Process Mining in Web Services: The WebSphere Case

W.M.P. van der Aalst and H.M.W. Verbeek
Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{w.m.p.v.d.aalst,h.m.w.verbeek}@tue.nl

Abstract

Process mining has emerged as a way to discover or check the conformance of processes based on event logs. This enables organizations to learn from processes as they really take place. Since web services are distributed over autonomous parties, it is vital to monitor the correct execution of processes. Fortunately, the "web services stack" assists in collecting structured event logs. This information can be used to extract new information about service processes (e.g., bottlenecks, unused paths, etc.) and to check the conformance (e.g., deviations from some predefined process). In this paper, we illustrate the potential of process mining in the context of web services. In particular, we show what a process mining tool like ProM can contribute in IBM's WebSphere environment.

1 Introduction

In a *Service Oriented Architecture* (SOA) services are interacting by exchanging messages, and by combining services more complex services are created. *Choreography* is concerned with the composition of such services seen from a global viewpoint focusing on the common and complementary observable behavior. Choreography is particularly relevant in a setting where there is not a single coordinator. *Orchestration* is concerned with the composition of such services seen from the viewpoint of single service. Independent of the viewpoint (choreography or orchestration) there is a need to make sure that the services work together to ensure the correct execution of business processes.

This paper explores the use of *process mining* [1] in the context of IBM's WebSphere product. WebSphere provides a state-of-the-art infrastructure for realizing a SOA and supports elaborate logging facilities [2]. The Common Event Infrastructure (CEI) offers a systematic way of recording events. Using this information, we can apply the many process mining techniques provided by the process mining tool *ProM* [4].

CEI provides facilities for the generation, propagation, persistence, and consumption of events. Events are represented using the Common Base Event (CBE) model, a standard XML-based format defining the structure of events. For many applications, the information stored in CEI may be too large. Hence, CEI is often only used as a transport layer and events are removed, filtered, or aggregated by IBM tools such as the WebSphere Business Monitor. (But also others such as the Web Services Navigator [3].)

The WebSphere Business Monitor [2] measures the performance of a process based on key performance indicators (KPIs) and the business metrics. Performance related results are displayed in dashboards and used

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

as reference for redesign. The monitoring and analysis tools are not able to discover causal relations between tasks or employees involved in the process, and, thus, they can not extract a process model from the event log. Moreover, an audit of the process to see if it conforms to the organizational procedures and regulations is hardly objective or efficient without having a good understanding of the real process.

This paper demonstrates that process mining is possible and valuable in a SOA context, using WebSphere as an example. However, our findings are quite general and can be applied to other platforms (e.g., using Oracle BPEL). The remainder is organized as follows. First, we discuss the requirements for process mining. Then, we provide insight into the analysis results that can be provided by process mining. Finally, we discuss in what way process mining tools extend capabilities of existing monitoring tools.

2 Getting Data: Correlation is Key!

More and more processes leave their "trail" in the form of event logs. Process mining techniques can use these logs in various ways, e.g., to discover the way that people/services really work, to find out if and where this way deviates from the planned process, to support people in performing their duties, and to improve the performance of processes. In order to do this, process mining techniques expect the logs to contain certain information. Therefore, we first elaborate on this information.

For every *process instance* (often referred to as *case*), a sequence of events is recorded. Examples of process instances are customer orders in a order handling process, patients in a treatment process, and applicants in a selection process. It is crucial to note that events need to be linked to process instances in order to enable process mining. The sequence of events describing a single process instance is called a *trace*. An *event log* is a set of process instances. An event may have various properties such as the associated time, the associated activity, the associated data, the person, organization, or software component responsible for emitting the event, and the associated transaction type (e.g., start, complete, and abort). Process mining assumes that each event is associated to an activity. All other event properties are optional, but can be exploited when present.

One of the major challenges in processing the collected data is to link events to process instances. This corresponds to the notion of *correlation*. For example, when tapping of a message exchanged between two services it is crucial to link this message to a particular process instance. In some cases this may be trivial, e.g., when using a workflow engine with a clear process instance concept or when there is a natural global identifier such as the patient id in hospital processes. In other cases, this may be very difficult. For example, in the context of an ERP system like SAP R/3 it is surprisingly difficult to correlate events. For example, events related to a customer order may refer to order line items rather than the customer order or the supplier and customer may use different keys.

To make things more concrete, we now focus on event logging in the context of WebSphere. IBM uses the so-called *Common Event Infrastructure* (CEI) to record, distribute, and manage events. IBM encourages clients to use the following four subsystems: (1) *WebSphere Business Modeler* to design business processes and to identify the things to be measured and analyzed at run-time, (2) *WebSphere Integration Developer* to translate business process models into actual executable code, (3) *WebSphere Process Server* to enact the configured processes, and (4) *WebSphere Business Monitor* to observe the processes, to measure Key Performance Indicators (KPIs), generate reports, show management dashboards, etc. Although these subsystems are connected, they can also be used independently. For example, the WebSphere Business Monitor can also be used in conjunction to other products such as FileNet P8 BPM, etc.

Correlation is important in both the Process Server and the Business Monitor. To execute processes, incoming events (e.g., messages) need to be routed to the corresponding BPEL process instances. For monitoring, it is also important to correlate events. Take, for example, a KPI that measures the average throughput time of a case. Clearly, to be able to measure such a KPI, it is necessary to correlate the events. The WebSphere Business Monitor uses the concept of "monitoring context" to define a container where all events related to the same

instance are brought together. It is also interesting to note that both CEI and the WebSphere Business Monitor use concepts such as times associated to events, etc.

We can summarize the above as follows. For process mining, events need to be correlated to process instances. Correlation problems may inhibit the application of process mining. However, as illustrated using the WebSphere suite, correlation is a foundational concept in the development of web services.

3 Analysis using Process Mining

The goal of process mining is to discover, monitor, and improve real processes by extracting knowledge from event logs. Clearly, process mining is particularly relevant in a setting where the actors involved are autonomous and can deviate or have emerging behavior. The more ways in which services, people, and organizations can deviate, the more interesting it is to observe and analyze processes as they are executed.

Three basic types of process mining can be identified:

- **Discovery**: There is no a-priori model, i.e., based on an event log some model is constructed. For example, using the well-known α-algorithm 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 one million Euro require two checks. 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 dynamically annotated with performance data (e.g., bottlenecks are shown by coloring parts of the process model).

In the context of web services, all three types of process mining can be applied. Using the CEI infrastructure and data used by components such as the WebSphere Business Monitor, it is possible to do a wide variety of analyses including the ones shown in Figure 1.

The top-right corner in Figure 1 shows a discovered process models using the EPC notation (i.e., the process modeling language used by systems such as ARIS and SAP). The lower half shows performance related results. The bottom-left corner is a nice illustration of "extension", i.e., the model discovered through process discovery is enriched with information about bottlenecks.

In the context of Websphere it is especially interesting to check conformance. First, using Websphere Business Modeler, a business analyst designs a service, which includes a process model and KPIs. Second, using Websphere Integration Developer, this design is implemented by an IT specialist. Third, using the Process Server, this implementation is executed. Using conformance checking, the business analyst could first check whether the implemented service actually fits (conforms to) the designed service. If not, then the KPI validation (does a KPI actually measure what the analyst thinks it is measuring?) is at stake.

4 Conclusion

The potential of applying process mining in the context of web services is huge. Data is omnipresent and issues like correlation can be addressed by using existing solutions. Moreover, the autonomous nature of services makes it interesting to observe processes as they actually take place.

Processes mining goes beyond classical monitoring components like WebSphere Business Monitor, because there is no need to model the processes beforehand. This offers several advantages. As an example, the deployment time can be reduced dramatically. Existing monitoring solutions typically require extensive modeling

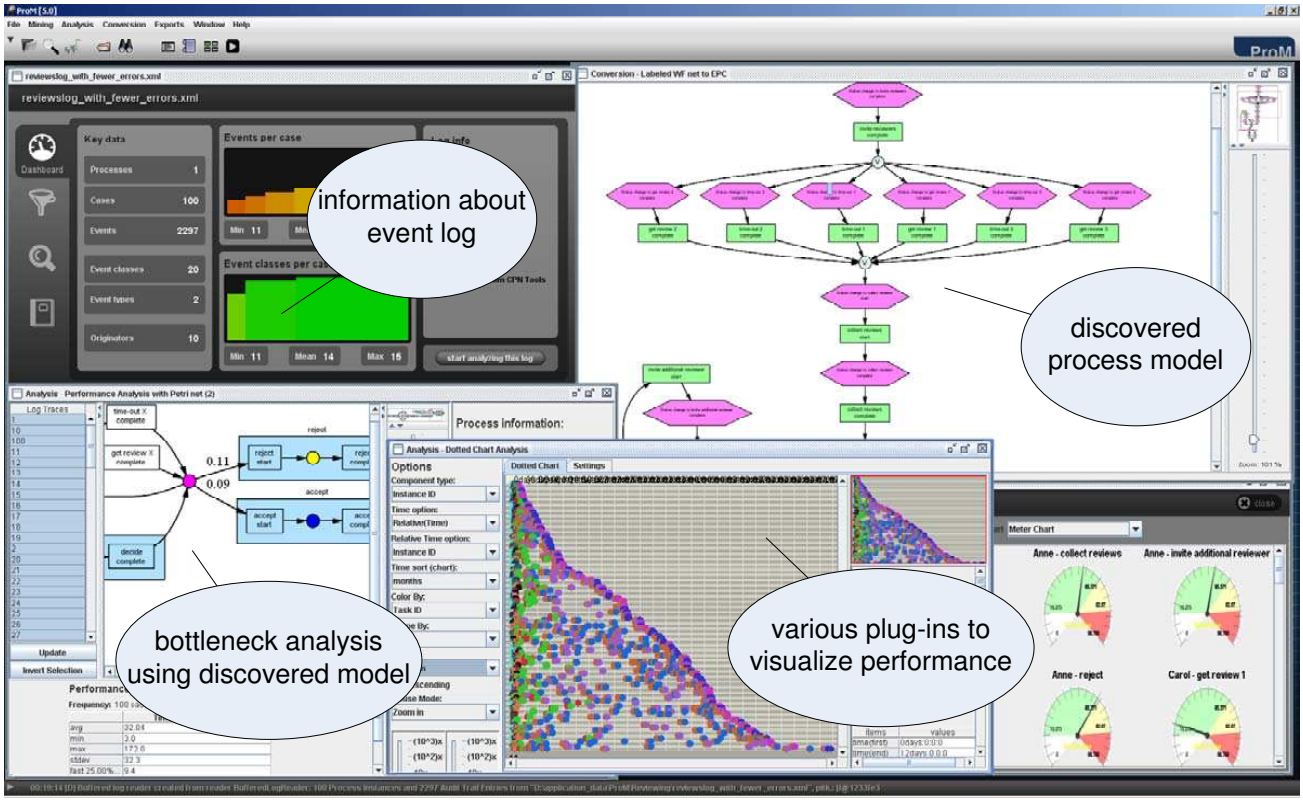

Figure 1: Screenshot of ProM showing some example results.

and configuration and cannot be changed easily. Since process mining techniques can "learn" processes, the modeling phase can be shortened and filtering techniques can be used to change the view on the process at any point in time. Process mining techniques are also able to detect (conformance) process changes and to adapt (discovery) the monitor model.

Process mining tools such as ProM have shown to be able to work with huge amounts of data and, therefore, process mining can be applied to real-life web services.

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] IBM Corporation. Business Activity Monitoring with WebSphere Business Monitor V6.1 Redbook. www.redbooks.ibm.com, 2008.

[3] 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.

[4] W. M. P. van der Aalst, at al. ProM 4.0: Comprehensive support for real process analysis. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer-Verlag, Berlin, 2007.