KNOWLEDGE PROCESS MINING AND OPTIMIZATION

Tadej Štajner, Carolina Fortuna, Dunja Mladenić, Marko Grobelnik
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773900
e-mail: {firstname.secondname}@ijs.si

ABSTRACT

This paper addresses the problem of optimizing knowledge processes that have been automatically identified from a data stream. We are proposing extension of TaskMiner tool for process mining and visualization in collaboration processes. The proposal is to enable TaskMiner graphical interface for process mining to supports a larger scope of the process analyst's workflow: process discovery, visualization and measurement, providing decision support for process refactoring and follow-up measurements of implemented optimizations.

1 INTRODUCTION

Knowledge processes, as seen in this paper, involve knowledge workers in an enterprise who are usually involved in several projects that require accessing different data sources, exchanging messages, browsing the Web etc. With the wide usage of computers in enterprises, one can expect that each knowledge worker has access to a personal computer, where a program can be installed to record activity on the level of complex events, such as, at time *T* a person *P* has accessed a document *D*. We assume that each event is associated to a context (e.g., a project) and that it is possible to cluster the events so that we automatically identify which events belong to the same context. Each context has data collections associated to it and possibly interconnected with some kind of relation. In our scenario, knowledge workers switch from working in one context to the other on weekly, daily or maybe hourly bases.

This paper presents bottom-up approach to optimization of *knowledge processes*, where knowledge processes are seen as loosely coupled sets of activities occurring in some context. The developed approach is semi-automatic, implemented as an extension of TaskMiner tool. TaskMiner enables displaying and reporting the state of the knowledge process which exposes the metrics needed for optimization. The initial version of TaskMiner has already been described in [2]. In this paper, we have extended TaskMiner with reporting capabilities that enable measurement of objective metrics, needed for process optimization.

The paper is structured as follows: Section 2 briefly presents the TaskMiner tool. The proposed approach to semi-automatic knowledge process optimization is presented in Section 3. Sections 4 gives conclusions and some directions for future work.

2 PROCESS MINING USING TASKMINER

Process Mining that is used as a starting point of our work is based on the bottom-up approach using data mining techniques to obtain a probabilistic process model. The setting is as follows: given a database, describing events in a business setting, executed by actors on resources, construct a probabilistic temporal model that best describes the action patterns appearing in the event. The model construction is performed by action mining followed by process mining [3]. Since the data is provided in the form of a graph, composed of multiple different node types, action mining is addressed as an example of a multi-relational clustering problem.

Process mining was performed by using Markov Models for finding frequent sequences of actions in the data, as one of the standard algorithms applied for process mining with the extension of pruning the obtained models by selecting only the statistically significant transitions [5]. It is developed for process mining on TNT (text, network, time) data proposed in [1]. Results from the developed prototype were shown on real-world datasets. The real-world data captured in TNT events reflects the situation of knowledge workers in a real-world setting, including interruptions and context switches, noise from different sources, under-defined tasks and contexts.

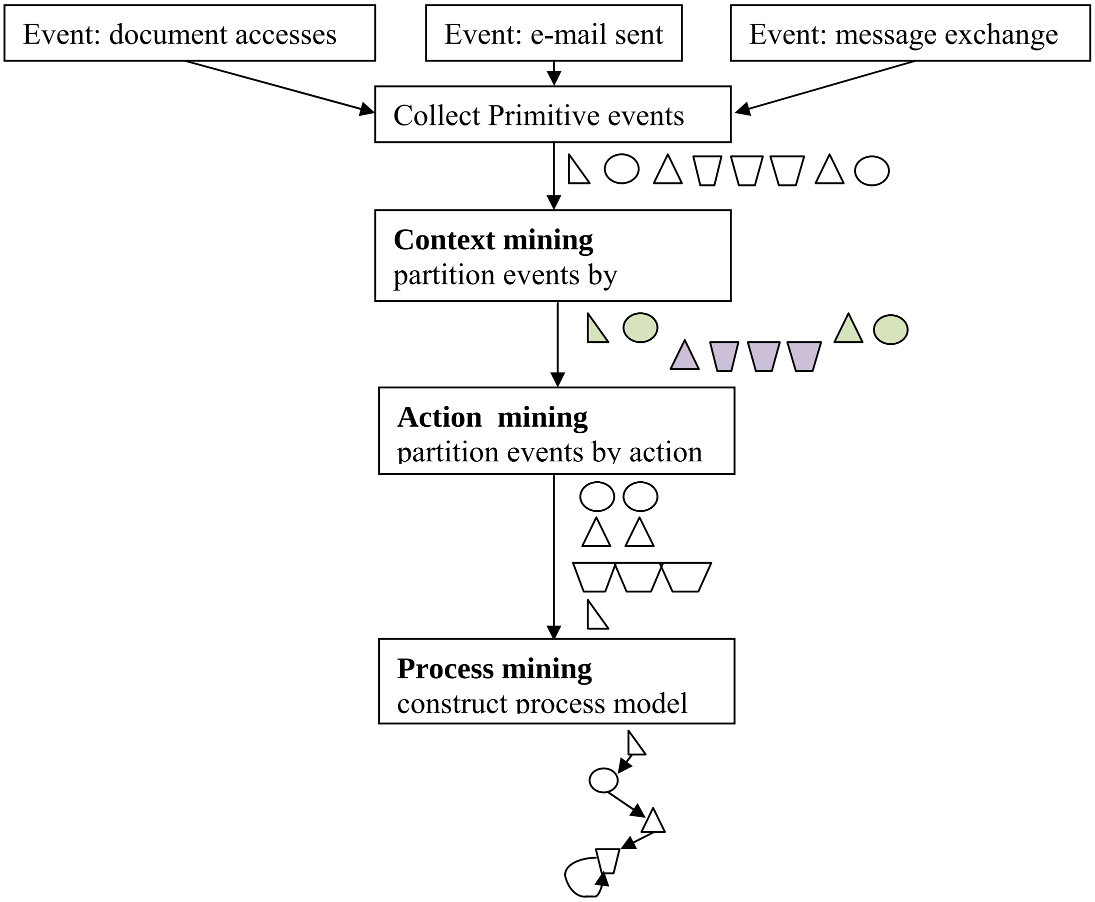

Figure 1: *Architecture of probabilistic process mining. Icons represent individual events, their shapes represent actions, and their colors represent contexts.*

Figure 1 shows the architecture of probabilistic process mining as implemented in TaskMiner. First, events are collected, such as, document access, e-mail send or receive, exchange of short messages between the users, etc. We then execute the steps of context and action mining.

In the case of context mining, context correspond to clusters of events using people and terms that appear in the events' contents, since these are the features which knowledge workers tend to consider as distinguishing for contexts.

On the other hand, action mining is a similar clustering procedure, but executed on a different feature set of the same event stream. In action mining, the features consist of content terms (without named entities), the type of the event and, an abstraction of social properties such as the roles of participants (i.e. is it a private conversation or a group, or does it span a single or multiple organizations). This view of features gives us a context-free representation of events. The purpose of clustering events into actions is to construct a process model from those actions.

Context mining is applied in order to identify contexts and partition events by context. Action mining is applied to partition events by actions. In process mining, a process model is constructed by finding sequences of actions based on the data. We assume that a process model contains actions belonging to the same context.

3 APPROACH DESCRIPTION

TaskMiner offers a graphical front end for the process mining service (see an example of visualization in Figure 2). It enables the process analyst to explore process models using different process mining views and parameters: viewing per-user or per-context model with varying degree of granularity. Its core process mining functionality, prototyped in [3] is also used for process-based prediction [2] and represents an implementation of a probabilistic model for the activities of knowledge workers [4].

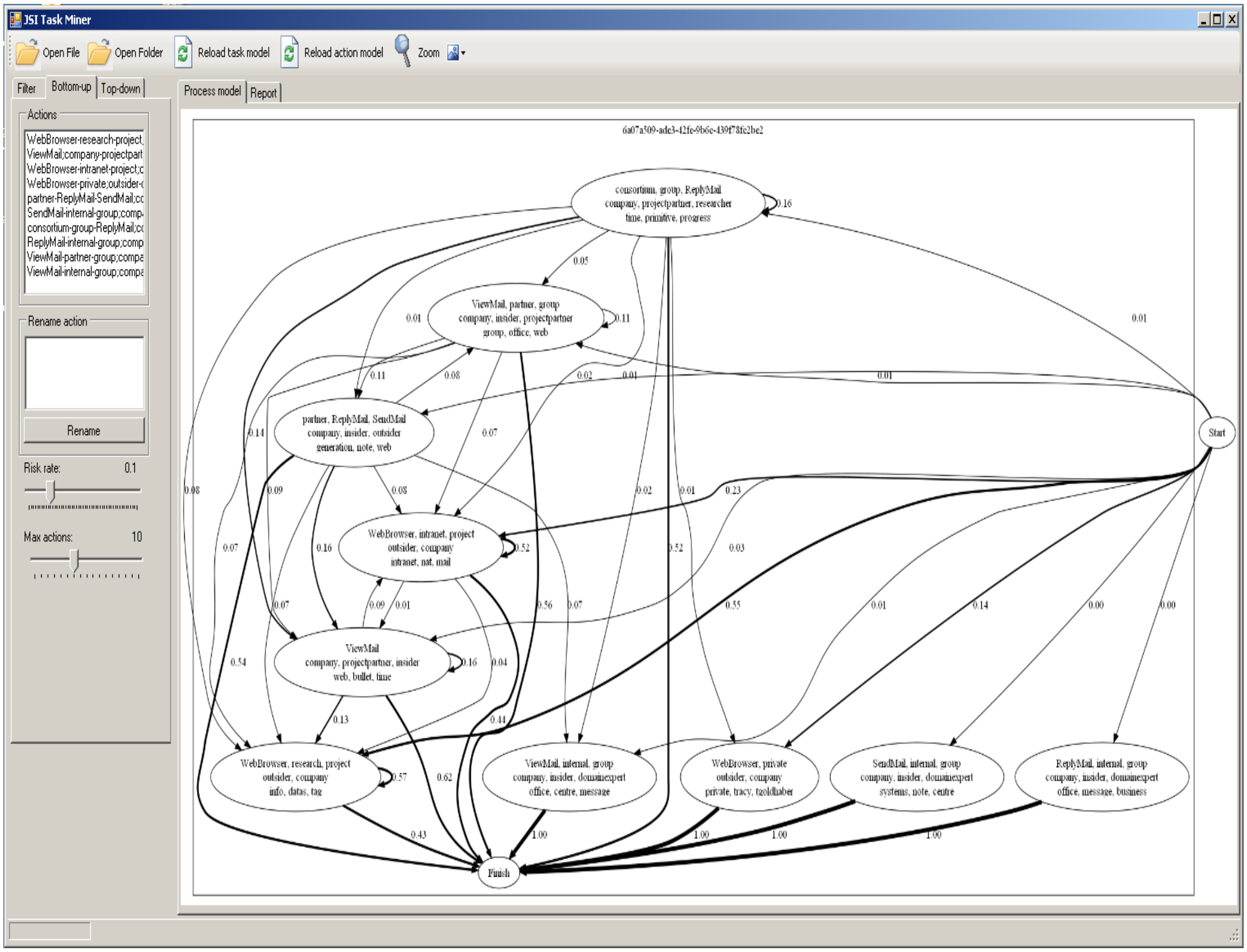

Figure 2: *TaskMiner process visualization view. Thicker lines represent more frequent transitions.*

Imagine a process analyst exploring the process visualizations, as seen in Figure 2, with the purpose of refactoring the process afterwards. Compared to other tasks, process optimization has another key component: objective metrics that one is able to optimize. With the data that we are using as input for process mining, we are able to measure metrics from the perspective of mined actions and their distribution over time:

- Total time spent executing some action
- Average time spent executing action
- Number of events representing action
- Percentage of time spent executing action

Although one may be able to derive some of these metrics from the total time and number of events per action alone, the average time spent per action is more easily interpretable as an objective metric. For instance, one is able to isolate the effects of a particular optimization approach either on average execution time (faster actions) or a lower number of required executions (less actions). Also, some optimization approaches affect only some actions - in those cases, one is still able to measure the effect on that particular type of action alone.

Since each individual event is only associated with its time stamp, we approximate its duration using the time difference from the previous event, as long as the last event occurred in the last 60 minutes with the assumption that individual actions are not longer than 60 minutes for this particular domain. The metrics reporting functionality are implemented as another tab in the TaskMiner interface, showing a data grid of metrics for each action.

As a prototype study, we have measured these metrics on the real-world log dataset from a major telecommunication provider. The process mining metric report in Figure 3 shows that all e-mail related actions tend to have very similar duration of around two to two and a half minutes, while all of the different browsing- related actions tend to average around 45 seconds. In other words, reading an e-mail takes on average the same effort as writing or responding to one.

6 CONCLUSION

We have described a bottom-up approach to optimization of Knowledge Processes, based on extending TaskMiner tool for process mining and visualization. The TaskMiner graphical interface for process mining now supports a larger scope of the process analyst's workflow: process discovery, visualization and measurement, providing decision support for process refactoring and follow-up measurements of implemented optimizations.

Future work involves using the implemented reporting to test various process optimization approaches for various common knowledge worker tasks, such as, applications for optimizing e-mail management or process-based resource prediction for document management.

7 ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under PASCAL2 (IST-NoE-216886) and ACTIVE (IST-2007-215040).

References

[1] M. Grobelnik, D. Mladenić, J. Ferlež; Probabilistic temporal process model for knowledge processes: handling a stream of linked text, Proc. of the 12th international multiconference on Information society IS-2009, SiKDD, pages 222–227. 2009.

[2] T. Štajner, C. Fortuna, D. Mladenić, M. Grobelnik; Prototype for forecasting future behavior and actions in collaboration process; ACTIVE Deliverable, http://wiki.active-project.eu/wiki/D3.5.1, 2010

[3] T. Štajner, C. Fortuna, D. Mladenić, M. Grobelnik; Scalable learning of probabilistic temporal models from data (prototype and evaluation); ACTIVE Deliverable, http://wiki.active-project.eu/wiki/D2.2.2, 2010.

[4] T. Štajner, D. Mladenić: Modeling knowledge worker activity; Workshop on Applications of Pattern Analysis - WAPA, Cumberland Lodge, 2010

[5] S. Jacquemont, F. Jacquenet, and M. Sebban. Mining probabilistic automata: a statistical view of sequential pattern mining. Machine Learning, 75(1):91–127, 2009.

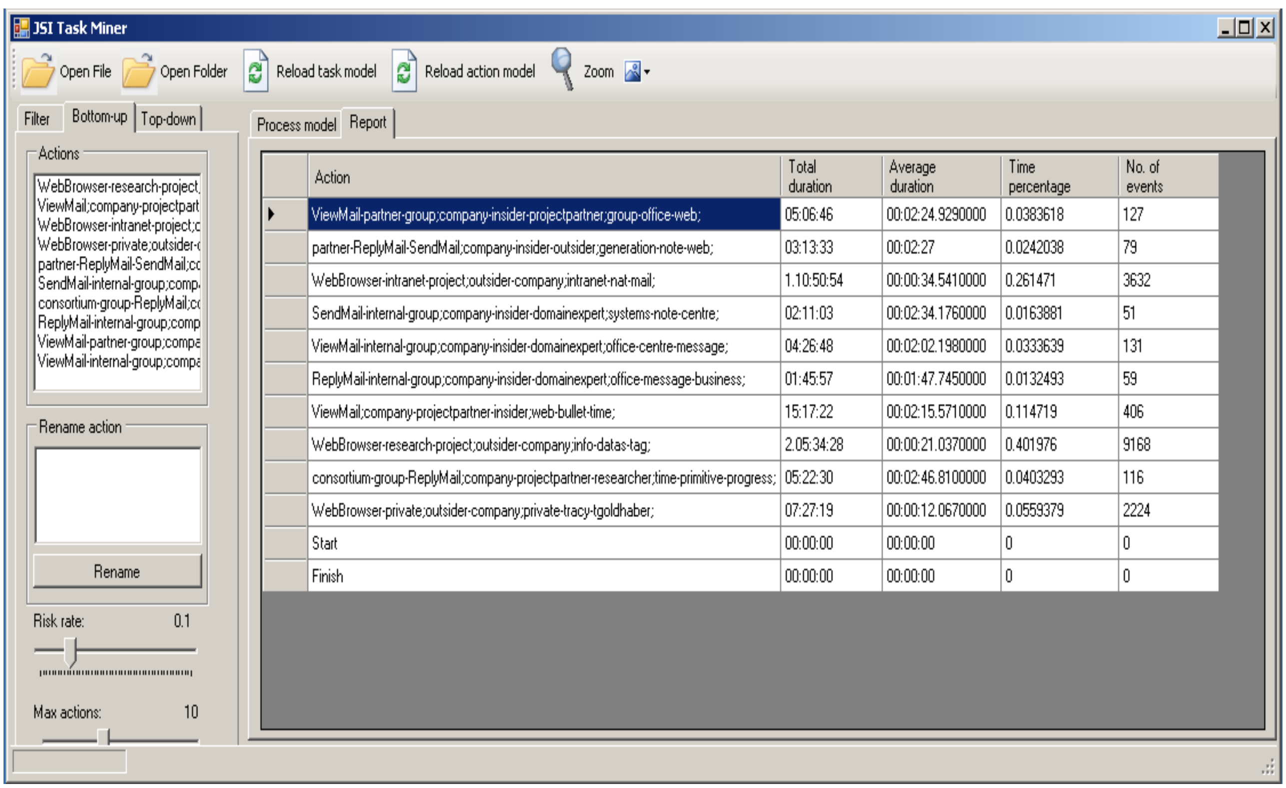

Figure 3: *Example process mining report in the extended TaskMiner*