XES Tools

H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
h.m.w.verbeek@tue.nl

Abstract. Process mining has emerged as a new way to analyze business processes based on event logs. These events logs need to be extracted from operational systems and can subsequently be used to discover or check the conformance of processes. ProM is a widely used tool for process mining. In earlier versions of ProM, MXML was used as an input format. In future releases of ProM, a new logging format will be used: the *eXtensible Event Stream* (XES) format. This format has several advantages over MXML. The paper presents two tools that use this format - *XESMa* and *ProM 6* - and highlights the main innovations and the role of XES. *XESMa* enables domain experts to specify how the event log should be extracted from existing systems and converted to XES. *ProM 6* is a completely new process mining framework based on XES and enabling innovative process mining functionality.

1 Introduction

Unlike classical process analysis tools which are purely model-based (like simulation models), process mining requires event logs. Fortunately, today's systems provide detailed event logs. Process mining has emerged as a way to analyze systems (and their actual use) based on the event logs they produce [1–4, 6, 15]. Note that, unlike classical data mining, the focus of process mining is on concurrent processes and not on static or mainly sequential structures. Also note that commercial Business Intelligence (BI for short) tools are not doing any process mining. They typically look at aggregate data seen from an external perspective (including frequencies, averages, utilization and service levels). Unlike BI tools, process mining looks "inside the process" and allows for insights at a much more refined level.

The omnipresence of event logs is an important enabler of process mining, as analysis of run-time behavior is only possible if events are recorded. Fortunately, all kinds of information systems provide such logs, which include classical workflow management systems like FileNet and Staffware, ERP systems like SAP, case handling systems like BPM|one, PDM systems like Windchill, CRM systems like Microsoft Dynamics CRM, and hospital information systems like Chipsoft). These systems provide very detailed information about the activities that have been executed.

However, also all kinds of embedded systems increasingly log events. An embedded system is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Examples include medical systems like X-ray machines, mobile phones, car entertainment systems, production systems like wafer steppers, copiers, and sensor networks. Software plays an increasingly important role in such systems and, already today, many of these systems log events. An example is the "CUSTOMerCARE Remote Services Network" of Philips Medical Systems (PMS for short), which is a worldwide internet-based private network that links PMS equipment to remote service centers. Any event that occurs within an X-ray machine (like moving the table or setting the deflector) is recorded and can be analyzed remotely by PMS. The logging capabilities of the machines of PMS illustrate the way in which embedded systems produce event logs.

The MXML format [7] has proven its use as a standard event log format in process mining. However, based on practical experiences with applying MXML in about one hundred organizations, several problems and limitations related to the MXML format have been discovered. One of the main problems is the semantics of additional attributes stored in the event log. In MXML, these are all treated as string values with a key and have no generally understood meaning. Another problem is the nomenclature used for different concepts. This is caused by MXML's assumption that strictly structured process would be stored in this format [10].

To solve the problems encountered with MXML and to create a standard that could also be used to store event logs from many different information systems directly, a new event log format is under development. This new event log format is named XES, which stands for eXtensible Event Stream. Please note that this paper is based on XES definition version 1.0, revision 3, last updated on November 28, 2009. This serves as input for standardization efforts by the IEEE Task Force Process Mining [13]. Minor changes might be made before the final release and publication of the format.

The remainder of this paper is organized as follows. Section 2 introduces the new event log format XES. Of course, we need to be able to extract XES event logs from arbitrary information systems in the field. For this reason, Section 3 introduces the XES Mapper tool. This tool can connect to any ODBC database, and allows the domain expert to provide the details of the desired extraction in a straightforward way. After having obtained an XES event log, we should be able to analyze this log in all kinds of ways. For this reason, Section 4 introduces ProM 6, which is the upcoming release of the ProM framework [8]. ProM 6 supports the XES event log format, and provides a completely new process mining framework. Finally, Section 5 concludes the paper.

2 XES: eXtensible Event Stream

Fig. 1 shows the XES meta model, which is taken from [11]. In XES the log, trace and event objects only define the structure of the document: they do not contain

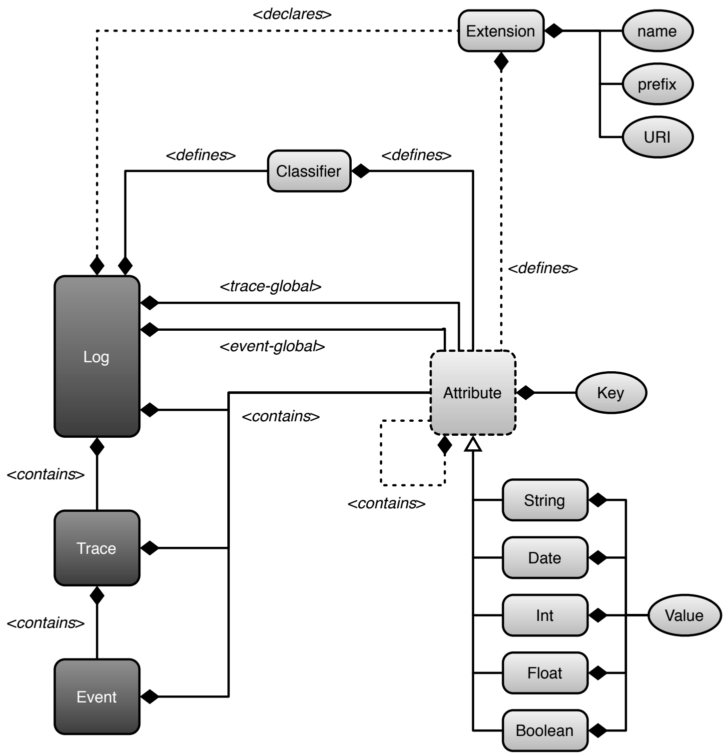

Fig. 1. XES Meta Model.

any information themselves. To store any information, attributes are used. Every attribute has a string based key and a value of some type. Possible value types are string, date, integer, float and boolean. Note that attributes can have attributes themselves which can be used to provide more specific information.

The precise semantics of an attribute is defined by its extension, which could be either a standard extension or some user-defined extension. Standard extensions include the *concept* extension, the *lifecycle* extension, the *organizational* extension, the *time* extension, and the *semantic* extension. Table 1 shows an overview of these extensions together with a list of possible keys, the level on which these keys may occur, the value type, and a short description. Note that the semantic extension is inspired by SA-MXML (Semantically Annotated MXML) [14].

Furthermore, *event classifiers* can be specified in the log object which assign an identity to each event. This makes events comparable to other events via their assigned identity. Classifiers are defined via a set of attributes, from which the class identity of an event is derived. A straightforward example of a classifier is the combination of the event name and the lifecycle transition as used in MXML.

Table 1. List of XES extensions and the attribute keys they define.

Extension	Key	Level	Type	Description
Concept	name	log, trace, event	string	Generally understood name.
	instance	event	string	Identifier of the activity whose execution generated the event.
Lifecycle	model	log	string	The transactional model used for the lifecycle transition for all events in the log.
	transition	event	string	The lifecycle transition represented by each event (e.g. start, complete, etc.).
Organizational	resource	event	string	The name, or identifier, of the resource having triggered the event.
	role	event	string	The role of the resource having triggered the event, within the organizational structure.
	group	event	string	The group within the organizational structure, of which the resource having triggered the event is a member.
Time	timestamp	event	date	The date and time, at which the event has occurred.
Semantic	modelReference	all	string	Reference to model concepts in an ontology.

3 XES Mapper

Although many information systems record the information required for process mining, chances are that this information is not readily available in the XES format. Since the information is present in the data storage of the information system, it should be possible to reconstruct an event log that contains this information. However, extracting this information from the data storage is likely to be a time consuming task and requires domain knowledge, knowledge which is usually held by domain experts like business analysts.

For the purpose of extracting an event log from an information system, the ProM Import Framework [9] was created. Although there is a collection of plug-ins for various systems and data structures, chances are that a new plug-in needs to be written by the domain expert in Java. The main problem with this approach is that one cannot expect the domain expert to have Java programming skills. Therefore, there is a need for a tool that can extract the event log from the information system at hand without the domain expert having to program. This tool is the XES Mapper [5], or *XESMa* for short.

We use an example to explain XESMa. From some company, we received a database export in the form of thirteen CSV (Comma Separated Values) tables. From the thirteen tables, only two were required for the event log extraction.

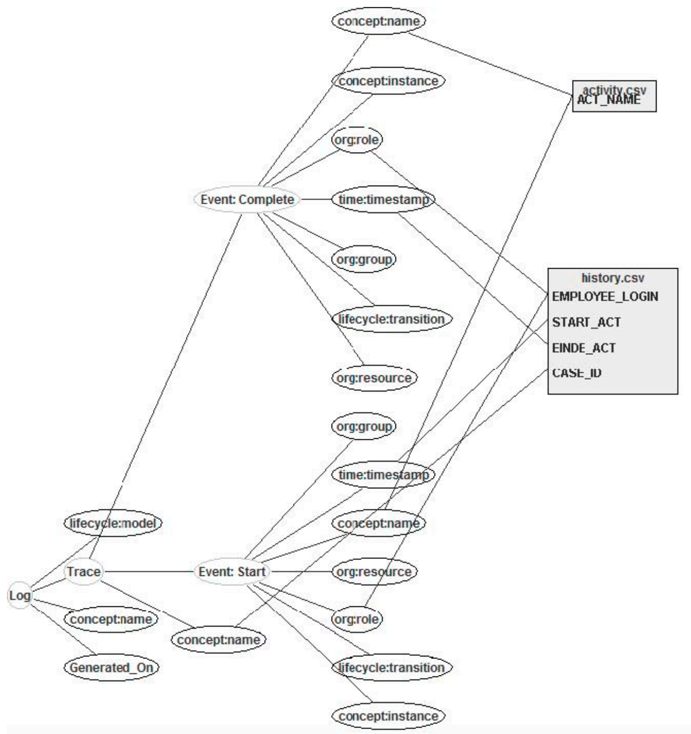

Fig. 2. Mapping visualization.

The first table (history.csv) contains 19,223,294 records, measures 2.14 GB and holds the history of all activities performed in the year 2008, while the second table (activity.csv) contains 811 records, measures 45 KB and holds additional information on the tasks defined in the system.

First, the domain expert needs to tell XESMa how the event log should be extracted from both tables. Fig. 2 shows the visual representation of this mapping. The left-hand side of Fig. 2 shows a log, a trace, two events, and their attributes, whereas the right-hand side shows both tables. The lines from the attributes to the tables indicate how the actual value for this attribute is extracted from the tables. As an example, the time:timestamp attribute of a Start event will be extracted from the START_ACT field of the history.csv table. Note that although we only have two events in the mapping, the resulting event log will contain almost 40 million events as for every record from the history.csv table both a Start event and a Complete event will be generated, and that although we only have a single trace, the resulting log will contain as many traces as the history.csv table contains different values for the CASE_ID field.

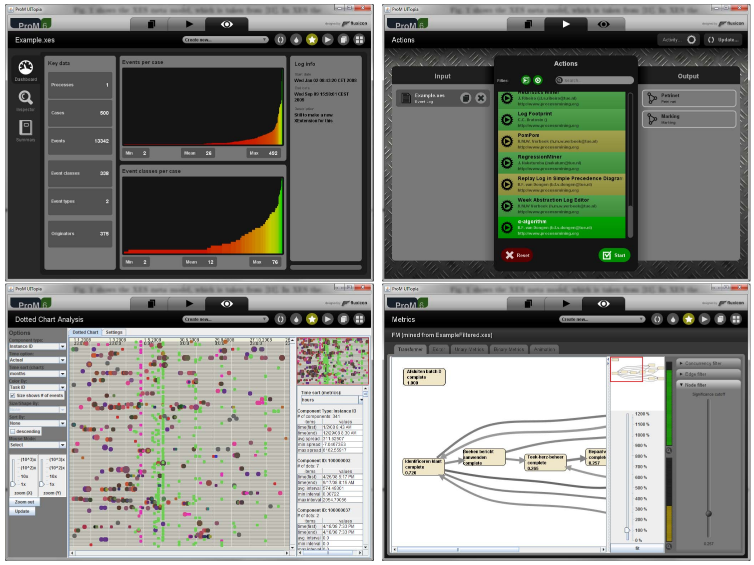

Fig. 3. ProM 6 results.

4 ProM

After having extracted the event log from the information system, we can analyze the event log using ProM [8], the plugable generic open-source process mining framework. As XES is a new log format that is still under development, the older versions of ProM do not handle XES logs. Fortunately, the upcoming version of ProM, ProM 6, will be able to handle XES logs. ProM 6 will be released in the Summer of 2010, but interested readers may already obtain so-called 'nightly builds' through the Process Mining website (www.processmining.org).

The fact that ProM 6 can handle XES logs where earlier versions of ProM cannot is not the only difference between ProM 6 and its predecessors (ProM 5.2 and earlier). Although these predecessors have been a huge success in the process mining field, they limited future work for a number of reasons. First and foremost, the earlier versions of ProM did not separate the functionality of a plug-in and its GUI. As a result, a plug-in like the α-miner [3] could not be run without having it popping up dialogs. As a result, it was impossible to run the plug-in on some remote machine, unless there would be somebody at the remote display to deal with these dialogs. Since we are using a dedicated process grid for process mining, this is highly relevant. Second, the distinction between the different kind of plug-ins (mining plug-ins, analysis plug-in, conversion plug-ins, import plug-ins, and export plug-ins) has disappeared; leaving only the concept of a generic plug-in. Third, the concept of an object pool has been introduced: plug-ins take a number of objects from this pool as input, and produce new objects for this pool. Fourth, ProM 6 allows the user to first select a plug-in, and then select the necessary input objects from the pool. As some plug-in can handle different configurations of objects as input, ProM 6 also introduces the concept of plug-in variants. The basic functionality of variants of some plug-in will be identical, but every variant will be able to take a different set of objects as input.

We use a selection of the XES event log obtained from XESMa, as described in the previous section, to showcase ProM 6. Fig. 3 shows some results obtained. The left upper view shows some basic characteristics of the log, like the number of traces, number of events, and distribution of trace length. The right upper view shows the list of installed plug-ins with the α-miner selected. On the left-hand side of this view the necessary inputs for this plug-in are shows, while on the right-hand side the expected outputs are shown. Note that ProM is aware of these inputs and outputs, which allows us to chain series of plug-ins into workflows to conduct larger process mining experiments. The left bottom view shows a dotted chart [16] on a filtered part of the log, whereas the right bottom view shows the result of the fuzzy model [12] mined from this filtered log.

5 Conclusions

This paper has introduced the new event log format XES. The XES format enhances the existing MXML [7] in many ways, as is shown in this paper. XES is used as input for standardization efforts within the IEEE Task Force on Process Mining [13].

This paper also introduced a tool that allows the domain expert to extract an XES event log from some existing system. This tool, XESMa [5], improves on the ProM Import framework [9] in the way that it is generic, and that it does not require the domain expert to create a Java plug-in for doing the extraction. Instead, XESMa allows the domain expert to simply specify from which fields in the database which attributes in the event log should be extracted.

Finally, this paper has introduced a new version of the ProM framework [8], ProM 6. In contrast to earlier versions of ProM, ProM 6 can handle XES event logs, can be executed on remote machines, and can guide the user into selecting the appropriate inputs for a certain plug-in. As a result, it better supports the analysis of event logs than any of the earlier releases did.

Acknowledgements

The authors would like to thank Christian Günther for his work on the XES standard and the new UI of ProM 6.

References

1. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.

2. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
3. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
4. R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.
5. J.C.A.M. Buijs. Mapping Data Sources to XES in a Generic Way. Master's thesis, Eindhoven University of Technology, 2010.
6. A. Datta. Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches. *Information Systems Research*, 9(3):275–301, 1998.
7. B.F. van Dongen and W.M.P. van der Aalst. A Meta Model for Process Mining Data. In J. Casto and E. Teniente, editors, *Proceedings of the CAiSE'05 Workshops (EMOI-INTEROP Workshop)*, volume 2, pages 309–320. FEUP, Porto, Portugal, 2005.
8. B.F. van Dongen, A.K. Alves de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A New Era in Process Mining Tool Support. In G. Ciardo and P. Darondeau, editors, *Application and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer-Verlag, Berlin, 2005.
9. C. Günther and W.M.P. van der Aalst. A Generic Import Framework for Process Event Logs. In J. Eder and S. Dustdar, editors, *Business Process Management Workshops, Workshop on Business Process Intelligence (BPI 2006)*, volume 4103 of *Lecture Notes in Computer Science*, pages 81–92. Springer-Verlag, Berlin, 2006.
10. C. W. Günther. *Process Mining in Flexible Environments*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2009.
11. C. W. Günther. XES Standard Definition. Fluxicon Process Laboratories, November 2009.
12. C.W. Günther and W.M.P. van der Aalst. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer-Verlag, Berlin, 2007.
13. IEEE Task Force on Process Mining. www.win.tue.nl/ieeetfpm.
14. A.K. Alves de Medeiros, C. Pedrinaci, W.M.P. van der Aalst, J. Domingue, M. Song, A. Rozinat, B. Norton, and L. Cabral. An Outlook on Semantic Business Process Mining and Monitoring. In R. Meersman, Z. Tari, and P. Herrero, editors, *Proceedings of the OTM Workshop on Semantic Web and Web Semantics (SWWS '07)*, volume 4806 of *Lecture Notes in Computer Science*, pages 1244–1255. Springer-Verlag, Berlin, 2007.
15. A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95, 2008.
16. M. Song and W.M.P. van der Aalst. Supporting Process Mining by Showing Events at a Glance. In K. Chari and A. Kumar, editors, *Proceedings of 17th Annual Workshop on Information Technologies and Systems (WITS 2007)*, pages 139–145, Montreal, Canada, December 2007.