the actual delivery times deviate from planned lead times?
3. Systematic irregulations in deliveries - Do any vendors systematically and over time deliver more or less than what is planned?
4. Order confirmation - To which extent do the vendors send confirmations when orders are placed?

For the model discovery goal, ProM and its modeling plug-ins were used. For the vendor analyses, the constructed process instances enriched with features like planned lead times, and delivery amounts served as the bases for documentation and statistical analyses.

The experiences from the projects is that the preprocessing requires substantial efforts for defining meta descriptions and locating SAP data sources. However, the necessary work of table joins, SQL queries, and date format conversions are automated and hidden at the user level. As a consequence, the time spent on preprocessing and extraction of MXML is drastically reduced. The preprocessing support is especially valuable when longer process chains are mined and the amount of involved database tables is large.As the database structures in implemented SAP systems are mostly consistent, the meta descriptions for preprocessing in one process mining project can be reused and applied in other organizations and process mining projects. That means that if one business object description and its relationships is described for one SAP related process mining project, this information can be reused directly in other SAP process mining projects where the same business object type is involved.

The quality of ontological data in the process mining projects is critical for interpreting events at the system level and constructing meaningful process models. In SAP, information for how to interpret each transaction code is available in the underlying database, but it is not possible to map a transaction directly to tasks and business process hierarchies in the SAP Reference Model. This makes it hard to identify gaps (delta analyses [9]) between the mined models and existing reference models.

Business process models that can be extracted from the MXML describe how transactions, users, departments and other entities depend on each other in the daily operations of a company. In difference from the static reference models, mined process models contain instance data that enable drill-down analyses and the extraction of key performance indicators.

5 Related Work

Also other research activities that target process mining on SAP data have been carried out. In 2004 a master thesis [10] identified potentials for doing process mining on SAP R/3. A two-step methodology was developed, where the first step is to identify the database tables that are needed for the process mining project. They implemented a application, TableFinder, which locates the required tables through business objects[4] in the SAP reference model. The second step in their methodogy is retrieval of the document-flow. In this step they export the data in the tables from step one to an XML format suitable for process mining. The main conclusion of the work is that process mining in SAP R/3 is feasible, but the retrieval of the document-flow is very laborious. The process mining project methodology in figure 1 assumes that the user has good knowledge of the

[4] The EVS Model Builder and the SAP reference model uses the term "business object" to describe different concepts.

underlying data sources. In this way, the methodology by Giessel complements the project methodology in figure 1 with an additional pre-phase for cases where good data knowledge does not exist.

Other reseach projects aim at supporting the conversion of application specific event log data to MXML. The ProM Import Framework[5] allows us to extract process enactment event logs from a set of information systems, including WebSphere, FLOWer, Staffware, PeopleSoft, Eastman, and others. In difference from SAP, these systems offer more complete and defined event log structures.

TeamLog is another tool that supports the pre-processing of event logs from process aware collaboration systems. Like the dependencies between transactions in SAP systems, the processes in such collaboration systems have a highly ad-hoc structure. The output from TeamLog is XML data that can be analyzed by the EMiT (Enhanced Mining Tool) process mining application [11].

In practical settings, it is more common to use data warehouse solutions to analyze important aspects of ERP-supported business operations. These solutions do not reveal new knowledge patterns, though, as they only reflect the performance indicators already defined by the companies.

6 Conclusions

Doing process mining on SAP data have both positive and challenging aspects. Transactional, master and ontological data that are required for constructing meaningful process models are all available in the underlying SAP database. As the transactions cannot be directly mapped to tasks, we are unfortunately not capable of aggregating the mined transaction flows to the defined processes in the SAP Reference Model.

In this paper, we have addressed pre-processing challenges for process mining projects on SAP transactions. With the aid of pre-processing tools, like the EVS Model Builder, large scale process mining in SAP is feasible. SAP related process mining projects are still a complex task, and to set them up and interprete extracted models correctly we need project members with good knowledge both on the data and business level.

In spite of recent successes in process mining, current tools are still hampered by their reliance on clean and well-structured transaction logs. The logs from ERP-supported industrial business operations are however so complex and large that they do not naturally fit into existing process mining tools. As seen from SAP above, there may also be discrepancies between the application itself and the provided business models, which further complicates the analysis of their logs. As a consequence, there is still very little work on process mining of real large-scale logs, and the technology has so far not shown itself to be very useful in the practical ERP world. Our work on the EVS Model Builder, though, demonstrates that complex SAP transaction logs can be pre-processed and transformed into structures that lend themselves for conventional process mining techniques. It shows not only that mining SAP data per se is valuable, but also that current

[5] http://sourceforge.net/projects/promimport

process mining techniques are relevant in real industrial settings, provided that we have the necessary ontological knowledge of the ERP systems and can deal with the magnitude of data.

References

1. Mendling, J., Moser, M., Neumann, G., Verbeek, H.M.W., van Dongen, B.F., van der Aalst, W.M.P.: Faulty epcs in the sap reference model. In: Business Process Management, pp. 451–457 (2006)
2. van der Aalst, W., Weijters, A.: Process mining: A research agenda (2003)
3. Ingvaldsen, J.E., Gulla, J.A.: Model bases business process mining. In: Information Systems Management, Auerbach Publications, vol. 23 (2006)
4. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., de Medeiros, A.K.A., Song, M., Verbeek, H.M.W.E.: Business process mining: An industrial application (2007)
5. Keller, G., Teufel, T.: Sap R/3 Process Oriented Implementation. Addison-Wesley Longman Publishing Co. Inc., Boston, MA (1998)
6. Brancroft, N.H., Seip, H., Sprengel, A.: Implementing SAP R/3, 2nd edn. Manning Publications Co., Greenwich, CT (1998)
7. van Dongen, B.F., van der Aalst, W.M.P.: A meta model for process mining data. In: EMOI-INTEROP (2005)
8. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The prom framework: A new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, Springer, Heidelberg (2005)
9. van der Aalst, W.M.P.: Business alignment: using process mining as a tool for delta analysis and conformance testing. Requir. Eng. 10(3), 198–211 (2005)
10. van Giessel, M.: Process mining in sap r/3: A method for applying process mining to sap r/3. Master Thesis, Eindhoven University of Technology (2004)
11. Dustdar, S., Hoffmann, T., van der Aalst, W.: Mining of ad-hoc business processes with teamlog. Data Knowl. Eng. 55(2), 129–158 (2005)