

Using Process Mining to Analyze and Improve Process Flexibility

- *Position Paper* -

Wil van der Aalst^{1,2}, Christian Günther¹, Jan Recker², and Manfred Reichert³

¹ Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{w.m.p.v.d.aalst, c.w.gunther}@tm.tue.nl

² Queensland University of Technology
126 Margaret Street, Brisbane QLD 4000, Australia
j.recker@qut.edu.au

³ University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
m.u.reichert@ewi.utwente.nl

Abstract. Contemporary information systems struggle with the requirement to provide flexibility and process support while still enforcing some degree of control. Workflow management systems are typically considered as too restrictive while groupware applications (e.g., e-mail) tend to offer hardly any process support or control at all. Therefore, we consider *adaptive process management systems* that allow for run-time changes at both process type and process instance level (e.g., a system like ADEPT). Assuming that these process changes are recorded explicitly, we discuss how this information can be used as input for *process mining*. So far, process mining has only been applied to operational processes, i.e., knowledge is extracted from event logs (process discovery), or event logs are compared with some a-priori model (conformance checking). In this paper, we propose to apply process mining to *change logs*, i.e., we not only analyze the operational processes but also the adaptations made at the process type or process instance level. Using such analysis, we envisage to be able to better support flexibility by understanding *when* and *why* process changes become necessary.

1 Introduction

The notion of flexibility has emerged as a pivotal research topic in Business Process Management (BPM) over the last years [1–4]. It denotes an essential requirement for maintaining a close “fit” between real-world business processes and the workflows as supported by Process Management Systems (PMSs) [5, 6], the current generation of which is known under the label of *Process-Aware Information Systems* (PAIS) [7]. Due to continuous and often unprecedented changes in their business environments [8], the alignment of business processes to PMSs requires continuous attention. The notion of flexibility plays a central role

in this context as not only business processes need to be designed for flexibility but also their supporting systems. In general, flexibility denotes the ability to yield to change without loss of identity [9]. Business process flexibility is the capability to react to external changes by modifying only those parts of a process that need to be changed and keeping other parts stable, viz., the ability to change the process without completely replacing it [1, 3]. Process flexibility can be classified according to three dimensions: *where* does change occur (abstraction level of change), *what* is being changed (subject of change), and *how* is it changed (properties of change) [10].

Recently, many efforts have been undertaken to make process management systems more flexible and several approaches for “adaptive” process management have emerged [11], most of which relate to some aspects of flexibility described in [10]. The basic idea behind them is to enable dynamic changes of different process aspects (e.g., control flow, organizational, functional, and informational perspectives) and at different process levels (e.g., instance and type level). For example, ad-hoc changes conducted at the process instance level (e.g., to add or shift steps) allow to flexibly adapt single process instances to exceptional or changing situations [12]. As a positive side-effect we obtain much more meaningful process logs when compared to existing PMSs.

Referring to the abstraction level of change, adaptive PMSs such as ADEPT [12, 13] offer flexibility at both process type level and process instance level [4]. However, so far adaptive PMSs have not addressed the fundamental questions what we can learn from this additional information and how we can derive optimized process models from it. Process mining techniques [14] offer promising perspectives in this context, but have focused on the analysis of pure execution logs (i.e., a behavioral and operational perspective) so far.

This position paper seeks to discuss approaches for the intelligent mining of flexible processes. This involves three problems: First, we have to determine which information about ad-hoc process deviations should be logged to achieve optimal mining results. Second, we have to develop advanced mining techniques which utilize execution and change logs. Third, we have to integrate these techniques with existing *adaptive* PMSs in order to provide full process life cycle support.

Section 2 introduces the state-of-the art of process mining. Section 3 then discusses the research problem of providing integrated support of flexible processes and process mining techniques. Therein we will propose and briefly elaborate on four dedicated research problems that need to be approached. Section 4 closes this paper by providing some outlook to future work.

2 Process Mining

Process-aware information systems, such as WfM, ERP, CRM and B2B systems, need to be configured based on process models specifying the order in which process steps are to be executed [15]. Creating such models is a complex and time-consuming task for which different approaches exist. The most traditional

one is to analyze and design the processes explicitly making use of a business process modeling tool. However, this approach has often resulted in discrepancies between the actual business processes and the ones as perceived by designers [16]; therefore, very often, the initial design of a process model is incomplete, subjective, and at a too high level.

Instead of starting with an explicit process design, process mining aims at extracting process knowledge from “process execution logs”. Process mining techniques such as the alpha algorithm [14] typically assume that it is possible to sequentially record events such that

- each event refers to an activity (i.e., a well-defined step in the process) and
- each event refers to a case (i.e., a process instance).

Moreover, there are other techniques using additional information such as

- the performer, also referred to as originator of the event (i.e., the person/resource executing or initiating the activity),
- the timestamp of the event, or
- data elements recorded with the event (e.g., the size of an order).

This information can be used to automatically construct process models. For example, the alpha algorithm [14] can construct a Petri net model describing the behavior observed in the log. The Multi-Phase Mining approach [17] can be used to construct an Event Process Chain (EPC) based on similar information. In the meantime there are mature tools such as the ProM framework (cf. Figure 1) to construct different types of models based on real process executions [18].

However, process mining research so far has mainly focused on issues related to control flow mining, i.e., a behavioral and operational perspective. Different algorithms and advanced mining techniques have been developed and implemented in this context (e.g., making use of inductive learning techniques or genetic algorithms). Tackled problems include concurrency and loop backs in process executions, but also issues related to the handling of noise (e.g., exceptions). Furthermore, some initial work regarding the mining of other model perspectives (e.g., organizational and informational perspectives) has been conducted [19]. For example, work on the mining case handling systems puts more emphasis informational perspective while social network mining techniques focus on the organizational perspective.

Figure 1 shows a screenshot of ProM. ProM [18] offers a wide variety of process mining techniques and also supports multiple types of models, e.g., it is possible to extract a process model represented in terms of an EPC, then convert this to a Petri net for analysis purposes (e.g., verification or simulation), and then translate the model to YAWL [20] such that the YAWL engine can immediately execute the discovered models. ProM also allows for the analysis of all perspectives, e.g., discovering social networks, building decision trees, conformance checking, etc. However, thus far, not specific extensions have been developed to mine for changes.

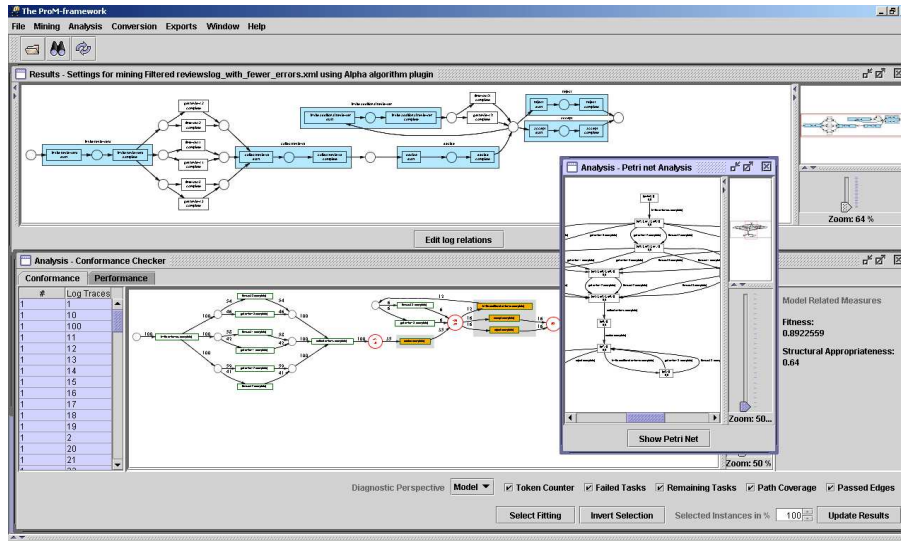


Fig. 1. Screenshot of ProM.

3 Using Process Mining for Improving Process Flexibility: A Research Statement

Obviously, the practical benefit of process mining depends on the content and the quality of the available log data. In WfMS, for example, respective execution logs can only reflect situations the WfMS is able to handle. Particularly, if the WfMS does not support process instance changes it needs to be bypassed in exceptional situations (e.g., by executing unplanned process activities outside the scope of the WfMS). Consequently, the WfMS is unaware of the applied deviations and thus unable to log (let alone analyze) information about them. This missing traceability limits the benefit of current process mining and delta analysis approaches significantly.

But how is the situation if we apply current process mining techniques to flexible or adaptive processes? Since the execution logs of flexible processes reflect parts of the applied instance changes we get at least more meaningful results. For example, for deleted process steps no related event will be recorded in the process execution log. By contrast, the execution of a newly inserted step will be reflected in the execution log (at least if this step was started or finished). Though the mining of respective execution logs is already more promising (when compared with rigid process executions) we cannot achieve optimal results as long as we ignore the available knowledge about the reasons and the context of a change. As a consequence, current process mining techniques are only able to capture the reactive part of process flexibility but not the stimulus for change. Furthermore, existing mining approaches mainly consider the control flow perspective, whereas

ad-hoc changes often constitute a mix of control and data flow adaptations (as well as adaptations of other process aspects) to preserve correctness.

In order to achieve integrated support for flexible processes and process mining techniques, several problems have to be addressed. In essence, we see three main challenges:

1. We have to find a way to effectively *discover* knowledge about process instance changes.
2. We have to find a way to effectively *use* knowledge about process instance changes.
3. We have to find a way to effectively *learn* from knowledge about process instance changes.

In order to be able to ultimately design process mining techniques for adaptive processes, first of all, we must better understand which kind of information is needed within the process execution logs. This triggers the first research question.

Problem 1 *What information about process changes is needed to achieve optimal mining results?*

When designing advanced mining techniques for flexible, adaptive processes one fundamental question is what type of information about changes needs to be recorded in order to achieve optimal results. Minimally, a *change log* should keep syntactical information about the kind and the context of the applied changes (e.g., the type and position of a dynamically inserted process activity). While this already helps to incorporate change knowledge into the mining process, it is by far not sufficient to handle more complex scenarios. One challenging problem is to enrich change logs with semantical information, i.e., *context information* (e.g., about the reasons for a change). For example, assume that in a patient treatment process an unplanned lab test is dynamically added for a considerable number of process instances. Then the respective log entries should reflect information about the semantical context of these changes (e.g., that insertions have been mainly applied for patients that are older than 60 years and who suffer from diabetes) in order to derive optimal process model adaptations (e.g., to embed the activity to be inserted in a conditional branching with respective transition conditions when bringing this change up to the process type level). Many challenging issues arise, e.g., how to gather the semantical information, how to represent it in the logs, how to make it efficiently accessible, etc.

There are more challenges related to Problem 1. For instance, how can such context information best be structured and described in change logs? What process changes require what type of context information? How do you treat noise in change logs?

Anticipating a solution to these and other related questions triggers the question of how process mining can be adapted to consider this additional data. Existing techniques are mainly limited to the control flow perspective, hence, Problem 2 arises:

Problem 2 *How to apply mining techniques to learn from flexible processes?*

A solution to this problem needs to be based on the combined analysis of process models, related execution logs and change logs. Syntactical information contained in change logs allows us to deduce adapted versions of the initial process models, and to categorize changes based on their structural effect on the process. From the combination of all versions of the process models and their respective execution logs, a process mining algorithm can detect bottlenecks and malfunctions. These problems can then be traced back to change operations that have potentially caused them, and thus provide means for the evaluation of changes. Context information from change logs will provide semantic information about the drivers of process changes, and can potentially be used to categorize these changes in a meaningful manner. The result of such process mining technique should be a categorized set of abstract changes, indicating those that have led to execution problems.

Whether or not existing process mining techniques can be adapted or extended, the mining of process change logs poses additional challenges that have to be tackled. First, mining techniques should examine change operations from all process-relevant perspectives (e.g., control flow, organizational, etc.). This is a precondition for achieving a comprehensive solution, which is able to capture the nature of changes adequately. Second, process mining procedures should consider the semantics of the applied change operations (e.g., add, delete or shift process steps) when comparing instance changes, determining their degree of overlap, etc. Third, semantical information about the context of a change should be taken into consideration as far as possible (cf. Problem 1). Fourth, we have to cope with variations regarding change definitions. Note that ad-hoc changes applied to a collection of process instances of the same type are usually performed by different users. Even if two ad-hoc changes deal with the same basic situation there can be variations regarding their definition, e.g., with respect to the level of granularity or the parameterization of the changes (e.g. the position an activity is added to). Depending on their level of experience users may perceive or perform a certain task in different ways. In a hospital, for example, junior physicians tend to document their work in a more fine-grained way when compared to senior physicians. By considering such factors as well, we could obtain different variants of the same process model, which leads to Problem 3:

Problem 3 *How to support the reuse of ad-hoc changes?*

Current systems do not provide any knowledge about previously applied process instance changes when a new ad-hoc deviation becomes necessary; viz., users have to define each ad-hoc change from scratch. This does not only lead to high efforts and lower user acceptance, but also to greater variability of change definitions and more noise (cf. Problem 1). The enrichment of process change logs with semantical information (cf. Problem 1) provides promising perspectives regarding the “reuse” of changes. In particular, we can apply semantical information together with other change data to build up *case bases* where each ad-hoc change is represented as a particular case. When a user wants to devi-

ate from a pre-modeled process in exceptional or unanticipated situations, he or she can make use of this process knowledge. Thereby, the system should allow either to define a new process instance change (i.e., to create a new case), or, if a similar situation has happened before, to reuse a previously defined ad-hoc change. A solution to this problem could substantially benefit from existing work on case-based reasoning. For example, YAWL supports the dynamic selection of “worklets” using ripple-down rules [21]. This selection could be guided by case-based reasoning. Note that an actor-driven reuse of cases also results in valuable information about the ad-hoc changes (e.g., How often was the respective change applied? How did users rate the respective case?), which can be considered by the mining algorithms to be developed as well.

Important challenges arise from the creation and management of such a case base. The cases need to be organized along the defined, flexible process models, their different versions, and the related process instances. We see an opportunity to derive a meaningful structure for such case bases from the aforementioned classification of process flexibility [10]. Furthermore, powerful interfaces for updating and querying case bases are needed. Among other requirements this calls for sophisticated algorithms for arguing about the similarity of ad-hoc changes, the similarity of the exceptions that triggered the change, the similarity of the change context, etc.

Problem 4 *How can an architecture be provided in which solutions for the aforementioned problems can be implemented?*

When providing adequate solutions for Problems 1–3 powerful concepts for the effective use of existing knowledge about process instance changes will evolve. Together these concepts make up a framework that enables the reuse of process knowledge when introducing new ad-hoc changes. It will provide techniques for learning from the process instance changes and for deriving optimized process models out of them. However, the practical implementation of these concepts in a coherent architecture, let alone an integration with existing PMS frameworks such as ADEPT [12] or ProM [18] is far from trivial. As an example consider the maintenance of the case bases mentioned above. When bringing process optimizations to the process type level, a solution to Problems 1-3 would already support the migration of running process instances to a new process model version. However, not only process models evolve over time, but also the case bases related to them, and consequently, cases have to be migrated to new process model versions as well. In particular, the new case base version should only keep semantic information that still is relevant for process instances of the newly created process model (and for changes of them). Similar to these issues, there are various challenges related to the design of a coherent, integrated architecture providing adaptive process mining of flexible business processes.

4 Conclusions

This paper reported on the role that process mining can play in providing comprehensive support for flexible business processes. We briefly examined current state-of-the-art of process mining approaches in light of the need for providing support for flexible (i.e., changing) processes. We investigated how *adaptive PMSs* can be engineered to maintain flexibility in business processes via advanced process mining. Such a proposal incorporates several research challenges, some of which we outlined and discussed briefly.

Our line of argumentation rests on the observation that business process flexibility is closely linked to, if not restrained by, the supporting PMS. We argue therefore that adaptability to flexible processes and changes within these processes must be regarded a compulsory design principle for PMSs. More specifically, we argue that the engineering of adaptive PMSs is dependent on the provision of syntactic and semantic context information in the form of *change logs*. This context knowledge will enable future PMS architectures to discover, use and learn from process changes in an intelligent manner, and will thus make them adaptive to flexible processes.

In this paper we did not discuss legal, ethical, and social issues related to process mining. Clearly, event logs can be used to systematically measure the performance of employees. When providing flexibility on the one hand and monitoring deviations on the other hand, workers may feel threatened by the system. Moreover, in some countries there are legal restrictions, e.g., the Dutch Personal Data Protection Act (“Wet Bescherming Persoonsgegevens”) which is based on a directive from the European Union [22–24]. Clearly these issues need to be addressed carefully. However, there are many ways to deal with these issues, e.g., explicit employee approval. Moreover, tools such as ProMimport allow for the automatic “anonymization” of data.

References

1. Bider, I.: Masking flexibility behind rigidity: Notes on how much flexibility people are willing to cope with. In Castro, J., Teniente, E., eds.: CAiSE05 Workshops, Porto, Portugal, FEUP (2005) 7–18
2. Bider, I., Regev, G., Soffer, P., eds.: Proceedings of the 5th Workshop on Business Process Modeling, Development and Support (BPMDs’04). Volume 2 of Caise’04 Workshops. Riga Technical University, Latvia (2004)
3. Soffer, P.: On the Notion of Flexibility in Business Processes. In Castro, J., Teniente, E., eds.: Proceedings of the CAiSE’05 Workshops. Vol. 1. FEUP, Porto, Portugal (2005) 35–42
4. Regev, G., Soffer, P., Schmidt, R.: Taxonomy of flexibility in business processes. (Input BPMDs’06 workshop) (2006)
5. van der Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. Requirements Engineering Journal **10** (2005) 198–211

6. Knoll, K., Jarvenpaa, S.L.: Information Technology Alignment or “Fit” in Highly Turbulent Environments: the Concept of Flexibility. In Ross, J.W., ed.: Computer Personnel Research Conference on Reinventing IS. ACM Press, Alexandria, Virginia (1994) 1–14
7. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M., eds.: Process Aware Information Systems: Bridging People and Software Through Process Technology. Wiley Publishing, Hoboken, New Jersey (2005)
8. Quinn, J.B.: Intelligent Enterprise: A Knowledge and Service Based Paradigm for Industry. Free Press, New York, New York (1992)
9. Regev, G., Wegmann, A.: A Regulation-Based View on Business Process and Supporting System Flexibility. In Castro, J., Teniente, E., eds.: Proceedings of the CAiSE05 Workshops. Vol. 1. FEUP, Porto, Portugal (2005) 91–98
10. Regev, G., Soffer, P., Schmidt, R.: Taxonomy of Flexibility in Business Processes. Input to the 7th Workshop BPMDS’06, <http://lamswww.epfl.ch/conference/bpmds06/taxbpflex>. (2005)
11. Rinderle, S., Reichert, M., Dadam, P.: Correctness Criteria For Dynamic Changes in Workflow Systems: A Survey. *Data and Knowledge Engineering* **50** (2004) 9–34
12. Reichert, M., Dadam, P.: ADEPTflex Supporting Dynamic Changes of Workflows Without Loosing Control. *Journal of Intelligent Information Systems* **10** (1998) 93–129
13. Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating Process Learning and Process Evolution A Semantics Based Approach. In van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management: 3rd International Conference BPM 2005. Volume 3649 of Lecture Notes in Computer Science. Springer, Nancy, France (2005) 252–267
14. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering* **16** (2004) 1128–1142
15. van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge, Massachusetts (2002)
16. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering* **47** (2003) 237–267
17. van Dongen, B.F., van der Aalst, W.M.P.: Multi-phase Process Mining: Building Instance Graphs. In Atzeni, P., Chu, W.W., Lu, H., Zhou, S., Ling, T.W., eds.: Conceptual Modeling ER 2004: 23rd International Conference on Conceptual Modeling. Volume 3288 of Lecture Notes in Computer Science. Springer, Shanghai, China (2004) 362–376
18. van Dongen, B.F., Alves de Medeiros, A.K., Verbeek, M., 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.: Applications and Theory of Petri Nets 2005. Volume 3536 of Lecture Notes in Computer Science. Springer, Berlin, Germany et al. (2005) 444–454
19. van der Aalst, W.M.P., Song, M.: Mining Social Networks: Uncovering Interaction Patterns in Business Processes. In Desel, J., Pernici, B., Weske, M., eds.: Business Process Management: Second International Conference BPM 2004. Volume 3080 of Lecture Notes in Computer Science. Springer, Potsdam, Germany (2004) 244–260
20. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. *Information Systems* **30** (2005) 245–275

21. Adams, M., Hofstede, A., Edmond, D., Aalst, W.: Facilitating Flexibility and Dynamic Exception Handling in Workflows. In Belo, O., Eder, J., Pastor, O., Falcao e Cunha, J., eds.: Proceedings of the CAiSE'05 Forum, FEUP, Porto, Portugal (2005) 45–50
22. CBP: College Bescherming persoonsgegevens (Dutch Data Protection Authority). <http://www.cbpweb.nl/index.htm> (2003)
23. Hulsman, B., Ippel, P.: Personeelsinformatiesystemen: De Wet Persoonsregistraties toegepast, Registratiekamer, The Hague (1994)
24. Sauerwein, L., Linnemann, J.: Guidelines for Personal Data Processors: Personal Data Protection Act, Ministry of Justice, The Hague (2001)