EVS PROCESS MINER

Incorporating Ideas from Search and ETL into Process Mining

Jon Espen Ingvaldsen, Jon Atle Gulla
Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU),
Sem Sælands vei 7-9, NO-7491 Trondheim, Norway
jonespi@idi.ntnu.no, jag@idi.ntnu.no

Keywords: Search, Process Mining, Business Process Intelligence, Decision Support Systems, ETL

Abstract: Search is the process of locating information that matches a given query. Extract, Transform and Load (ETL) editors provide a user friendly and flexible environment for creating operation chains and digging into and explore data. In this paper, we describe the implementation of a process mining framework, the EVS Process Miner, which incorporates ideas from search and ETL. We also describe two industrial cases that show the value of applying search and graphical operation chain environments in process mining work.

1 INTRODUCTION

The goal of process mining is to extract knowledge from event logs recorded by information systems [Aalst & Weijers, 2005].

Figure 1 show the phases involved in a typical process mining project. The basis for all phases 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 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.

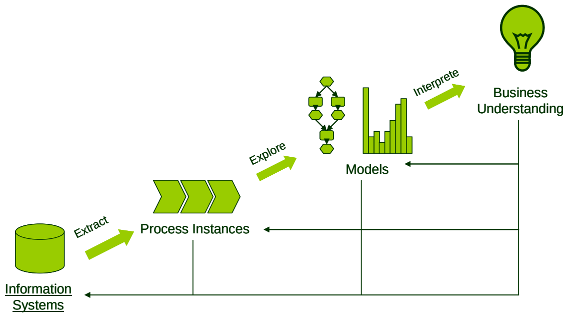

Figure 1: Phases of a process mining project.

To gain a proper business understanding, a typical process mining project has to extract models that focus on different aspects of the business flows (i.e., control flow [Aalst, et al., 2004][Weijters, et al., 2007], load distributions [Ingvaldsen & Gulla, 2006] [Ingvaldsen, et al., 2005] and social networks [Aalst, et al., 2005][Song & Aalst, 2007]) and dig down and investigate subsets of the data.

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. In this paper, we discuss how the exploration phase of process mining work can be supported with search operations and handled in an Extract, Load and Transform (ETL) inspired manner.

Use of search is a simplified yet sophisticated way to gain access to the wealth of information exists in event log related data repositories in Enterprise Resource Planning (ERP) systems. By *search*, we mean an information retrieval system where data objects are serialized and stored in a search index structure. By forming queries of keywords, users can retrieve relevant information without requiring much knowledge about underlying data structures.

ETL editors, on the other hand, enable users to compose chains of relevant operations to dig into and work with the data and construct valuable outputs. The operation chains can be edited graphically, and it is easy for the user to replace operations and modify how they are related.

By combining search and ETL inspired editors for composing operation chains, a process mining worker gets an explorative and easily customizable environment for investigating executed events and process instances. In this paper, we describe a process mining framework, the EVS Process Miner, which integrates search and graphical operation chains.

A description on the EVS Process Miner is given in Section 2. Section 3 describes two industrial cases that show the value of combining search and operation chains in process mining. Section 4 discusses challenges related to search in process mining projects and alternative solutions. Related work is given in Section 5, followed by a conclusion in Section 6.

2 EVS PROCESS MINER

EVS Process Miner is a plug-in based framework for mining business process instances. It is a part of the Enterprise Visualization Suite (EVS), which is developed by Businesscape AS in cooperation with the Information Systems group at NTNU. EVS is a family of process mining related applications and the motivation behind the whole framework is to provide:

- A framework that targets process mining of SAP transactions and the magnitude and diversity of transaction logs in SAP databases.
- Process mining of process instance logs that are enriched with substantial amounts of context information.

Figure 1 shows an overview of the EVS architecture. EVS contains a pre-processing module, named EVS Model Builder, which supports extraction of process related information fragments from SAP transactions and constructs process instance information objects that are serialized and stored in a Lucene based search index [Ingvaldsen & Gulla, 2007]. This search index forms the basis for EVS Process Miner and the search operations presented in this paper.

When designing the EVS Process Miner, we were inspired by the flexible work environment provided in ETL editors. ETL is a process in data warehousing that involves **extracting** data from outside sources, **transforming** it to fit business needs, and ultimately **loading** it into the end target, typically a data warehouse [Karel, R., 2007][Chaudhuri & Umeshwar, 1997]. An operation processes a set of input objects and offers an output object, which is available for further processing by other operations. Most ETL tools consist of a graphical editor where the user can insert operations and simply drag input-output dependencies between them.

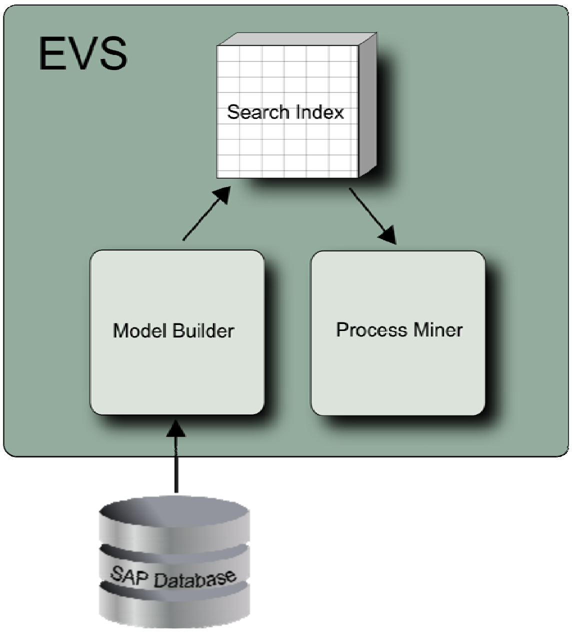

Figure 2: EVS architecture.

The core of the EVS Process Miner is an empty framework for creating chains of operations and serializing operation properties. The framework is constructed such that operations are accessed as plug-ins through defined extension points. This way, it is easy to extend the application externally with new operations. The plug-in architecture is based on the Eclipse Runtime and the Open Services Gateway initiative (OSGI) framework

An operation in the EVS Process Miner consists of programming code for processing inputs and user interfaces for settings property values. One of the implemented operations is *search*. It requires an available Lucene index with process instance related information as input, contains a user interface for specifying a query, and provides an iterator with search results as output. In addition to this search operation, the EVS Process Miner contains operations for visualization, data export, data transformation and extraction of data mining models.

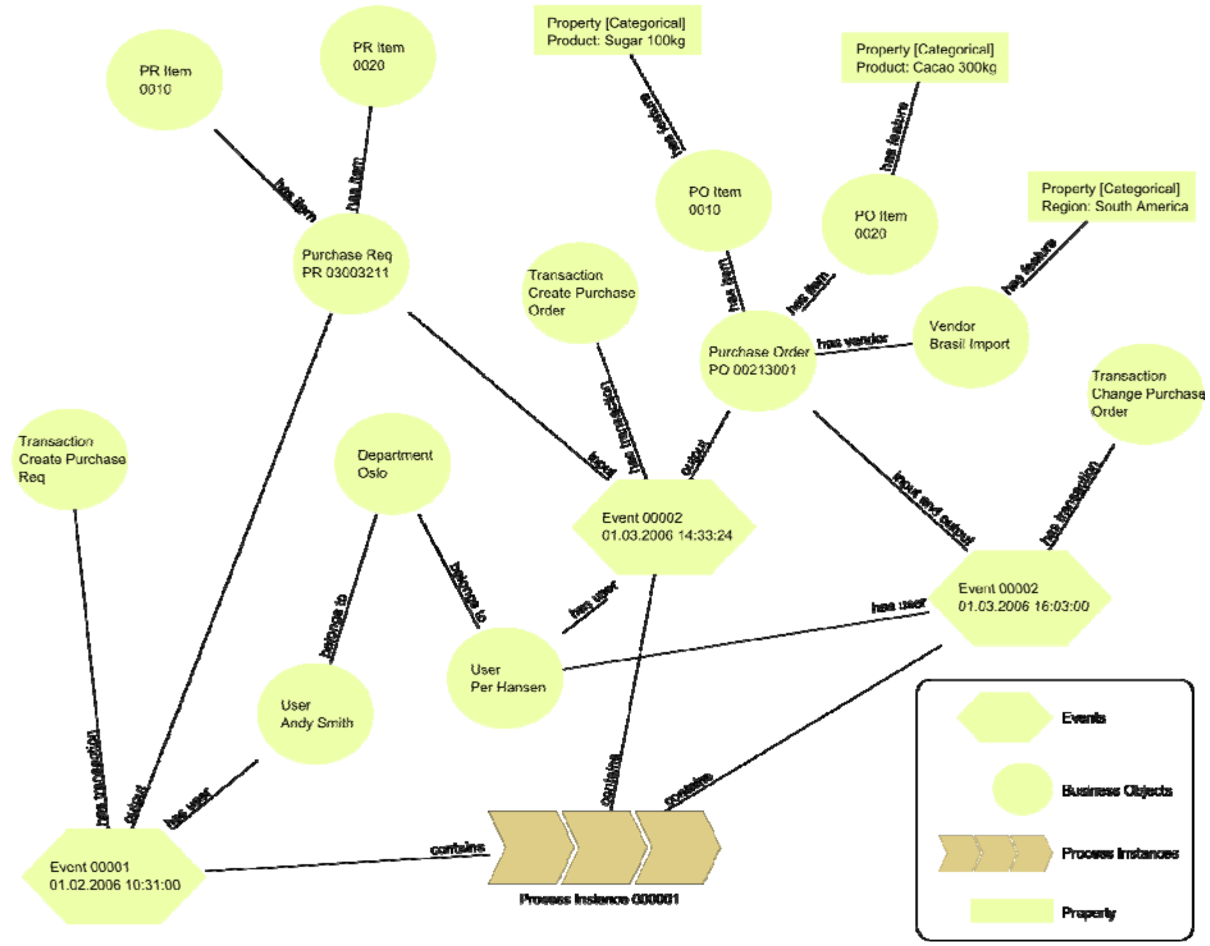

Figure 3: Example of the data structures stored in the search index.

EVS structures business transaction data according to an underlying business process ontology that includes the following concepts:

- Process: “*the set of partially ordered process steps intended to reach a goal*” [Feiler & Humphrey, 1993].
- Process Instance: a chain of event objects that depend on each others output resources. The event objects in an event chain are ordered by their timestamp values.
- Event objects: A happening at a point of time. An event object typically consumes a set of resources to produce outputs. An event object contains a timestamp and relationships to business objects that were involved in the event and form its context.
- Business objects: Domain entities. Business Objects contain a unique identifier and a name.
- Properties: Descriptive attributes which are valuable for our analyses. A property contains a name, a value and a description of its data type (categorical, numerical or date).

Process instances, and event and business objects contains a key value that uniquely identify them. Event and business objects contain also a reference to their type descriptions.

Figure 3 shows an example of such process instance related information objects from a SAP environment. The process instance with the unique id 0000001 contains three events. Each event has a relation to a transaction business object that contains a description of the activity carried out. As we can see, the process instance is initiated by the creation of a purchase requisition, followed by the creation and change of a purchase order. Two different users, Andy Smith and Per Hansen, are involved in the process instance and both belong to the same department. Both the purchase requisition and the purchase order contain two items. The items of the purchase order are enriched with categorical property that provides a textual description of the ordered products. A categorical property is also used to provide vendors with region information.

2.1 Process-Aware Search

Traditionally, search applications assume documents to be unstructured and construct indices on the basis of normalized frequency counts like tf.idf for all the terms that appear in the document set. In this particular process mining case, we have structured log information that can help us interpret and mine the organization's business processes. The representations can help us construct structured indices of process instance-related data fragments, provided that we can transform them into a format that is understood by the process mining application.

The approach taken is to annotate process instance data with ontological information that interprets the data with respect to our underlying business process ontology.

To make the process instance related data fragments searchable, each object is serialized as a set of strings. The set of string fields are set to enable detailed queries from the advanced users, and they contain all data necessary to reconstruct Java objects and their internal relationships. The index contains four fields:

1. `key`: Contains a unique identifier value for each entry. This field can be included in the query when we want to retrieve exactly one known entry.
2. `type`: The index contains entries for events, business objects and process instances. The type field can be specified to filter out only entries of a certain type in the result set.
3. `path`: This field contains a ordered list with the name of each event in a process instance. It is used to retrieve process instances that contain a certain event pattern.
4. `timestamp`: This field contains timestamp information for event entries. This field is useful for filtering out entries within a certain period of time.
5. `data`: This field contains all data necessary to reconstruct Java object representations and relate the objects to each other. This field makes it also possible to search after a given business object, like a vendor, and get all events or process instances for where this business object is related.

Below follows an example of how the process instance from Figure 3 would be represented as a set of searchable strings.

```
key: 000001
type: Process Instance
path: Create Purchase Requisition >
Create Purchase Order > Change
Purchase Order
timestamp: <NULL>
data: eventchain {
  events {
    event {
      type {
        Create Purchase Requisition
      }
      id {
        type = Create Purchase
        Requisition; cdhdr.changenr
        = 009009931002;
      }
      timestamp { 1180994400000 }
        rels {
          rel {
            reltype { output }
              business object {
                type {
                  Purchase
                  Requisition }
                id {
                  type = Purchase
                  Requisition;
                  Eban.banfn
                  = 03003211;
                }
                value {03003211}
                rels { …
```

Note that the data string in the example is shortened and describes just a subset of the related event and business objects.

A detailed query is a query where the user targets specific fields in the index. One example of such a query is

```
path:"Create Purchase Requisition >
Create Purchase Order"
```

Such a query would result in hits for all entries in the index where parts of the *path* field contain a *"Create Purchase Requisition"* followed by a *"Create Purchase Order"*.

These ontology-driven search operations allow us to retrieve, aggregate and interpret process instance data in accordance with a unified semantic model of business processes. This unified model –

the ontology – makes sure that all EVS components understand the data in the same way and provides the standardized terminology needed to add new components later.

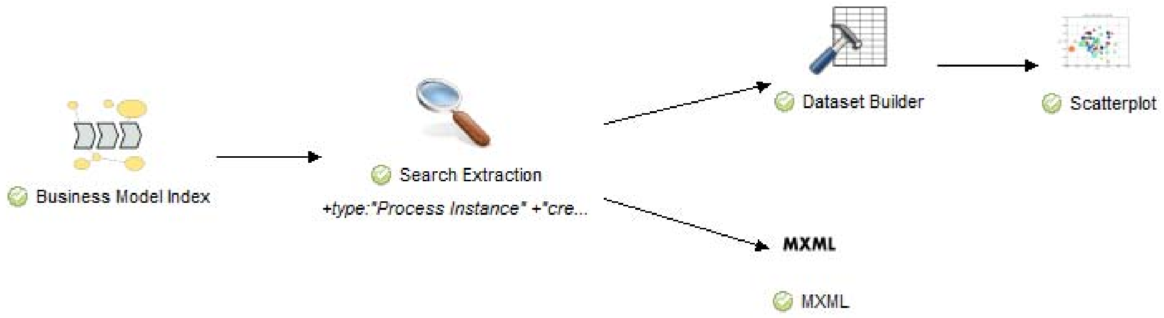

Figure 4: Operation chain that both visualizes data in a scatter plot and exports MXML.

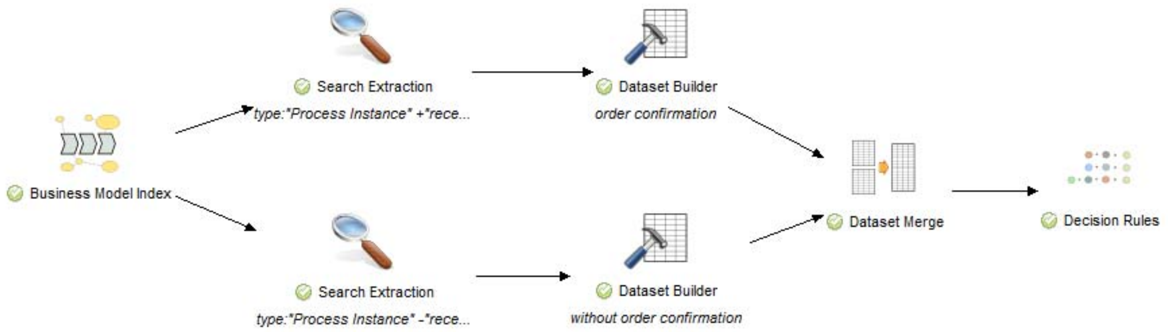

Figure 5: An operation chain that uses a split and merge of output data to achieve the final decision rules.

3 INDUSTRIAL CASES

In this Section we will describe two scenarios that show the value of applying search and operation chains in process mining projects. EVS Process Miner has been tested out in a process mining project on SAP data at a Norwegian medium sized company. The scenarios in this section are based on the scope and findings from this project.

3.1 Case 1: Detailed analysis of sub datasets

The user, a process owner, needed and assessment of how the purchase processes are carried out. The process that he / she wanted to investigate included the activities *"create purchase order"* and *"goods receipt"*. Although the process owner has good knowledge about this process, he needed AS-IS models that describe which activities are actually used, how the loads are distributed, and how the activities depend on each other. He also wanted to investigate how much time each department spends on completing this process.

To carry out the investigations, the process owner composed an operation chain as shown in the editor screenshot in Figure 4. The initial operation contained property values for accessing a Lucene index with process instances, events and business objects. The following search operation was specified with the query:

```
+type:"Process Instance"
+"create purchase order"
```

```
+“goods receipt”.
```

In Lucene, the plus sign means that the following term must occur in the result set entries. The result set of this search contained process instance objects from the string representations in the index that matched the query phrases. The result set was used to construct both a dataset and a MXML file.

Datasets are created to form a basis for statistical analysis, data mining and visualization. The dataset builder operation identifies a large set of potential attributes based on the search result information. The user can then select a subset of attributes that should be included in the dataset. In our simplified scenario, the process owner selected department-name and the duration between *create of purchase order* and *goods receipt* as attributes for further analysis. In the final operation, the dataset was visualized as a scatter plot.

In such a scatter plot, the process owner can see how long time each department spends on the processes and he can compare their performance. As every single process instance is scattered, the process owner also gets an overview of variances and loads for each department.

MXML is the import format for ProM [Dongen & Aalst, 2005]. ProM is an open source process mining framework that contains plug-ins that focus on extraction of different models, i.e., Petri-Nets, EPC, Heuristic Networks, Social Networks, etc. A screenshot of an extracted model using the Fuzzy Miner plug-in is shown in Figure 6. Using such a process mining tool, the process owner can mine the data and visually see all the involved activities in the purchase process, how often they are executed and the relationships between them.

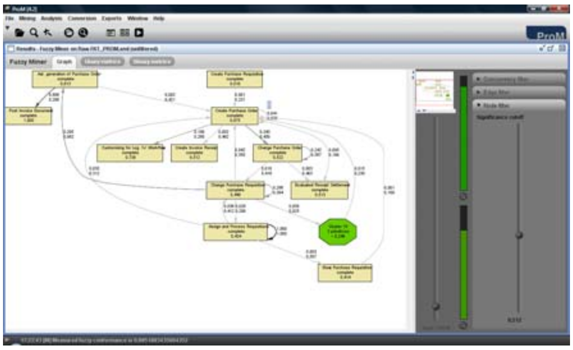

Figure 6: Screenshot of control flow models extracted by the Fuzzy Miner plug-in in ProM.

3.2 Case 2: Analysis of alternative process flows

The process owner recognized in the ProM models from Scenario 1 that the purchase process also contained the activity “*Receive Order Confirmation*”. However, in the ProM models, the process owner could see that large amounts of purchases were done without getting an order confirmation from the vendors. This annoyed the process owner and he wanted to know which vendors, departments and products that are frequently involved when no order confirmation is received.

The process owner wanted to create and analyse a dataset with process instance entries that are labelled with information about whether an order confirmation was received or not. To carry out the investigations, the process owner modifies his operation chain, shown in Figure 5. As for Case 1, the initial operation offers a search index with process instances, events and business objects. As the process owner wanted to investigate process instances for where the activity *“Receive Order Confirmation”* occurs vs. those where the activity is not involved, he created two search operations. The first search operation has the query

```
+type:“Process Instance”
+“receive order confirmation”.
```

The second search operation has a query which is similar, except that the pluss sign in front of *“receive order confirmation”* is changed with a minus sign. In the Lucene query language a minus sign in front of a term is used to specify that term should not occur in the results. The search operations are followed with operations that construct a dataset based on the result sets. In these operations the set of interesting dataset attributes is also selected. In our case, the process owner selected the attributes *vendor*, *department*, and *product*. The rows of the two available datasets are merged together. The output of the merge operation is a single dataset where each row is labelled with the name of its origin dataset. The labelling is added as a separate attributes, named *class*.

An example with entries from such a merged dataset is shown in Table 1. Here, we can see the three selected attributes and a class attribute containing the names of the original datasets. The merged dataset is provided as input to a decision rule operation that identifies significant IF – THEN rules that describes the class attribute and its values.

Vendor	Department	Product	Class
Brasil Import	Oslo	Cacao 300 kg	Order confirmation
Brasil Import	Oslo	Sugar 100 kg	Order confirmation
Henry Wholesale Inc.	Bergen	Palm Oil, 100	Without order confirmation

Table 1, Example entries from the merged dataset.

4 DISCUSSION

The industrial cases described how the process owner approached the process mining task. The graphical editor enabled the process owner to compose operation chains that constructed MXML and graphs that visualize how much time each department spend on the process. Based on these first findings the process owner could customize the operation chains and narrow the focus of further investigations.

In the second case, search was used to retrieve two alternative paths for executing the process. The search results were converted to datasets that were merged and analyzed. Investigation of alternative process executions is not limited to process paths. The same approach could also be applied to investigate and compare process executions where different vendors, departments, or other business objects are involved.

Search provides a simple interface to business process information. An alternative to search indices and keyword queries is more traditional solutions with database tables and Structured Query Language (SQL) statements. Figure 7 shows a comparison between keyword search, detailed search querying, and SQL. By keyword search, we mean search queries where the user does address specific index fields. Detailed search querying, on the other hand, addresses query terms in specific index fields. The comparison involves the following aspects:

- **Data access simplicity**: How quick and easy is it to formulate a query?
- **Required data-structure knowledge**: How much knowledge about underlying data structures is required from the user?
- **Result-set quality**: To which extent is the result-set matching the user request? Is the user getting the information requested, and how much of the result-set is relevant?

As visualized in Figure 7, the approaches that require some background knowledge of the underlying data structures also have the highest result-set quality. However, in the user environment there are trade-offs between the three dimensions. In analysis work like process mining, we have users, like the process owner, that often possesses limited technical knowledge. As we want to explore the data interactively, it is also a big gain if the user can dig into and investigate perspectives and data subsets quickly. In such settings, search is preferable. As the technical knowledge of process mining users varies, it is favourable if both simple keyword search and detailed search querying is supported.

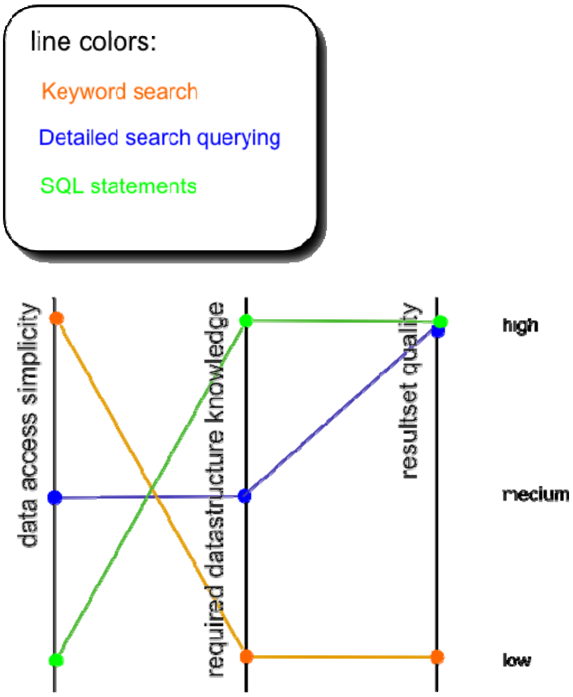

Figure 7: Comparison of alternative approaches for accessing data sources.

5 RELATED WORK.

According to Forrester Research the leaders and strong performers of the ETL market are IBM, Informatica, Business Objects, Oracle, Ab Initio, SAS Institute, Microsoft, IWAY Software, and Pervasive Software [Karel, 2007]. These software vendors incorporate ETL as parts of their data warehousing and business intelligence solutions.

Lately, these software vendors have given considerable attention to enterprise search and search engine technologies. Many ETL vendors have purchased or established partnerships with vendors of enterprise search solutions [Brown, 2006]

There are several motivations behind the merging of the two areas. At one hand, search can be applied to retrieve complete business intelligence reports. On the other hand search can also be applied in the process of investigating data and revealing unknown patterns [Priebe & Pernul, 2003]. Here, search represents a paradigm shift in business intelligence. The benefits of using search technologies on structured data sets have been emphasized by commercial companies like Fast Search & Transfer and Autonomy Inc [Olstad, 2005][Brown, 2006].

At an even more ambitious level, search can be applied to formulate and answer questions directly. Here, we require semantic understandings of both user requests and data available [Gulla, et al., 2006]. Within several domains, efforts are done to integrate semantics, in form of ontologies, in information retrieval systems [Guha, et al., 2003]. The ontological information is used both to extend the query language and annotate the contents of available documents [Sheth & Ramakrishnan, 2003].

[Medeiros, et al., 2007] discusses several directions for the development of semantic process mining and monitoring tools. They point out that the main opportunity provided by such systems is the link between the event log structures and the actual concepts they represent. This linking is achieved by annotating the elements with concepts in ontologies. Important here is also the fact that this ontology is used throughout the process mining environment, not just for the search operations.

6 CONCLUSIONS

This paper have shown how the exploration phase of process mining work can be supported with search and graphical editors for customizing operation chains and working with the data.

Search provides a simple interface to process mining sources, without requiring extensive knowledge about underlying data structures. By combining search and ETL inspired editors, a process mining worker gets an explorative and easily customizable environment for investigating executed events and process instances. The practical value of such an environment was demonstrated in two industrial cases.

In future work, we will look into how search technologies can be applied to map executed transactions in SAP systems with business process definitions in the reference models. We also see potentials for applying ontological reasoning on process instance logs that are enriched with context information from related data sources in the SAP databases.

REFERENCES

Aalst, W.M.P. van der, Reijers, H.A., & Song, M. (2005). Discovering Social Networks from Event Logs *Computer Supported Cooperative Work*, 14-6, 549-593

Aalst, W.M.P. van der, & Weijters A.J.M.M. (2005). *Process-Aware Information Systems: Bridging People and Software through Process Technology*, (pp. 235-255). Wiley & Sons.

Aalst, W.M.P. van der, Weijters, A.J.M.M., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineerin*g, 16-9, 1128-1142

Alves de Medeiros, A.K., Pedrinaci, C., Aalst, W.M.P. van der, Domingue, J., Song, M., Rozinat, A., Norton, B. & Cabral, L. (2007). An Outlook on Semantic Business Process Mining and Monitoring**.** *OTM Workshops (2),* 1244-1255

Bloor Research. (2004). ETL^Q From SAS Institute. An extract from the Bloor Research Data Quality Report. *http://www.sas.com/offices/europe/germany/download/files/solutions/SAS_Bloor_ETLQ.pdf,* accessed: 16.11.2007

Brown, M. (2006). The Forrester Wave™: Enterprise Search Platforms, Q2 2006. *Forrester Research.*

Chaudhuri, S., & Umeshwar D. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Rec. 26-1, 65-74.*

Dongen, B.F. van, & Aalst, W.M.P. van der. (2005). A Meta Model for Process Mining Data. *Proceedings of the CAiSE'05 WORKSHOPS, volume 2, 309–320*

Feiler, P.H., & Humphrey, W.S. (1993). Software process development and enactment: concepts and definitions. *Second International Conference on the Software Process*. 25-26

Guha, R., McCool, R., and Miller, E. (2003). Semantic search. *In proceedings of the 12th international conference on World Wide Web, ACM Press, 700-709.*

Gulla, J. A., Borch, H. O., & Ingvaldsen, J. E. Unsupervised Keyphrase Extraction for Search Ontologies. *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems (NLDB'2006), 25-36*

Karel, R. (2007). The Forrester Wave™: Enterprise ETL, Q2 2007. *Forrester Research.*

Ingvaldsen, J.E., & Gulla, J.A. (2007). Preprocessing support for large scale process mining of SAP transactions. To appear in proceedings *of the 3rd International Workshop on Business Process Intelligence (BPI).*

Ingvaldsen, J.E., & Gulla, J.A., (2006). Model based business process mining. *Information Systems Management*, Vol 23, Issue 1, 19-3. Auerbach Publications.

Ingvaldsen, J.E., Gulla, J.A., Hegle, O.A., & Prange, A. (2005). Revealing the Real Business Flows from Enterprise Systems Transactions. *Proceedings of the Seventh International Conference on Enterprise Information Systems (ICEIS).*

Olstad, B. (2005). Why Search Engines are Used Increasingly to Offload Queries from Databases. *VLDB 2005: 1*

Priebe, T., & Pernul, G. (2003). Integration of OLAP and Information Retrieval. *DEXA Workshops*, IEEE Computer Society.

Sheth, A.P., Ramakrishnan, C. (2003). Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. *IEEE Data Engineeting Bulletin. IEEE Data Engineering. 40-48*

Song, M.S., & Aalst, W.M.P. van der. (2007). *Towards comprehensive support for organizational mining.* Beta working paper series (Rep. No. WP 211), 28 pp.

Weijters, A.J.M.M., Aalst, W.M.P. van der, & Alves De Medeiros, A.K. (2006). Process mining with the heuristics Miner-algorithm., BETA Working Paper Series (Rep. No. WP 166), 34 pp.