

# Generating Stream Processing Applications for IoT-based Process Event Abstraction: A Framework and Architecture

Ronny Seiger <sup>1</sup>, Aaron F. Kurz <sup>1</sup>, and Marco Franceschetti <sup>1</sup>

## Abstract:

Events play an important role in information systems engineering to facilitate the analysis of relevant happenings in a system or business process via process mining. The Internet of Things (IoT) provides new ways of collecting execution-related events in real world—*physical*—systems using sensors and actuators. However, these new sources emit data at a too fine-grained level, which prevents process mining from deriving meaningful insights. We present a generic event abstraction framework to lift low-level data to higher level events. Starting with annotated IoT data, we generate stream processing applications that encode change patterns derived from the low-level data. These applications are then used for detecting events and activities at runtime within our proposed software architecture. We evaluate the approach for process executions in smart manufacturing and smart healthcare.

**Keywords:** Internet of Things, Process Mining, Activity detection, Event abstraction

## Extended Abstract

Event abstraction is needed in many cases where low-level—*fine-grained*—trace data is generated by software, but analysis has to be conducted at higher abstraction levels to gain meaningful insights into processes (e.g., when analyzing user interactions or execution traces). This extended abstract refers to our work published in the International Journal on Future Generation Computer Systems (FGCS) [SKF26]. The contributions are: 1) a framework for generating stream processing applications to enable event abstraction, and 2) a software architecture to enable the runtime detection of process events from IoT data.

The **Event Abstraction Framework** (cf. Fig. 1, top left) starts with a domain expert identifying and annotating an *activity prototype* in the IoT data collected from sensors and actuators, sent via a message broker (0). Using an interactive dashboard (1), the expert annotates one activity with process-level start and end events that represent one execution (cf. Fig. 1, bottom left). These annotations and the corresponding, logged IoT data are retrieved by the *Stream Processing App Generator and Deployer* (2). This component analyzes the IoT data for changes over time, which are used to derive higher level events and trace the activity execution. The changes are encoded as event sequence patterns in a generated stream processing app. The first pattern indicates the start; the sequence of all intermediate change patterns until the last indicates the activity end. We support different pattern matching strategies (e.g., only first and last, the first x percent, any x percent).

<sup>1</sup> University of St.Gallen, Institute of Computer Science, St.Gallen, Switzerland,  
ronny.seiger@unisg.ch,  <https://orcid.org/0000-0003-1675-2592>;  
aaron.kurz@unisg.ch,  <https://orcid.org/0000-0002-2547-6780>;  
marco.franceschetti@unisg.ch,  <https://orcid.org/0000-0001-7030-282X>

The **Runtime Architecture** (cf. Fig. 1, top right) has a Complex Event Processing (CEP) platform at its core (3), which deploys and executes the stream processing apps. Upon activation of an app, configured connections to the IoT data sources—sensors and actuators—are established via a message broker (4) and the pattern matching rules in the stream processing app become active. Once a change pattern is detected, a higher level process event is emitted, which can be consumed by subscribed process mining applications (5).

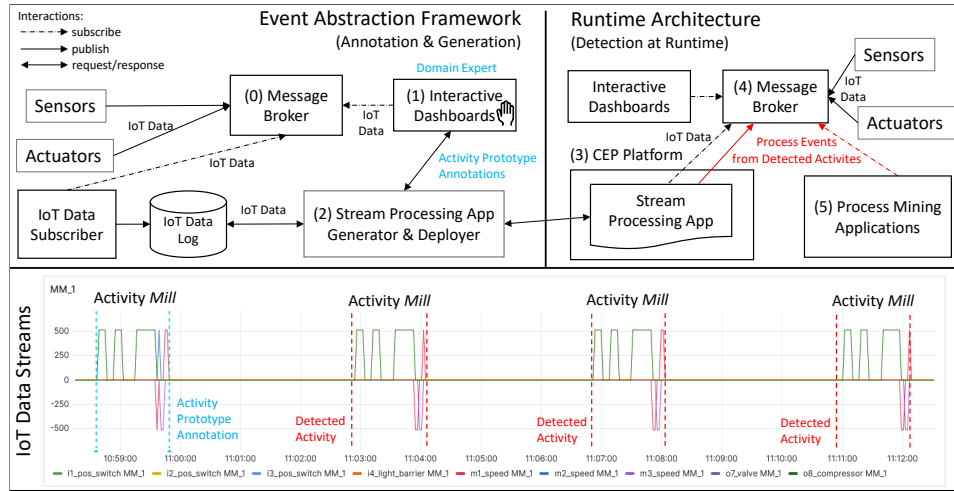


Fig. 1: Generation of stream processing apps from annotated IoT data (top left); detection of activity executions from IoT data at runtime (top right); IoT data streams from sensors and actuators (bottom)

**Evaluation and Results:** We evaluated the approach in two case studies, executing processes and detecting activities in 1) smart manufacturing with sensors, motors, valves, and compressors as data sources; and 2) healthcare with data sources comprising NFC, motion, proximity, weight, and environmental sensors. We compared the quality of activity detections using different pattern matching strategies with a recorded ground truth. The dataset and implementations are available on Zenodo. The results of our evaluations confirm the validity of the framework and architecture to enable activity detection based on IoT data at runtime. We identified fluctuations and variability in the IoT data during activity executions as the main factors influencing the detection quality. Using one annotated activity as a prototype leads to light-weight stream processing apps that perform activity detection efficiently, but these apps show limitations when faced with variations in IoT data. Additional CEP patterns have to be encoded in the apps to address these fluctuations.

## References

- [SKF26] Seiger, R.; Kurz, A. F.; Franceschetti, M.: Online detection of process activity executions from IoT sensors using generated event processing services. *Future Generation Computer Systems* 174, p. 107987, 2026.