A Dashboard-based Approach for Monitoring Object-Aware Processes

Marius Breitmayer, Lisa Arnold and Manfred Reichert

Institute of Databases and Information Systems, Ulm University, Germany
{marius.breitmayer,lisa.arnold,manfred.reichert}@uni-ulm.de

Abstract. Data (e.g., event logs) gathered during the execution of business processes enable valuable insights into actual process performance. To leverage this knowledge, these data should be analyzed and interpreted in the context of the respective processes. Corresponding analyses allow for a comprehensive process monitoring as well as the discovery of weaknesses and potential process improvements. This also applies to object-aware processes, where data drives process execution and, thus, is treated as first-class citizen. This paper introduces a dashboard with advanced monitoring functions for object-aware processes.

Keywords: object-aware processes, process monitoring, dashboard

1 Introduction

Process monitoring leverages the data created during the execution of business processes in order to gain insights into actual process performance and to ensure process conformance with regulations, policies, and business rules [12]. This information, in turn, can then be used to monitor, analyze and improve processes [8]. Despite existing approaches for monitoring activity-centric processes (e.g., [5]), adequate monitoring support for data-centric paradigms to business process management (BPM) [16], e.g., artifact-centric processes [7], object-aware processes [10], and case handling [2], is still lacking. As opposed to activity-centric processes, in object-aware process management, large process structures involving multiple concurrently executed and interacting object lifecycle processes, emerge during runtime [15,4]. The monitoring of such dynamically evolving process structures constitutes a particular challenge not sufficiently addressed by contemporary approaches [14]. As data is treated as first-class citizen in object-aware process management, however, the monitoring of data-centric processes yields great potential for more advanced analyses and better comprehensibility of processes in general due to the tight integration of process and data.

To set a background, a short introduction of object-aware process management is provided. PHILharmonicFlows, our approach to data-centric BPM [10], introduces the concepts of *objects*, *object behavior,* and *object interaction*. Each business object of a real-world business process is represented as one such object. The latter comprises data, represented in terms of *attributes,* and a state-based process model describing object behavior in terms of an *object lifecycle* model. In a practical application of the PHILharmonicFlows system, which resulted in a data- and process-aware e-learning system, examples of business objects include *Exercise*, *Submission,* and *Lecture* [3]. At runtime, when necessary data (i.e., object attributes), such as *Points* or *Feedback,* become available, this enables the transition between the states of a lifecycle. Finally, interactions between object lifecycles are managed by coordination processes [13].

Section 2 discusses related work. Section 3 sketches three approaches for calculating process progress, followed by a description of the monitoring dashboard in Section 4. Finally, Section 5 provides a summary and outlook.

2 Related Work

In the BPM lifecycle, process monitoring is a key component for analyzing and improving processes [8,17]. A wide range of approaches and tools exist for activity-centric processes, especially in the field of process mining [1]. To identify problems of a process, conformance checking verifies whether a process is executed as intended by replaying or aligning an event log with the corresponding process model [6]. However, there only exist few approaches for monitoring data-centric processes, which focus on artifact-based processes [11]. Process monitoring and conformance checking for activity-centric processes mainly focus on the identification of bottlenecks, deviations between event log and process model, and the prediction of the next activities [9]. By contrast, our work deals with the monitoring of object-aware processes and the large process structures emerging in this context during runtime.

3 Towards Progress Calculation

To understand the calculation of progress metrics displayed in the dashboard (c.f. Fig. 1), this section describes the three approaches implemented for calculating the progress of object lifecycle instances. Though none of the described calculations covers all aspects, their combination (e.g., individually weighted average) constitutes a good approximation and is therefore supported by the dashboard.

Approach 1 (Lifecycles). This approach divides the number of completed lifecycle steps by the number of total lifecycle steps for each object instance. Note that this calculation is non-trivial as different steps may be executed depending on data (and routing decisions). Therefore, the number of completed steps may vary depending on the path taken through the lifecycle. To reconstruct the latter, for each lifecycle step it is checked whether a value is assigned in the event log. This allows determining the actual number of steps executed for each object instance, which then may be divided by the number of remaining steps.

Approach 2 (Relations). Using the hierarchies specified in the data model, the progress of an object instance can be calculated as the average progress of its lower-level instances. This approximates how constraints between states of object instances influence the progress of the overall object-aware process.

Approach 3 (Temporal Distance). This approach calculates the temporal distance between object instantiation and a certain point in time in the event log. To calculate the progress, this distance may then be divided by the average duration of all completed object instances of the same type.

4 Monitoring Dashboard

The monitoring of an object-aware process needs to consider the information contained in its *data model* (e.g., objects, attributes, relations, hierarchies), its object *lifecycles* (e.g., steps and states), and its *coordination process* [10] in conjunction with the data recorded during process execution (e.g., event logs). We created a monitoring dashboard that allows us to display various aspects of an object-aware process (see Fig. 1). As input, this dashboard takes an object-aware process (i.e., the data model, lifecycles and coordination process) and the event log created during process enactment. Note that event logs of object-aware processes differ from the ones of activity-centric processes. While the latter consists of case-related activity events enriched with additional information [1], a data-centric event log contains information about object instances (e.g., ids, object types, attributes and their values), lifecycle processes (e.g., steps, states, state changes), and object interactions. Using this information, the monitoring dashboard can provide an overview of the overall object-aware process (see Fig. 1), as well as drill-down and roll-up functions for inspecting selected aspects. The dashboard was evaluated for two processes with corresponding event logs.

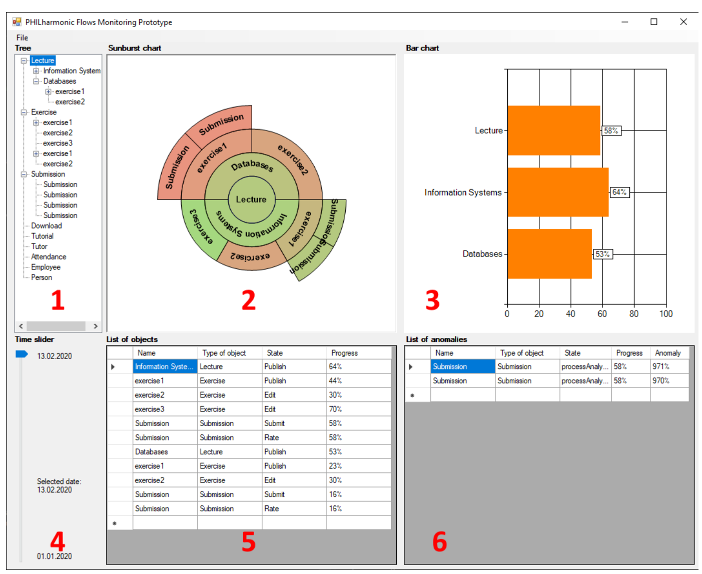

Fig. 1. PHILharmonicFlows Monitoring Dashboard

1. The *tree navigation element* lists the various object types (e.g., *Lecture*, *Exercise*, *Submission*) derived from the data model, together with the corresponding instances (e.g., "Databases" and "Information Systems" as instances of object *Lecture*) recorded in the event log. Additionally, the tree navigation displays lower-level instances of any object instance that can be derived from the hierarchical structuring of the objects in the data model (e.g., "Exercise1" as lower-level instance of the "Database" lecture).
2. The *sunburst chart* provides an overview of selected objects, together with their instances and corresponding lower-level instances. Each layer contains the object instances belonging to the same hierarchical level (e.g., all exercises of a lecture series). However, when the number of object instances grows, the ratio of each individual instance shrinks. To tackle this issue, we implemented "drill-down" and "roll-up" functions. This allows inspecting individual process aspects (e.g., object instances) instead of the entire process. Additionally, color coding of elements enables fast bottleneck identification.
3. The *bar chart* provides a quick and easy method to either compare the progress of similar object instances (see Fig. 1) or the average progress of instances of the same type. We added "drill-down" and "roll-up" functions.
4. The *time slider* allows displaying the state of the object-aware process at any point in time based on recorded event logs. If a point in time other than the most recent one is selected, the status of displayed instances is reconstructed through partial event log replay. As a result, all dashboard elements display the state of the process at the selected point in time. This allows replaying the process as well as detecting former issues that have been resolved.
5. The *list of object instances* displays additional information of the object instances being of interest. The table may be sorted according to any criterion.
6. The *anomalies table* lists object instances that are likely to be outliers. To identify these outliers, the dashboard allows for the comparison with a relative or absolute threshold. For relative comparison, this is accomplished using the following formula:
   $$(1 + \frac{threshold(\%)}{100}) < \frac{InstanceValue}{ObjectAverage} \text{ or } \frac{InstanceValue}{ObjectAverage} < (1 - \frac{threshold(\%)}{100})$$
   For absolute comparison, the difference between an object instance and the object median is calculated and compared with the absolute threshold (e.g., 3 days). To enable a more sophisticated outlier detection, the dashboard is able to combine both methods using AND/OR operations.

5 Conclusion and Outlook

This paper presents a monitoring dashboard for object-aware processes which enables advanced monitoring functions. It combines knowledge from the object-aware process model with information about the execution to visualize the states of all object instances at any moment during the process execution. This allows detecting bottlenecks and outliers. Additionally, the dashboard was tested with two real-world processes and corresponding event logs. In future work, we will improve outlier detection based on more sophisticated algorithms and incorporate different user perspectives (i.e., personalized monitoring views).

Acknowledgments This work is part of the ZAFH Intralogistik, funded by the European Regional Development Fund and the Ministry of Science, Research and Arts of Baden-Württemberg, Germany (F.No. 32-7545.24-17/12/1)

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer (2016)
2. van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. DKE **53**(2), 129–162 (2005)
3. Andrews, K., Steinau, S., Reichert, M.: Engineering a highly scalable object-aware process management engine using distributed microservices. In: Int'l Conf on CoopIS'18. pp. 80–97 (2018)
4. Andrews, K., Steinau, S., Reichert, M.: Enabling runtime flexibility in data-centric and data-driven process execution engines. Information Systems p. 101447 (2019)
5. Bülow, S., Backmann, M., Herzberg, N., Hille, T., Meyer, A., Ulm, B., Wong, T.Y., Weske, M.: Monitoring of business processes with complex event processing. In: Business Process Management Workshops. Springer (2014)
6. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking. Springer (2018)
7. Cohn, D., Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull. **32**(3), 3–9 (2009)
8. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, 2nd edn. (2018)
9. Jorbina, K., et al: Nirdizati: A web-based tool for predictive process monitoring. In: BPM Demo Track and BPM Dissertation Award (CEUR Workshop Proceedings, Volume 1920), pp. 1–5 (2017)
10. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. J of Soft Maint & Evo **23**(4), 205–244 (2011)
11. Meroni, G.: Artifact-driven Business Process Monitoring. Ph.D. thesis, Politecnico di Milano Milan Italy (2018)
12. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, Berlin-Heidelberg (2012)
13. Steinau, S., Andrews, K., Reichert, M.: Modeling process interactions with coordination processes. In: CoopIS'18. pp. 21–39. LNCS, Springer (2018)
14. Steinau, S., Andrews, K., Reichert, M.: The relational process structure. In: CAiSE 2018. pp. 53–67. No. 10816 in LNCS, Springer (2018)
15. Steinau, S., Andrews, K., Reichert, M.: Executing lifecycle processes in object-aware process management. In: Data-Driven Process Discovery and Analysis. pp. 25–44. Springer (2019)
16. Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., Reichert, M.: DALEC: A framework for the systematic evaluation of data-centric approaches to process management software. Softw & Sys Modeling **18**(4), 2679–2716 (2019)
17. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2019)