Process Mining and Monitoring Processes and Services: Workshop Report

Wil van der Aalst (editor)

Eindhoven University of Technology, P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tm.tue.nl

Abstract. In a service-oriented architecture, but also in classical enteprise systems, a variety of events (e.g., messages being sent and received) are being logged. This information can be used for *process mining* purposes, i.e., based on some event log it is possible to *discover* processes or to *check conformance*. The goal of process discovery is to build models without a-priori knowledge, i.e., based on sequences of events one can look for the presence or absence of certain patterns and deduce some process model from it. For conformance checking there has to be an initial model. One can think of this model as a "contract" or "specification" and it is interesting to see whether the parties involved stick to this model. Using conformance checking it is possible to quantify the fit (fewer deviations result in a better fit) and to locate "problem areas" where a lot of deviations take place.
One of the four workshops organized within the context of the Dagstuhl seminar on *The Role of Business Processes in Service Oriented Architectures* (Seminar 06291, 16.07.2006-21.07.2006) was a Workshop on Process Mining and Monitoring Processes and Services. In this paper, we report on the results of the workshop.

1 Introduction

For an overview of process mining techniques, the reader is referred to [4] and [3]. Process mining can be seen in the broader context of Business Process Intelligence (BPI) and Business Activity Monitoring (BAM). The need for monitoring web services has been identified by many users, software developers and researchers. For example, several research groups have been experimenting with adding monitor facilities via SOAP monitors in Axis. [11] introduces an assertion language for expressing business rules and a framework to plan and monitor the execution of these rules. [6] uses a monitoring approach based on BPEL. Monitors are defined as additional services and linked to the original service composition. Another framework for monitoring the compliance of systems composed of web-services is proposed in [13]. This approach uses event calculus to specify requirements. [12] is an approach based on WS-Agreement defining the Crona framework for the creation and monitoring of agreements. In [9, 8], Dustdar et al. discuss the concept of web services mining and envision various levels (web

Dagstuhl Seminar Proceedings 06291
The Role of Business Processes in Service Oriented Architectures
http://drops.dagstuhl.de/opus/volltexte/2006/834

service operations, interactions, and workflows) and approaches. Our approach fits in their framework and shows that web-services mining is indeed possible. In [14] a tool named the Web Service Navigator is presented to visualize the execution of web services based on SOAP messages. The authors use message sequence diagrams and graph-based representations of the system topology. In [1] to the monitoring of web services is discussed and concrete techniques based on BPEL and SOAP messages in the context of ProM are presented in [2].

To goal of the workshop was to discuss the various forms of process mining both from a technological (Is it possible to log the right events? Are there mature techniques available?) and a requirements (Which types of process mining are particularly useful?) point of view.

2 Questions

Within the context sketched above and practical experiences with the monitoring and analysis of logs from a variety of systems (e.g., SAP and Websphere), the following questions were raised:

1. How to obtain useful audit trails/event logs?
2. How to deal with correlation?
3. How to design for monitoring?
4. How to classify the various techniques for process mining/BAM/BPI/... ?
5. Which are the most relevant questions/techniques?
6. How to present management information (cf. Business Cockpits, abstraction, viz. etc.)?
7. Does the "I" in "BPI" stand for "Intelligence" or "Ignorance"?

These questions were discussed in detail and the main findings are reported in the remainder.

3 Obtaining Event Logs

Based on a discussion of various systems, the conclusion was that there are basically two types of systems: those that are process-aware and those that are not. *In process-aware information systems the right event logs are available or can be obtained easily.* Process-aware systems are systems that driven by process models, e.g., workflow management systems, case handling systems, and the various execution/choreography engines in the context of webservices. *In systems that are not process aware, i.e., an explicit process model is missing, it may be more difficult to obtain the right logs.* For example, ERP systems like SAP and PeopleSoft have not been designed around explicit processes. Instead, these systems are built around a large set of database tables. Information about a purchase order may be scattered around various tables recording different aspects at different levels. Typically, extensive domain knowledge is needed to link these tables to concrete processes. Hence, it is often difficult and labor-intensive to collect the right event logs. Sometimes, however, it is easier to collect logs in systems that are not process-aware, e.g., in a hospital information systems it is often easy to link things together using the ID of the patient.

Despite problems with existing systems (e.g., ERP systems), it seems relatively easy to add the right logging functions when building a new systems. Moreover, many systems are being built on a process layer, making logging at the right level a standard functionality. We would even like to claim that *logging should be first class citizen* for any system that is used to support business processes! This is also the case in the context of auditing; without the right logs it is difficult to measure compliance and the see how processes really unfolded. New legislation (e.g., SOX and other laws enforcing minimal standards for auditing) will probably force software vendors and user organizations to devote more attention to the logging of events.

During the workshop it was noted that the topic of event logs is closely related to the correlation of (SOAP) messages. Just like an engine needs to know which messages are relevant for a particular process instance, process mining is only possible if events are linked to the right corresponding process instances. It was also noted that it may be useful to consider *trees of process instance ID's*, i.e., a process instance may consist of smaller process instances and this structure may be used for process mining. For example, a purchase order may consist of multiple order lines and multiple shipments. The ID's of an order line or shipment needs to be linked to a unique purchase order. Clearly, these ID's are logical identifiers rather than physical ID's. For example, in BPEL one can define so-called "correlation sets" for each conversation. Clearly, higher level concepts should be identified here to avoid an ah-hoc bookkeeping of dependencies between the different ID's. Note that simplifying issues related to correlation is both relevant for the enactment or processes and the monitoring of processes (i.e., generating meaningful event logs).

4 Classification of Process Mining Techniques

After establishing the way in which process logs can be obtained, the workshop focussed on a classification of the various process mining techniques. The following dimensions were discussed:

1. The presence of an a-priori model:
   - *Discovery*: There is no a-priori model, i.e., based on an event log some model is constructed. For example, using the alpha algorithm [5] a process model can be discovered based on low-level events.
   - *Conformance*: There is an a-priori model. This model is used to check if reality conforms to the model. For example, there may be a process model indicating that purchase orders of more than 1 million euro require two checks. Another example is the checking of the four-eyes principle. Conformance checking may be used to detect deviations, to locate and explain these deviations, and to measure the severity of these deviations.

  - *Extension*: There is an a-priori model. This model is extended with a new aspect or perspective, i.e., the goal is not to check conformance but to enrich the model. An example is the extension of a process model with performance data, i.e., some a-priori process model is used to project the bottlenecks on.
2. Functional/non-functional:
  - *Functional*: Questions related to "How?", "What?", and "Who?" are considered, i.e., the focus is on the structuring of the various perspectives.
  - *Non-functional*: Questions related to "How often?" and "How frequent?", i.e., the focus is on the way the functional properties are realized.
3. The functional/non-functional properties under investigation may be further classified into perspectives:
  - Functional perspectives:
    - *Control-flow perspective*: The ordering of activities.
    - *Data perspective*: The data relevant for the handling of process instances.
    - *Resource/organizational perspective*: The resources, i.e., people or other capacitated resources, involved in the process.
    - ...
  - Non-functional perspectives:
    - *Performance/time perspective*: Typical key performance indicators such as flow time, utilization, service level, etc.
    - *Quality perspective*: Non-functional properties not linked to classical key performance indicators but to quality measures, e.g., the number of failures, near-failures, customer satisfaction, etc.
    - ...
4. Number of process instances (i.e., cases) involved:
  - *Single instance*: The focus is on a single process instance, e.g., constructing a model for only one instance.
  - *Multiple instances*: The focus is on a selected group of process instances, e.g., checking the conformance of all purchase orders handled by a particular employee.
  - *All instances*: The focus is on all instances of a given process.
5. Time period considered:
  - *History*: A particular time period in the past is considered, e.g., all recorded events in the last year.
  - *Real-time*: The focus is on the current situation. Note that in this case still historic information may be used, e.g., to compute averages or to reconstruct the current state. However, the goal is to analyze the current situation and not the "typical" behavior of the process.
6. Result type:
  - *Inform*: The analysis is merely used to inform the people involved (e.g., managers and decision makers), i.e., the findings are not used to automatically take action. Note that managers and decision makers may use this information to control the process differently or to even redesign the process. Note that process mining results may be used for an "intelligent" redesign of the process, i.e., computer-assistent redesign driven by redesign rules and diagnostic information.

– *Act*: Based on the findings an action is performed, e.g., allocating resources to a process with a degrading service level or reducing the number of checks in case of fewer violations. Note that we only consider actions related to the management and control of the process being analyzed. The execution of operational activities in this process is not the task of process mining.

Of the six dimensions given, the third is clearly a refinement of the second one. Hence we will only consider dimensions 1 and 3-6. These five dimensions are orthogonal, for each combination it is possible to come up with meaningful examples. Some examples are given below.

The alpha algorithm [5] constructs a process model expressed in term of a Petri net. This can be classified using the five dimensions: *Discovery*, *Control-flow perspective*, *All instances* (or *Multiple instances*), *History*, and *Inform*.

The conformance checker in ProM [7] "replays" a log in a given process model and detect and measures the violations. The checker can also be classified using the five dimensions: *Conformance*, *Control-flow perspective*, *All instances* (or *Multiple instances*), *History*, and *Inform*.

The social network analyzer in ProM [7] tries to establish relationships between the various actors in the process (e.g., how many times does one employee pass on work to another employee). The social network analyzer can characterized as: *Discovery*, *Resource/organizational perspective*, *All instances* (or *Multiple instances*), *History*, and *Inform*.

The performance analysis plug-in of ProM [7] projects performance data (flow times, waiting times, service times, etc.) on a given process model. The plug-in can characterized as: *Extension*, *Performance/time perspective*, *All instances* (or *Multiple instances*), *History*, and *Inform*.

The prediction engine of Staffware [15] tries to predict when a process instance (i.e., case) is finished by combining historic information with the current state of the workflow. The feature can characterized as: *Extension*, *Performance/time perspective*, *single instance*, *Real-time*, and *Inform*.

Most of the functionality of ARIS PPM [10] is related to performance analysis based on historic information using an a-priori process model. This basic functionality can characterized as: *Extension*, *Performance/time perspective*, *All instances* (or *Multiple instances* when drilling down), *History*, and *Inform*.

None of the examples given can be classified as *Act*. However, it is easy to think of scenarios where the information generated is actually used to automate actions related to process management. It appears that the five dimensions (dimensions 1 and 3-6) are orthogonal. This leads to at least 3*5*3*2*2 = 180 different classes of process mining techniques. Although not explored in detail, it seems that each of these 180 classes is potentially meaningful and useful. This illustrates the broadness of the process mining domain.

5 Conclusion

During the workshop we concluded that it is desirable and possible to generate high-quality event logs. These event logs can be used to enable many different process mining techniques. Using five dimensions, 180 different classes of process mining techniques have been identified. Clearly, most of the existing commercial systems support merely a few of these 180 different classes. Moreover, process mining is a relatively immature research domain. Therefore, research and development should aim at a better coverage of the 180 classes. This is really needed because some vendors claim to have tools that support "Business Process Intelligence (BPI)" while their tools are not always very intelligent. In fact, one could even say that BPI stands for "Business Process Ignorance" when considering current practise.

References

1. W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, N. Russell, H.M.W. Verbeek, and P. Wohed. Life After BPEL? In M. Bravetti, L. Kloul, and G. Zavattaro, editors, *WS-FM 2005*, volume 3670 of *Lecture Notes in Computer Science*, pages 35–50. Springer-Verlag, Berlin, 2005.
2. W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, and H.M.W. Verbeek. Choreography Conformance Checking: An Approach based on BPEL and Petri Nets (extended version). BPM Center Report BPM-05-25, BPMcenter.org, 2005.
3. 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.
4. W.M.P. van der Aalst and A.J.M.M. Weijters, editors. *Process Mining*, Special Issue of Computers in Industry, Volume 53, Number 3. Elsevier Science Publishers, Amsterdam, 2004.
5. 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.
6. L. Baresi, C. Ghezzi, and S. Guinea. Smart Monitors for Composed Services. In *ICSOC '04: Proceedings of the 2nd International Conference on Service Oriented Computing*, pages 193–202, New York, NY, USA, 2004. ACM Press.
7. B. 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.
8. S. Dustdar, R. Gombotz, and K. Baina. Web Services Interaction Mining. Technical Report TUV-1841-2004-16, Information Systems Institute, Vienna University of Technology, Wien, Austria, 2004.
9. R. Gombotz and S. Dustdar. On Web Services Mining. In M. Castellanos and T. Weijters, editors, *First International Workshop on Business Process Intelligence (BPI'05)*, pages 58–70, Nancy, France, September 2005.
10. IDS Scheer. ARIS Process Performance Manager (ARIS PPM): Measure, Analyze and Optimize Your Business Process Performance (whitepaper). IDS Scheer, Saarbruecken, Gemany, http://www.ids-scheer.com, 2002.

11. A. Lazovik, M. Aiello, and M. Papazoglou. Associating Assertions with Business Processes and Monitoring their Execution. In *ICSOC '04: Proceedings of the 2nd International Conference on Service Oriented Computing*, pages 94–104, New York, NY, USA, 2004. ACM Press.
12. H. Ludwig, A. Dan, and R. Kearney. Crona: An Architecture and Library for Creation and Monitoring of WS-agreements. In *ICSOC '04: Proceedings of the 2nd International Conference on Service Oriented Computing*, pages 65–74, New York, NY, USA, 2004. ACM Press.
13. K. Mahbub and G. Spanoudakis. A Framework for Requirents Monitoring of Service Based Systems. In *ICSOC '04: Proceedings of the 2nd International Conference on Service Oriented Computing*, pages 84–93, New York, NY, USA, 2004. ACM Press.
14. W. De Pauw, M. Lei, E. Pring, L. Villard, M. Arnold, and J.F. Morar. Web Services Navigator: Visualizing the Execution of Web Services. *IBM Systems Journal*, 44(4):821–845, 2005.
15. Staffware. *Staffware Process Suite Version 2 – White Paper*. Staffware PLC, Maidenhead, UK, 2003.