Towards Continuous Behavior Mining

Sabine Wolny, Alexandra Mazak, Rafael Konlechner, and Manuel Wimmer *

Christian Doppler Laboratory for Model-Integrated Smart Production (CDL-MINT)
TU Wien, Austria
{lastname}@big.tuwien.ac.at

Abstract. With new advances in Cyber-Physical Systems (CPS) and Internet of Things (IoT), more and more discrete software controllers interact with continuous physical systems. Workflow models are a classical approach to define controllers. However, the effect of the associated actions that are activated by executing the workflow may not spontaneously be realized but have to be realized over time. Generally, behavioral model elements such as activities in workflow languages are displayed mostly as black box, meaning that it is not possible to trace variable changes over time in most of the classical modeling approaches. In this paper, we introduce an envisioned architecture to cope with this challenge.

1 Setting the Stage for Continuous Behavior Mining

While design models help in the engineering process by providing appropriate abstractions, data-driven approaches such as process mining may help to uncover some underspecified or unintended parts of these models. In many behavioral modeling languages, e.g., workflow-based languages or state-based languages, interactions are performed with physical systems where the effect of certain actions have to be realized by continuous system updates. Imagine a production system where actions do not occur instantly but require time as they are continuous flows (e.g., robot is changing place). However, in behavioral models, actions are mostly expressed as value assignments for variables. Thus, behavioral model elements such as activities in workflow languages are displayed as a "black box", meaning that the intended source and target state of the system is specified. But it is not possible to trace variable changes over time within activities. Whereas models abstract from this detail, it is important to open the black box of activities to reason about the shapes of continuous variables for different scenarios such as validation and verification, optimization, and evolution.

Today, sensors continuously generate and transmit a massive amount of data of systems published as data streams. Such streams are ordered and potentially unbounded sequences of data points created by a typically non-stationary generating process. This trend raises new challenges for model-driven approaches [2], but at the same time possibilities. For instance, is it feasible to exploit data streams to provide intra-activity specifications for continuous variable? In the following, we introduce a first architecture going in this direction.

* This work has been funded by the Austrian Federal Ministry of Science, Research and Economy (BMWFW) and the National Foundation for Research, Technology and Development.

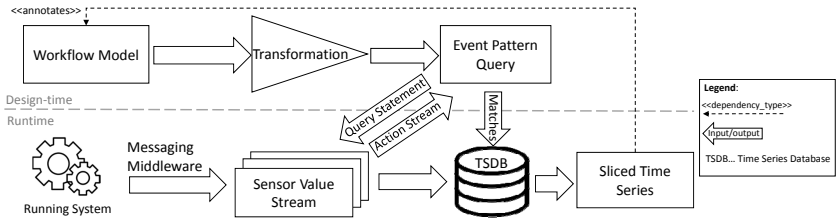

Fig. 1. Architecture for intra-activity behavior mining.

2 A First Architectural Style for Continuous Behaviour Mining

Consider a system where sensors continuously send data to a controller, which sends signals to the actuators based on these data according to a predefined workflow. This workflow is modeled at design-time and defines the expected execution order for the system. At runtime, it is important to trace and monitor whether the system behaves as intended or not. A promising way to achieve this requirement are event pattern queries [1]. Such a query describes a fixed pattern for certain event sequences. In this respect, the workflow model (WM) may be used to derive such event pattern queries. Figure 1 shows the sketch of an initial architecture exploiting event patterns to reason about continuous variable updates.

At runtime, each component within the system communicates via a messaging system middleware. The middleware can be also used to transfer sensor values to produce logs of a system in a time series database (TSDB). A single log contains the following fields: (i) *timestamp*, the actual time; (ii) *sensor*, the name of the sensor; and (iii) *value*, the received sensor value. By recording the sensor data in a TSDB, it is possible to illustrate variable changes of components continuously. Beside the recording of the data, the value streams can be analyzed for detecting the defined activities of the WM. Thereby, the value streams are filtered by an event pattern query which may be derived from the WM. By this query, one can continuously sample the runs of the system, e.g., to detect process duration changes over time (i.e., time between matches becomes longer or shorter). The query responses produce a stream of matched actions, the so-called "action stream". Now, it is possible to map the logs of the received sensor values and the query matches together in the TSDB to produce condensed figures of continuous variable value shapes within activities. With the help of the presented architectural style, complex behavior of the system can be continuously monitored.

Outline for future work. Based on the logs of the sensors and the logs of the query matches, it is possible to analyze specific slices of time series. For instance, hidden activities may be detected where variable changes are observed between two defined activities, i.e., they are not described in the WM.

References

1. D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, third edition, 2005.
2. A. Mazak and M. Wimmer. Towards liquid models: An evolutionary modeling approach. In *18th IEEE Conference on Business Informatics (CBI)*, pages 104–112, 2016.