Available online at www.sciencedirect.com
ScienceDirect
Procedia Technology 9 (2013) 489 – 497
Procedia Technology

CENTERIS 2013 - Conference on ENTERprise Information Systems / PRojMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

Discovering reference process models in the context of BPM projects

Asma MEJRI[a], Sonia AYACHI GHANNOUCHI[a,b,*]

[a]*Laboratory RIADI-GDL, ENSI, Mannouba 2010 , Tunisia*
[b]*High Institute on Management of Sousse, Tunisia*

Abstract

This paper is related to process mining, specifically the processes' discovery. Our goal, through this research work, is to build an approach that extracts a reference model, modeled in BPMN language, from the event logs related to different processes, based on the algorithm α. We also aim to make the configuration of the extracted process models in BPMN language. So, we developed a plug-in in ProM environment. We tested this plug-in by using test cases for which preliminary results are encouraging.

Keywords: Process Mining, Process Discovery, BPMN, event logs, ProM.

1. Introduction

If information technology continues to change rapidly, the economic environment is and will be turbulent. These turbulences are putting pressure on companies that must react in their environment in a quick and flexible

* Asma MEJRI. Tel.: +0-021-623-844-689
E-mail address: mejri.assma@gmail.com.

doi:10.1016/j.protcy.2013.12.054

way, in order to fit the changes. Because of the turbulences and rapid changes, business processes must be able to respond to the corresponding needs of the company.

A business process can be defined as a related group of tasks that together create value for a (internal or external) customer. Examples of business processes are the processing of orders in a factory or the handling of insurance claims at an insurance company [1].

The importance of business process is growing at a very rapid rate over recent years. There are several factors that can lead companies to change their business processes:
They are threatened by competition.
They need to develop new solutions to respond better to the customers' needs and requirements.
They must respond to organizational changes.
They must respond to the emergence of new technologies

Then the reference models can be used, because they provide improvements in companies. They are developed either inside or outside the company. They are designed to meet the needs of the company. Their application is an effective method to accelerate the improvement of business processes and adapt best practices (Enterprise Best Practices).

Since the emergence of disciplines such as Business Process Management (BPM) and Business Process Reengineering (BPR), organizations have been focusing more and more on their business processes in order to improve results and reduce costs [1].

To improve and support business processes, many organizations have installed Process-Aware Information Systems (PAIS). The emergence and proliferation of PAIS has led to the emergence of process mining. Process mining is a method to study the process execution. Its objective is to extract information (about the process) from event logs. Event logs are lists in which information about the actual execution of the process is recorded.

2. Background and Research questions

2.1. Process mining

Process mining consists in extracting valuable, process-related information from event logs [2]. It can be applied in a wide variety of information systems. There are several types of PAIS (Process Aware Information Systems) that produce event logs, e.g. workflow management system, ERP systems (like SAP)… These systems provide detailed information about the activities that have been performed. For this process mining, data logs of an enterprise information system are processed by a mining tool, e.g. EMiT, Little Thumb, Process Miner, etc [3].

In literature, there are many approaches that concentrate their researches on process mining algorithms. Process mining algorithms have been implemented in various academic and commercial systems [2].

2.2. Reference models

Business process management frequently uses models for modeling the enterprise, the information system specification or end-user training, etc [3].

Let's start with a general definition of a model. A model is defined as a representation of an abstraction of a part of the real world, which is expressed in a representation language. More precisely a business model is a set of models describing various aspects of an enterprise that we want to analyze.

Finally, a reference business model has the property of identifying, collecting and representing the elements of a business system [4]. It's also considered as a generic conceptual model that formalizes recommended practices for a given domain [5].

The main objective of reference models is to streamline the design of one particular enterprise models by providing a generic solution. Thus, reference modeling is closely related to the *reuse of information models* by

providing a generic model solution that can be adapted to a specific model reflecting individual requirements [5].

2.3. The α algorithm

The alpha algorithm is an algorithm used in process mining that aims at reconstructing a workflow net graph. A workflow net is a triple of places, transitions and flow relations. It maps a log file onto a workflow.

The alpha algorithm takes an event log and produces a Petri net explaining the behavior recorded in the log [2]. The pre-processed log files can serve as an input for a process mining algorithm. This algorithm searches for re-occuring patterns in the execution traces of the system in question and generalizes the overall process behavior through process models [6].

2.4. Research questions

The idea in this research work is to study the manner in which the process mining contributes to the extraction of the reference models.

Event logs can be extracted either from the same process or several processes that exist in different organizations. Thus the first research question is: Is it possible to use multiple processes from different organizations to identify reference models?

The logs collected from different processes must be used. Using multiple processes provided from different organizations will allow obtaining the most general representation of practices in the concerned domain which corresponds to a reference model. Thus the second research question is: How to use event logs to uncover business process models from which a reference model will be extracted?

In addition, several approaches use Petri nets to model business processes [6]. Other approaches have opted for BPEL [7]. In this paper, we chose BPMN.

The next section in this paper reflects on the solution that we propose to answer these research questions.

3. Proposed solution : Using multiple event logs for mining a BPMN Model

3.1. Using multiple event logs

Event data are generally scattered over different data sources. Data can be dispersed because of technical or organizational reasons. For example, there may be existing systems that contain critical data or information systems that are used only at the departmental level. The data can even be scattered across multiple organizations. Thus, we aim to use event logs that originate from several processes.

3.2. Mining of a BPMN model

Business Process Modeling Notation (BPMN) is an increasingly important standard for process modeling and has enjoyed high levels of attention and uptake in BPM practice. BPMN is indeed a rich language and allows us to define multitude business scenarios, ranging from internal process choreographies to inter-organizational process orchestrations, service interactions and workflow exceptions [8].

The strengths of the BPMN formalism can be expressed as follows:

- An intuitive and ergonomic notation that can be used by the actors of the organization and business management.
- A rich vocabulary (set of concepts and relationships) which is suitable for design requirements in complicated business processes, which is scrupulously defined to provide a robust tooling.

For all these reasons, to depict process models we will use the BPMN notation.
In this research work, our solution is based on some definitions that we propose in order to deal with obtaining BPMN models from multiple event logs.

Definition (BPMN process):
A central process of BPMN is a tuple P = {O, A, E, B, T, T^R, F }
O is a set of objects that can be partitioned into disjoint sets : activities A, events E, Gateways B.
- A can be partitioned into disjoint sets of tasks T.
- $T^R \subseteq T$ is the set of the received tasks.
- E can be partitioned in disjoint sets of start events $\{e^D\}$, intermediate events $\{e^I\}$ and end event $\{e^F\}$.
- B can be divided in disjoint sets:
  - B^{XOR}: exclusive decision and merging.
  - B^{OR}: inclusive decision and merging.
  - B^{AND}: forking and joining.
  - B^{comp}: complex conditions and situations.
- $F \subseteq O \times O$ is the control flow relation, a set of objects that link objects.

We limit our work on B^{XOR} and B^{AND} gateways. In order to define the BPMN gateways it's essential to determine the ordering relations. We distinguish four *log-based ordering relations* that aim to capture relevant patterns in the log.

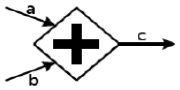
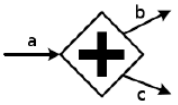
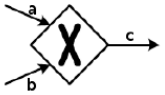
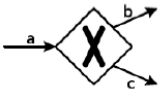
Definition (log-based ordering relations)
Let L be an event log over A. Let a, b ϵ A :
- $a >_L b$ if and only if there is a trace $\sigma = <t_1,t_2,t_3,\ldots,t_n>$ and i $\in$ {1,..,n-1} such that $\sigma \epsilon$ L and t_i=a ans t_{i+1}=b
- $a \rightarrow_L b$ if and only if $a >_L b$ and b not $>_L$a
- $a \#_L b$ if and only if a not $>_L$ b and b not $>_L$ b
- $a \|_L b$ if and only if $a >_L b$ and $b >_L a$

The log-based ordering relations can be used to *discover patterns* in the corresponding BPMN process model as it is illustrated in table 1.

In this paper, we focus on deriving the BPMN model from multiple log files. For this purpose, we consider each log file as a set of event traces. Each event trace is a set of activities.

The input for the multi α-algorithm is a multiple event logs L. The algorithm that we propose is based on the α algorithm.

Table 1. Modeling BPMN gateways using ordering relations

BPMN gateway	Modeling using ordering relations
	- a $\rightarrow_L$ c - b $\rightarrow_L$ c - a $\|\|_L$ b
	- a $\rightarrow_L$ b - a $\rightarrow_L$ c - b $\|\|_L$ c
	- a $\rightarrow_L$ c - b $\rightarrow_L$ c - a $\#_L$ b
	- a $\rightarrow_L$ b - a $\rightarrow_L$ c - b $\#_L$ c

Definition (multi α-algorithm)

Let L be the set of the different event logs. $L = \{L_i$ tel que L_i: $T_i^* \rightarrow N\}$ and N is the number of the event logs.

1. $T_{Li} = \{t \in T_i \mid \exists_{\sigma \in L}\ t \in \sigma\}$;
2. $T_{Ii} = \{ t \in T_i \mid \exists_{\sigma \in L}\ t \in first(\sigma)\}$;
3. $T_{Oi} = \{ t \in T_i \mid \exists_{\sigma \in L}\ t \in last(\sigma) \}$;
4. $TR_{Li} = \{(t_1,t_2,t_3) / t_1 \in T_{Li}, t_2 \in T_{Li}, t_3 \in TLi\}$;
5. $AS_{Li} = \{(t_1,t_2,t_3) \in TR_{Li} / t_1 \rightarrow t_2, t_2 \rightarrow t_3, t_1 \| t_3\}$;
6. $AJ_{Li} = \{(t_1,t_2,t_3) \in TR_{Li} / t_1 \rightarrow t_3, t_2 \rightarrow t_3, t_1 \| t_2\}$;
7. $XS_{Li} = \{(t_1,t_2,t_3) \in TR_{Li} / t_1 \rightarrow t_2, t_1 \rightarrow t_3, t_2 \# t_3\}$;
8. $XJ_{Li} = \{(t_1,t_2,t_3) \in TR_{Li} / t_1 \rightarrow t_3, t_2 \rightarrow t_3, t_1 \# t_2\}$;
9. $\alpha_i(L) = (TL_i, AS_{Li}, AJ_{Li}, OS_{Li}, OJ_{Li})$;
10. Increment and repeat first step until i=N+1 ;
11. $\alpha(L) = \{\alpha_i(L), i <= N\}$.

L_i is an event log that belongs to L. In step 1, it is checked which tasks do appear in the log T_{Li}. T_{Ii} is the set of start tasks (step2). T_O is the set of end tasks. In step 4, all the possible triplets of tasks are constructed. Steps 5, 6, 7 and 8 form the core of the algorithm. The challenge in these steps is to build the gateways. In steps 5 and 6 we aim at determining the B^{AND} gateways. Besides in steps 7 and 8 we aim at constructing the B^{XOR} gateways. In the next steps, the mined model is build: we deal with the next event log in order to mine the corresponding BPMN model.

4. Developed Plug-in and experimentation

4.1. Using ProM

To develop our Plug-in we used ProM. ProM is an open-source process mining tool supporting many techniques such as Process discovery, conformance checking, social network analysis, organizational mining, decision mining, history-based prediction and recommendation, etc [2]. *ProM framework* is a "plug-able" environment for process mining. New functionalities can be added to the framework by means of plug-ins.

The plug-in that we added to ProM framework was "Mine for a BPMN model using multi-log alpha algorithm". Figure 1 shows the main interface of our plug-in.

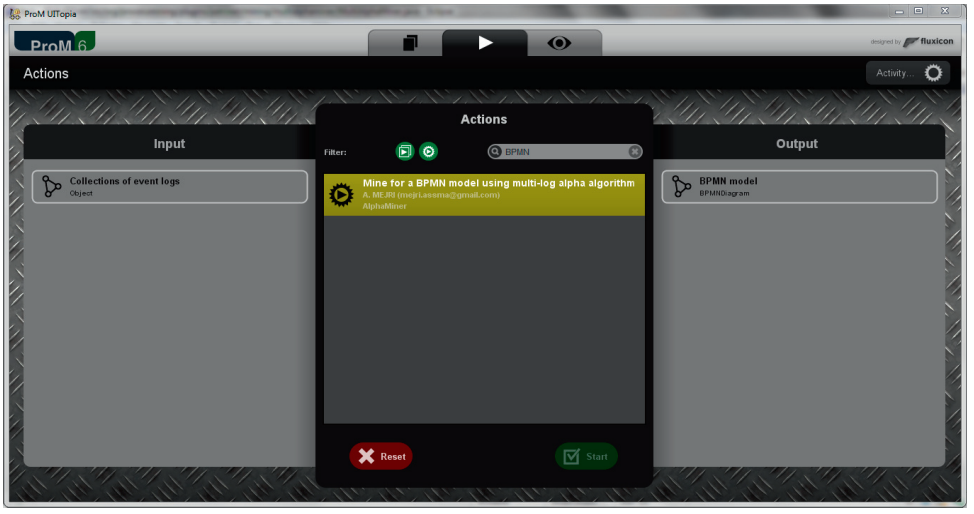


Fig. 1. The plug-in for a BPMN model using multi-log alpha algorithm

This interface consists essentially of three parts:
Actions: the user must select the plug-in "Mine BPMN model for a log using multi-algorithm"
Inputs (the left side): The user can choose a collection of event logs.
Output (right side): The output of the plug-in is composed of the corresponding BPMN models which will be exploited in order to obtain the targeted reference model.

4.2. Case studies "BPI challenge 2012"

Since we didn't find multiple existing event logs in the same domain, we opted to decompose an event log in order to get many logs that are in the same field. Thus we used the case study "BPI Challenge 2012". We have successfully applied our mining technique to this log event. The event log used for the BPI Challenge 2012 contains events related to the application process for a personal loan or overdraft within a Dutch financial institute. The event log contains 13087 cases and 262200 events distributed over 36 activities [9].

In order to test our plug-in, we decided to decompose the main event log "BPI challenge 2012" into three event logs. To do this, we used the plug-in "log filter using a simple heuristic." This plug-in was used to obtain event logs that contain events which were carefully chosen.

The different event logs can be classified according to the tree illustrated in Fig. 2.

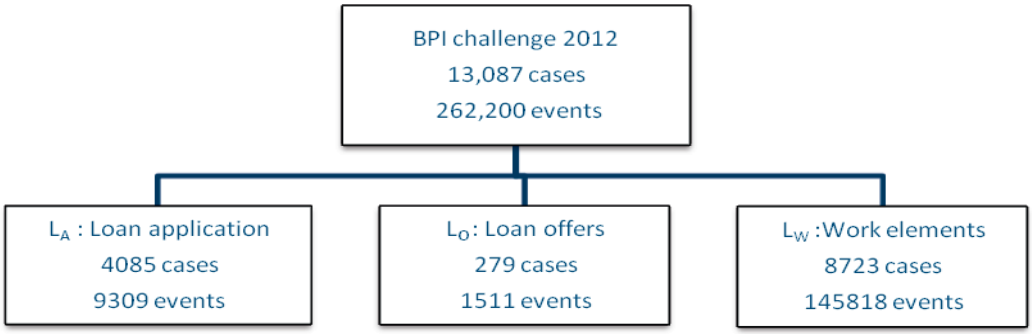


Fig. 2. Decomposition of the log file “BPI challenge 2012”

From modeling event log L_A (figure 3), we can deduce that:
The process begins with the submission of the initial application "A _PARTLY SUBMITTED”
Successful application ends with the following final tasks: "A_APPROVED ", "A_REGISTERED" or "A_ACTIVATED”.
The final tasks are preceded by a completed application "A_finalizedAcross."
The tasks "A_APPROVED" and "A_REGISTERED" are parallel.

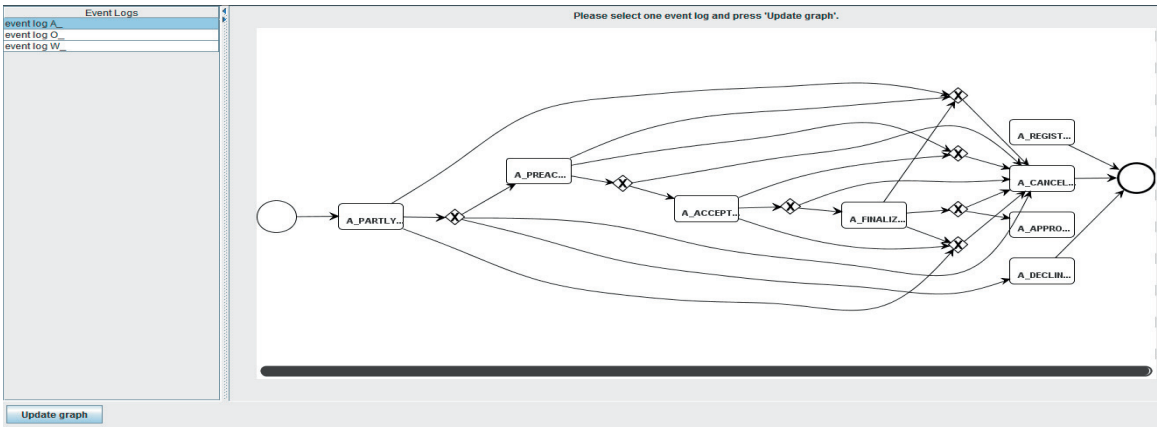


Fig. 3. Mined L_A BPMN model

Figure 4 shows the mined model L_O. We can deduce from modeling this event log that:
From the different offers, there are offers that were accepted, others that were rejected and others that were canceled. There are also offers that were abandoned ("O SENT" and "O SENT BACK").
The process begins with the receipt of the offer "O SELECTED".
After receiving the offer, it is either unsuccessful or created.

From modeling the event log L_w (Fig. 5), we can deduce that:

The mined model is complex. It contains a large number of tasks. The user can zoom on the model.
The process begins with either the monitoring of the initial incomplete submissions, or a fraud investigation.
There are several final tasks. The process can end with the completion of pre-accepted applications.

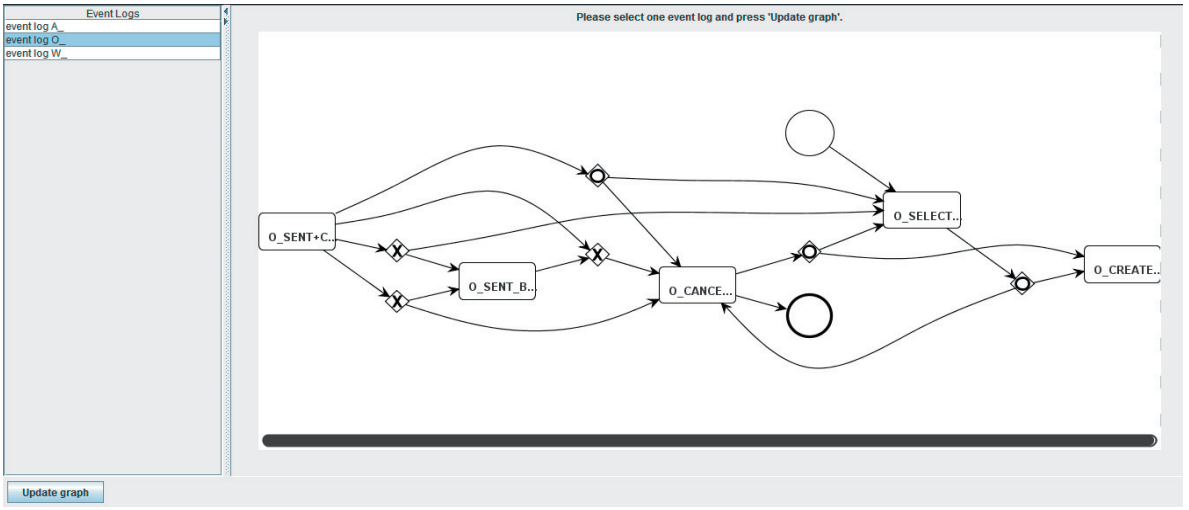


Fig. 4. Mined L_O BPMN model

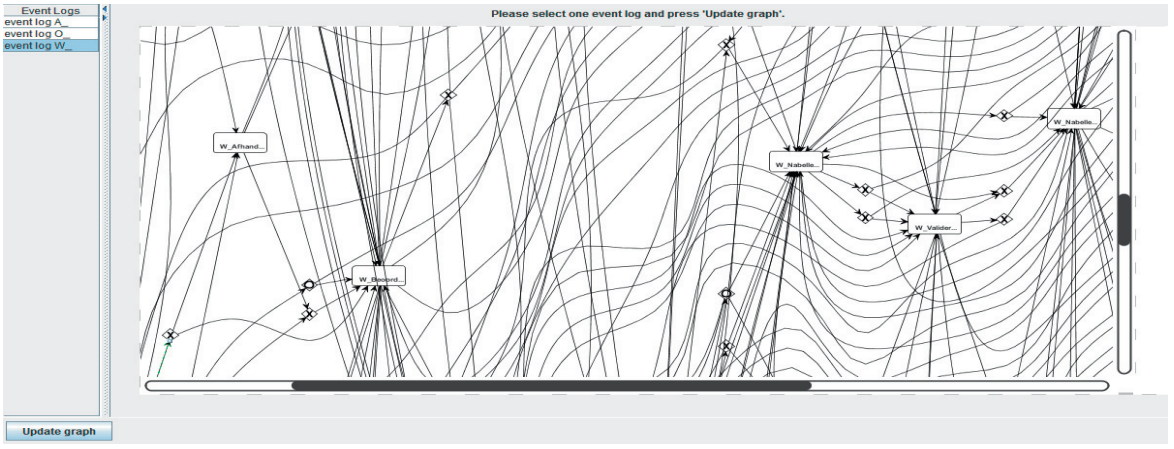


Fig. 5. Mined LW BPMN model

These same steps have to be applied in order to get process models when using logs that are extracted from processes from different organizations that are in the same domain.

5. Conclusion and future works

This paper presented the development and the description of the "multi alpha algorithm". This solution has two main originalities: the consideration of the multiple input event logs and the generation of a BPMN reference model by applying an adapted version of the alpha algorithm.

To implement this algorithm, we have developed a plug-in "extract BPMN models using multi-alpha algorithm". This plug-in has been implemented in ProM environment. To demonstrate the applicability of our solution, a case study has been carried out on event logs from different systems, but in the same domain. The results of these tests are encouraging.

In future research we have to setup experiments with larger real-world data to provide further guidance into the practical usage of the multi-α algorithm. We want also to improve our work by including in our study, all the elements of BPMN. We also plan to improve the multi-α algorithm to make it more efficient and able of giving results even if the event logs are very complex.

References

[1] Hornix, P., Performance analysis of business processes through process mining. Master's thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (2007)
[2] Van der Aalst, W. M.P, Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
[3] Jansen-Vullers, M.H., Van der Aalst, W.M.P., Rosemann, M., Mining Configurable Enterprise Information Systems. Data knowledge engineering 56 (3), pp. 195-244 (2006)
[4] DUCQ, Y., Evaluation de la performance d'entreprise par les modèles. Ph.D. thesis, L'université Bordeaux 1 (2007)
[5] Recker, J., Rosemann, M., van der Aalst, W., Jansen-Vullers, M., Dreiling, A., Configurable reference modeling languages. In : P. Fettke, P. Loos (eds.) Reference Modeling for Business Systems Analysis, pp. 22-46. IGI Global, Hershey, Pennsylvania (2006)
[6] Gottschalk, F., van der Aalst, W. M., & Jansen-Vullers, M. H., Mining Reference Process Models and Their Configurations. In On the Move to Meaningful Internet Systems: OTM 2008 Workshops, pp. 263-272, Springer Berlin Heidelberg (2008).
[7] Li, C., Reichert, M.,Wombacher, A., Discovering reference process models by mining process variants. IEEE 19th International Conference on Web Services , pp. 45-53 (2008).
[8] Recker, J.C., Bpmn modeling - who, where, how and why. BPTrends 5(3), pp. 1-8 (2008)
[9] 3TU.Datacentrum, http ://data.3tu.nl/repository/uuid :3926db30-f712-4394-aebc-75976070e91f