

Supporting Process Mining Workflows with RapidProM

R.S. Mans, W.M.P. van der Aalst, H.M.W. Verbeek

Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{r.s.mans,w.m.p.v.d.aalst,h.m.w.verbeek}@tue.nl

Abstract. Process mining is gaining more and more attention both in industry and practice. As such, the number of process mining products is steadily increasing. However, none of these products allow for composing and executing analysis workflows consisting of multiple process mining algorithms. As a result, the analyst needs to perform repetitive process mining tasks manually and scientific process experiments are extremely labor intensive.

To this end, we have RapidMiner 5, which allows for the definition and execution of analysis workflows, connected with the process mining framework ProM 6. As such any discovery, conformance, or extension algorithm of ProM can be used within a RapidMiner analysis process thus supporting process mining workflows.

1 Introduction

During last decade, process mining has become a mature technique for analyzing all kinds of business processes based on a so-called event log [3]. Not surprisingly, the number of process mining products has increased. However, for all products, the analysis steps need to be done in an ad-hoc fashion thereby losing the overview of all the steps that are done and their order. In other words, within the process mining domain there is currently *no support for the construction and execution of a workflow which describes all analysis steps and their order*.

Within the scientific workflow domain, special kinds of workflow management systems exist which are designed to compose and execute a series of computational or data manipulation steps (e.g. RapidMiner, KNIME, and Taverna). When applying scientific workflow concepts in the process mining field, several advantages can be realized. For example, comparable process mining analyses can be repeated by just one-click of a button and scientific experiments can be executed in an automated fashion. Furthermore, within several scientific workflow management systems many data and machine learning techniques are readily available. As such, different techniques can be easily combined for an end-to-end analysis.

To this end, we have integrated the process mining framework ProM 6 [4] within the scientific workflow management system RapidMiner 5. That is, process mining functionality is added as an *extension* to RapidMiner. In this paper, this extension, called *RapidProM*, is discussed in detail.

2 Defining and Executing Process Mining Workflows

We first provide a general introduction to RapidMiner. Then we present the extension which supports process mining analysis workflows.

RapidMiner is a software product allowing for advanced analytics, i.e. sophisticated quantitative methods (for example, statistics, descriptive and predictive data mining, simulation and optimization) to produce insights that traditional approaches to Business Intelligence (BI) are unlikely to discover [2]. Where BI focusses on querying and reporting combined with simple visualization techniques showing dashboards and scorecards, advanced analytics aims at automatically finding hidden patterns too complex for humans to find. Moreover, BI looks back at the past whereas advanced analytics also aims to provide predictions about the future. In addition, RapidMiner provides a GUI to design and execute an analytical pipeline. After execution the results can be inspected.

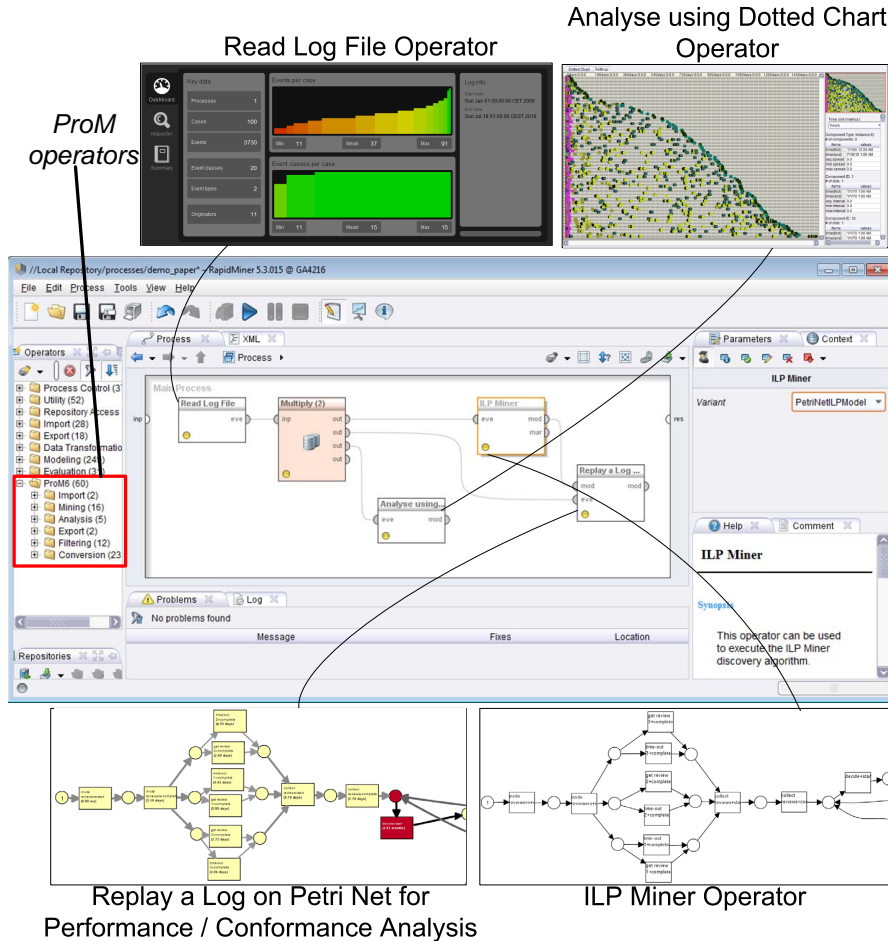


Fig. 1: A workflow in which several ProM 6 plug-ins are executed.

Table 1: For each subfolder a brief description is provided and some example operators are mentioned.

Subfolder	Description	Example Operators
Import	ProM objects are imported from file (e.g. an event log or a Petri net).	Read Log File and Read PNML file.
Mining	A process mining algorithm is executed. The algorithms may discover knowledge regarding the control-flow (e.g. a Petri net), organizational (e.g. a social network), and performance perspectives (e.g. a dotted chart).	ILP Miner, Passage Miner, and Inductive Miner.
Analysis	Typically an analysis is performed on a process mining result. For example, timing information is projected on a Petri net.	Replay a Log on Petri Net for Conformance Analysis and Repair Model.
Export	A process mining result is saved to disk.	Export Log and PNML Export.
Filtering	A filter is applied to an event log. For example, an artificial start and end event is added to each trace.	Add Artificial Start and End Event Filter and Add Noise Log Filter and Convert Process Tree into a Petri Net.
Conversion	One type of process mining result is converted into another type of process mining result.	Convert Process Tree into a Petri Net.

Looking at advanced analytics in general, one drawback is that processes are not made explicit. As a result, it is useful to extend RapidMiner with process mining capabilities. In the sequel, we will focus on the process mining capabilities that are available within the extension. In Figure 1 a screenshot of RapidMiner is provided thereby showing a process mining analysis within RapidMiner.

First, within the “Operators” panel at the left side all the operators that are available can be selected. There is a special “ProM6” folder in which all the process mining operators can be found. A description can be found in Table 1. Most of the operators correspond to existing ProM plug-ins.

In the “Process” panel in Figure 1 some of the available operators can be seen together with a visualization of the obtained results. First an event log is read (Read Log operator). Afterwards, a Petri net is discovered using the ILP miner (ILP Miner operator) and a dotted chart is created showing events in a graphical way such that a “helicopter overview” of the process is obtained (Analyse using Dotted chart operator). Finally, timing information is projected on the Petri net so that bottlenecks can be identified within the process (Replay a Log on Petri Net for Performance / Conformance Analysis operator).

A wide variety of workflows can be made. Some examples are:

- By using the Loop Parameters operator it is possible to iterate over a selection of operators for a set of parameter combinations. For example, the ILP Miner is repeated for each different option of its “Variant” parameter.

- For each item in a collection of objects, the same workflow can be executed. For example, the **Guide Tree Miner** operator provides a collection of logs. Subsequently, for each log the corresponding Petri net can be discovered.
- Within RapidMiner many data mining algorithms are available which can be used after converting the log into a feature set. For example, using the **Case Data Extractor** operator the log is converted into a feature set and subsequently a decision tree is obtained using the **Decision Tree** operator.
- Within RapidMiner also many statistical techniques are available. These can be used for evaluation of process mining experiments. For example, by using the **Loop Attributes** operator and the **Replay a Log on Petri Net for Performance / Conformance Analysis** operator, for a varying number of maximal states, the fitness between a log and a Petri net can be calculated. Afterwards, using the **Linear Regression** operator, the strength of the relationship between the fitness and the number of states can be determined.

3 Architecture and Implementation

In this section, we elaborate on the architecture of ProM 6 and RapidMiner and how both are connected. Furthermore, we focus on some implementation details.

To this end, in Figure 2 an architectural overview of RapidMiner and ProM 6 is given and how they are connected. First, the most important part of ProM 6 is the *framework* which roughly spoken contains all the necessary functionalities in order that process mining algorithms can be executed. The algorithms itself are provided by means of *packages*. A package may contain one or more *plug-ins* and a collection of *provided objects* that are needed or produced by the plug-in. Furthermore, a plug-in needs a *context* to run in. Depending on the type of context, the plug-in communicates in a different way with a user. Here, it is important that there is a clear separation between the actual process mining algorithm and the visualization of its results. Also, there is a clear separation between the plug-in and the parameter settings it needs. At the moment, two types of contexts are available: a GUI-aware context, called UITopia, and a headless context. So, a plug-in that is running in the UITopia context may communicate with the user through dialogs and/or wizards, whereas for a plug-in running in a headless context this is not obliged. So, the latter plug-ins can be ran using a *client*. In case such a plug-in requires parameter settings, these can be provided via its own input parameter object. Moreover, for the object that has been obtained, a visualization can be obtained by running the associated visualization plug-in.

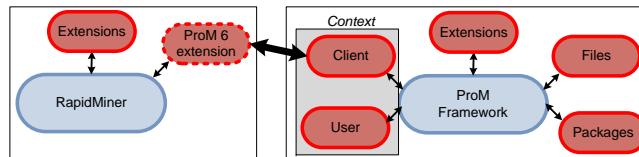


Fig. 2: Architectural overview of ProM and RapidMiner and how they are connected.

The headless context of ProM 6 is used in order that ProM plug-ins can be executed within RapidMiner. As can be seen in Figure 2, RapidMiner consists of a core in order that operators can be executed. New operators can be added by means of an extension together with the objects that are needed by the operators [1]. In order to do so, for each operator the algorithm needs to be defined and which objects it uses and produces. Furthermore, for each object it needs to be defined how it is visualized. In order to run ProM 6 plug-ins a special operator is created for each plug-in. For a provided object that needs to be visualized, simply the associated ProM visualizer is called. For example, regarding Figure 1, for the `ILP Miner` operator the `ILP Miner` plug-in of ProM is called and for the provided Petri net the associated `Visualize Petri Net` visualizer is called.

Currently, over 40 operators are available covering a selection of the plug-ins that are available within ProM 6. In case a plug-in runs in a headless context it can easily be added to the ProM extension. In order to speed up this process, we have developed a specific ProM plugin, called `RapidMiner Code Generator Plug-in`, that generates code for adding the plug-in.

The ProM extension has been tested for many scenarios. Furthermore, students following the “Advanced Process Mining” course at TU/e are using the software in order to do the assignments. In the end, we want to achieve that the extension becomes robust and mature in order that it can be successfully used by many people both from industry and science. Currently, the extension has been downloaded over 850 times at the RapidMiner marketplace¹.

4 Links

For the ProM extension of RapidMiner, a dedicated website is available: <http://www.rapidprom.org>. Amongst others, this website contains instructions for installing the extension, instructions for using it, it describes several use cases, and several screencasts.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs.

References

1. *How to Extend RapidMiner 5*. Rapid-I, 2012.
2. Gartner. Magic Quadrant for Advanced Analytics Platforms. 2014.
3. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.
4. H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst. ProM 6: The Process Mining Toolkit. In *Proc. of BPM Demonstration Track 2010*, volume 615, pages 34–39. CEUR-WS.org, 2010.

¹ <http://marketplace.rapid-i.com/>