Method to Identify Process Activities by Visualizing Sensor Events

Flemming Weyers, Ronny Seiger(✉), and Barbara Weber

Institute of Computer Science, University of St. Gallen, St. Gallen, Switzerland
ronny.seiger@unisg.ch

Abstract. With the onset of the Internet of Things (IoT) everyday objects suddenly become data sources equipped with sensors measuring the object's properties and surroundings. However, the lack of process-awareness in IoT environments (e.g., smart factories) prevents the adoption of more sophisticated process analysis and optimization. One hurdle is the differing abstraction level of low-level sensor data and process-level activities. We propose a method to identify activities step-by-step from raw IoT data using visualizations. By relying on minimal process knowledge, we discover process activities from sensor events. These activities are represented by specific sequences of sensor events–*Activity Signatures*–that serve as a basis for finding similar activities. We demonstrate the method's validity with a proof of concept in a smart factory.

Keywords: Business Process Management · Cyber-physical systems · Internet of Things · Activity detection · Sensors

1 Introduction

With the onset of the Internet of Things (IoT), more and more domains are pervaded with sensors and actuators controlled by software [22]. Physical objects suddenly produce data about their state, surroundings and the processes they are involved in [10]. However, process-awareness in the sense of Business Process Management (BPM) is still very-low in IoT as there usually is no workflow management system (WfMS) available to orchestrate or monitor processes [15]. The *BPM-IoT Manifesto* discusses various challenges and benefits of bringing both domains together [6]. With this work, we investigate how process activity executions in IoT can be linked to sensor data from the respective IoT devices and vice versa. The goal is to discover activities from raw IoT sensor data, thereby addressing the challenge of "Bridging the Gap Between Event-Based and Process-Based Systems" [6]. Existing approaches like supervised machine learning (ML) rely on inputs from the WfMS (e.g., activity labels). Unsupervised ML struggles with feature selection, especially with thousands of sensors as inputs. We propose a novel visualization-based method to enable the identification of activities from raw sensor data that serves as basis for future automation of sensor data analysis. The following research questions guide our investigations:

C. Cabanillas et al. (Eds.): BPM 2022, LNBIP 460, pp. 76–88, 2023.
https://doi.org/10.1007/978-3-031-25383-6_7

RQ1 What are reasonable assumptions and steps to analyze sensor data for process activities in IoT?
RQ2 How can sensor data be associated with activity executions to potentially automate the detection of activities from sensor data?

The paper is structured as follows: Sect. 2 introduces relevant background. Section 3 discusses related work. Section 4 introduces the new method for activity detection from sensor data. Section 5 demonstrates and discusses the method's validity. Section 6 concludes this paper and gives an outlook on future work.

2 Basic Concepts and Context

This work is closely related to research at the intersection of BPM and IoT/CPS (Cyber-physical Systems). We propose to adapt the "Ingredients of a business process" [3] combined with the "UML representation of the IoT Domain Model" [1] as basic conceptual model for bringing both fields together (cf. Fig. 1).

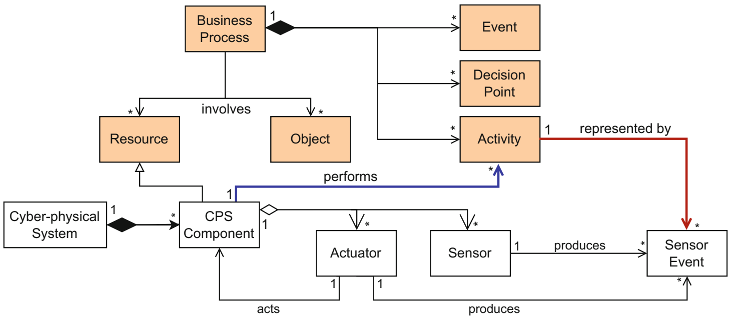

Fig. 1. Meta-model with basic concepts from BPM and CPS used in this work.

2.1 Basic Concepts

Business Process Management: BPM is "[...] a well-established discipline that deals with the identification, discovery, analysis, (re)design, implementation, execution, monitoring, and evolution of organizational procedures" [3]. *Business Processes* are chains of events, activities and decisions to achieve a desired outcome [3]. *Activities* can be both fine-grained or coarse-grained units of work [3]. *Resource* in the context of BPM is "[...] a generic term to refer to anyone or anything involved in the performance of a process activity" [3].

Cyber-Physical Systems and Internet of Things: CPS integrate computation and physical processes where both the digital and physical systems affect each other [11]. In IoT, everyday objects are interconnected and work together to accomplish an objective [10]. While IoT focuses on interoperability and communication among devices, CPS put emphasis on control and automation relying on IoT for connectivity. As this fits well with our research, we use the term *Cyber-physical System (CPS)* throughout this paper. In *Smart Factories*, hardware/-software components that are composed of sensors, actuators and controllers work together in manufacturing cells to achieve a production outcome [15]. We call these self-contained cells *CPS Components* of a smart factory representing CPS. We treat CPS components as resources that execute activities of a production-related business process [12]. Our investigations focus on the relation between sensor data and activity executions by the CPS components (cf. Fig. 1).

Sensors and Actuators: Sensors monitor a physical entity and provide information about its physical and virtual properties [10]. They can be attached to, or embedded in the entity's structure or be placed in its environment [10]. *Actuators* modify the entity's physical state or influence other entities' functionalities (e.g., via motors or valves in the CPS components) [10]. They also produce data (e.g., regarding their states) that we treat as sensor data. Thus, we refer to both, data from sensors and actuators, as *Sensor Data/Sensor Events* in this paper.

2.2 Context: Fischertechnik Factory Model

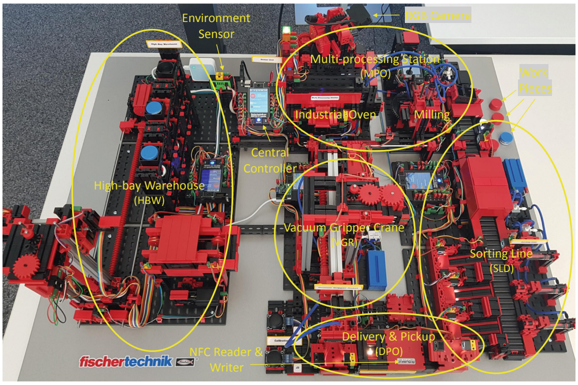

Fig. 2. Smart factory simulation model as a CPS representative [15].

Figure 2 shows the Fischertechnik smart factory model that represents CPS in our work [15]. Each highlighted station is one CPS component (e.g., HBW, VGR, SLD). The factory features realistic discrete manufacturing processes and CPS components, each equipped with a multitude of sensors (e.g., light-barriers, switches) and actuators (e.g., motors, valves) [12]. We rely on the software stack presented in [14] for controlling the smart factory.

```
1 {"UUID": "91c4fd59-27d7-477b-ae18-2ef8b6f04cb5",
2 "timestamp": "2022-02-23 10:34:23.72",
3 "i_1": 1, "...", "i_n": 0,
4 "o_1": 0, "...", "o_n": 0,
5 "m_1_speed": 0, "...", "m_n_speed": 5.12,
6 "target_pos_x": 0,"target_pos_y": 0,"target_pos_z": 0}
```

Fig. 3. Exemplary payload in JSON for one message in the *VGR* topic.

Sensor and Actuator Event Streams: An MQTT (Message Queuing Telemetry Transport [19]) broker streams sensor events from the smart factory during its operation. We use one *Topic* per CPS component, which is a message channel for clients to subscribe to and receive messages. Figure 3 shows the payload of an MQTT message including multiple attributes that represent individual sensor and actuator events: unique identifier of the message (Line 1); timestamp when the message is generated (Line 2); `i_1 ... i_n` for input values from sensors (Line 3); `o_1 ... o_n` for states of output devices, e.g., valves or compressors (Line 4); `m_1 ... m_n` for motor speeds (Line 5); additional component-related parameters, e.g., target positions (Line 6). We subscribe to all topics (= CPS components from Fig. 2) and record the messages for offline analysis.

2.3 Process Awareness in Sensor Data

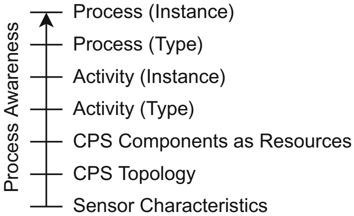

Fig. 4. Bottom-up process awareness associated with sensor data.

Process execution knowledge (called *Process Awareness*) that can be associated with sensor data exists on a spectrum (cf. Fig. 4) [2]. It ranges from only knowing the characteristics of individual sensors (as discussed, e.g., in [8]) to being able to associate a concrete activity and process instance with a given sensor event (as discussed, e.g., in [13]). Our basic assumption is that a WfMS does not always exist to coordinate and monitor process executions in CPS [15]. Our goal is to gradually increase process awareness in sensor data following a bottom-up approach.

3 Related Work

CPS introduce new approaches, e.g., for condition monitoring or predictive maintenance based on recorded sensor data [5]. Massive amounts of data, different data formats, sampling rates, and data quality are among the challenges that come with using sensor data [9]. Existing works use different types of data for the discovery of events and activities at different levels [2]. Koschmider et al. provide a framework to discover processes from sensor data. Accordingly, we focus on “Activity Discovery”, and “Event Abstraction”, where we relate events to the start or completion of process activities [8].

Going from sensor data to data suitable for process mining poses challenges regarding event extraction, abstraction, and event correlation [2,6]. Identifying relevant data for process mining and extracting it from different sources is part of *Event Extraction* [2]. This data often resides in traditional databases and information systems in the form of *Event Logs*. Existing approaches identify this data using, e.g., database schemas, process documents, domain models, event models, and/or domain knowledge [2,7]. In our work, we use sensors and actuators in CPS as data sources. Event abstraction in the context of BPM focuses on the *abstraction gap* between the granularity at which the data is recorded and at which it is analyzed [2,6,25]. When considering sensor data, the challenge of mapping fine-grained sensor data to more abstract process activities becomes more pronounced [6,24]. In literature [24], various approaches exist to bridge this abstraction gap using, e.g., Complex Event Processing (CEP) [15,18] or machine learning (ML) [4,8,20].

Most works assume rather high levels of process knowledge when discovering activities and processes from sensor data (cf. Fig. 4), i.e., existing activity labels only have to be connected to the raw events [2,13]. With almost no process knowledge and limited CPS topology knowledge (lower end of the spectrum, cf. Fig. 4) most approaches are not applicable. Supervised ML needs activity labels to learn. Unsupervised methods like clustering could find activities, yet the amount of features (thousands of sensor) make it hard to gain valuable insights. We exploit existing CPS topology knowledge and assume CPS components to act as resources to then use data visualizations for building a *Knowledge Base* (KB) of CPS-based activities which can be labeled by domain experts and be used for further identification and labeling of activities in unknown datasets.

4 Method to Identify Activities from Sensor Events

The method to identify activities from sensor data is comprised of multiple steps to visualize data, filter by CPS component, and incrementally refine activities (cf. Fig. 5). We base the method on the *Visual Information Seeking Mantra*: "Overview first, zoom and filter, then details-on-demand" [17]. In the following, we explain each step from a conceptual point and illustrate it with an example.

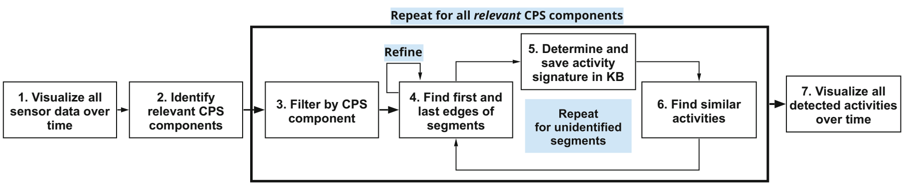

Fig. 5. Method to identify activities by visualizing sensor events.

4.1 Visualize All Sensor Data Over Time

We plot all sensor data from a given dataset to provide an overview for the analyst (*Overview First* [17]). The y-axis shows the concrete values of all sensors and the x-axis shows the associated timestamps.

Example: Figure 6 shows all sensor data from our smart factory over a recorded timeframe. Each graph represents one sensor from a CPS component, e.g., *VGR_i1* refers to sensor *i1* from *VGR* (cf. Sect. 2.2). Even with this small CPS setting, there is a plethora of graphs for all sensors that populate data (cf. Fig. 3).

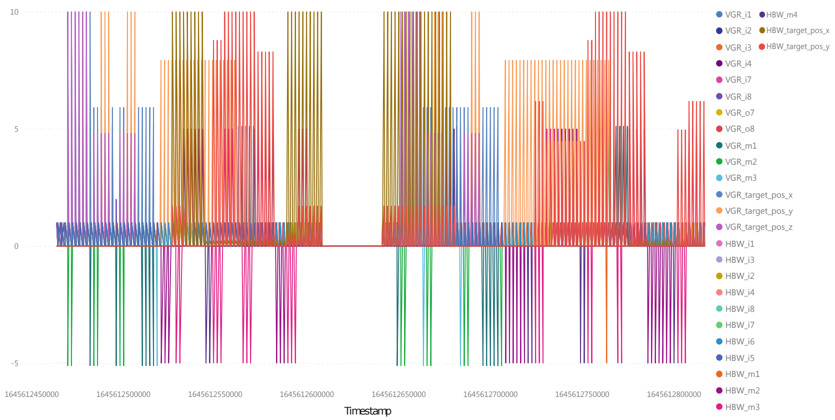

Fig. 6. Recorded sensor data from smart factory model.

4.2 Identify Relevant CPS Components

The *relevant* CPS components are identified. We assume that components relevant for the activity identification show changes within their associated sensor data when executing activities. However, not all CPS components that populate sensor data are also executing activities. This step cannot be fully automated as the relevance should be confirmed by the analyst.

Example: We identify the vacuum gripper crane ("VGR") as relevant CPS component since the values of its associated sensors change over time (cf. Fig. 7). An example for an "irrelevant" component is an environment sensor constantly measuring temperature that may not be associated with a specific activity.

4.3 Filter by CPS Component

We *filter* the sensor data by the first relevant CPS component to only visualize data related to one CPS component. Here we assume that one activity is executed by one CPS component (cf. Fig. 1), i.e., only one CPS component shows changes in its sensor values that are relevant for activity detection. This is step is **repeated** for **all relevant** CPS components.

Example: Figure 7 shows the sensor data for component "VGR" which we identified as a relevant component in step 2. Still, the entire recorded timeframe is shown.

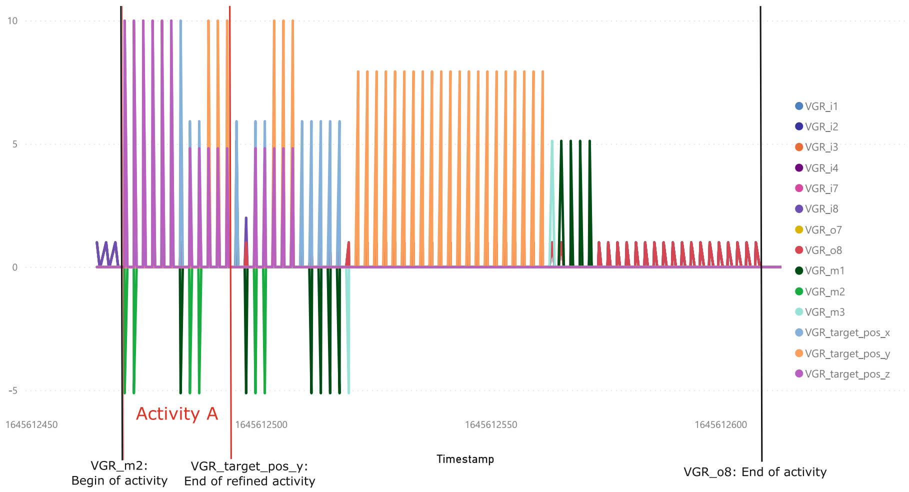

Fig. 7. Identifying and refining activity boundaries for the VGR component based on the first and last edges of segments.

4.4 Identify First and Last Edges of Segments

We search for first and last edges of segments that can be indications of activities. We assume that one CPS component executes only one activity at a time (i.e., there is no batch processing, but only discrete manufacturing steps [21]). By definition an actuator performs work. Thus, an actuator becoming "active" or the occurrence of a "start pattern" (i.e., a combination/sequence of multiple sensors/actuators becoming active) are good indications for an activity's start. The same holds for the end when an actuator stops or an "end pattern" occurs. Times where CPS components are inactive can be short "breaks" that are part of the execution of one activity or they can be an indicator for an activity's end. Additionally, we can factor in *context* to differentiate activities, e.g., the switch to another CPS component is an indicator for an activity's end rather than a break. Since we may not be able to distinguish between the execution of two activities by the same CPS component in sequence, we have to *zoom in* to the identified activity and **repeat** this step to **refine** an activity. An indication for these refinements could be that the set of the CPS component's involved sensors/actuators or the pattern of sensor data changes significantly. This strongly relies on the visualization and analyst's knowledge.

Example: In Fig. 7 we see the first raising and last falling edges of an activity block. With "VGR_m2", an actuator (here: motor) of VGR is switched on which indicates that an activity starts. The last edge shows "VGR_o8" (here: a compressor) switching off followed by a break which indicates the end of the activity. Figure 8 shows this identified activity (black borders). This figure also shows the result of refining the activity (red borders): the VGR executes transport activities and the sensor "VGR_target_pos_y" provides context as to where the crane moves. Since this value changes in the next segment we assume that one transport activity stops and another starts. We now have identified a single activity "Activity A" that does not need to be refined further. As stated before, these refinement steps depend heavily on the knowledge of the analyst.

4.5 Determine and Save Activity Signature in Knowledge Base (KB)

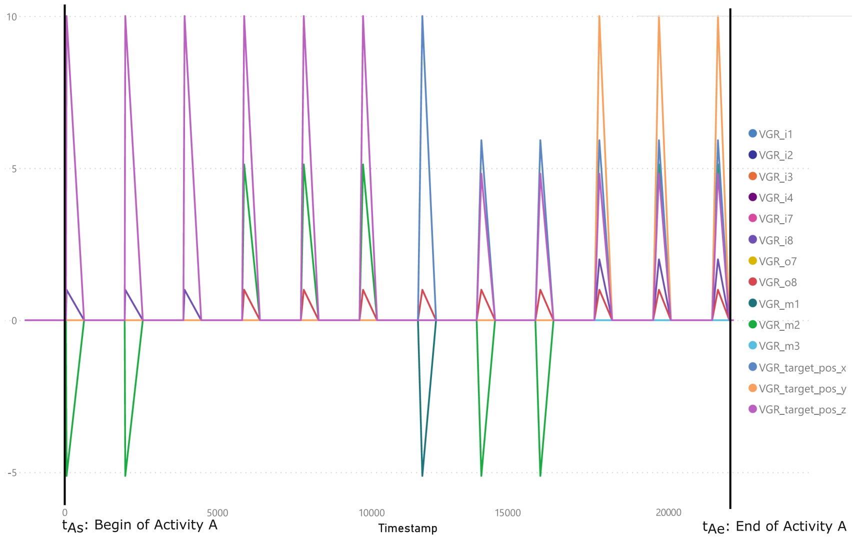

Fig. 8. Activity signature of "Activity A" executed by CPS component "VGR".

The *Activity Signature* of the identified activity is determined (*details-on-demand* [17]). This signature refers to the distinct sequence of all sensor data for the specified CPS component within the identified activity boundaries, i.e., its start time t_{As} and its end time t_{Ae}. The determined activity signature is saved in a KB with a label (e.g., "Activity A") and the respective sensor data as multivariate time series for all timestamps t_n: $t_{As} \leq t_n \leq t_{Ae}$. Thereby, the absolute event timestamps are replaced with relative timestamps starting at $t_{As} = 0$ for the start of the activity until the end of the activity t_{Ae} to search for similar activity signatures within the given dataset in the next step.

Example: Figure 8 shows the *Activity Signature* for “Activity A” executed by the vacuum gripper crane (“VGR”). More specific activity labels have to be provided by the analyst. This signature is stored in a time series database.

4.6 Find Similar Activities

We look for activities based on similarity with the determined signature. This can be achieved by zooming out and visually finding the activity signature for the specific CPS component within the dataset. Visual pattern matching quickly becomes infeasible for determining similarity here since activity signatures can easily grow in complexity, i.e., they may consist of a multitude of sensors and actuators, and patterns within the sensor data over a longer period of time. Moreover, our data shows that the same activities do not necessarily have identical activity signatures, e.g., due to minor variations in sensor and actuator behavior or different process parameters. Thus we need a way to approximate the similarity of the time-series data where the analyst can define a threshold for when an activity is accepted as similar [16]. We are currently investigating the use of *Matrix Profiles* in multi-variate time series as a novel way of calculating these approximations to determine the similarity of signatures [23].

Example: Figure 9 shows multiple identified activities along the entire timeline. The activity signatures of activities “A”, “B”, “C” and “D” were determined in the previous steps. We zoomed out and found similar segments in the dataset that can also be marked as “A”, “B”, “C” or “D”.

4.7 Visualize All Detected Activities over Time

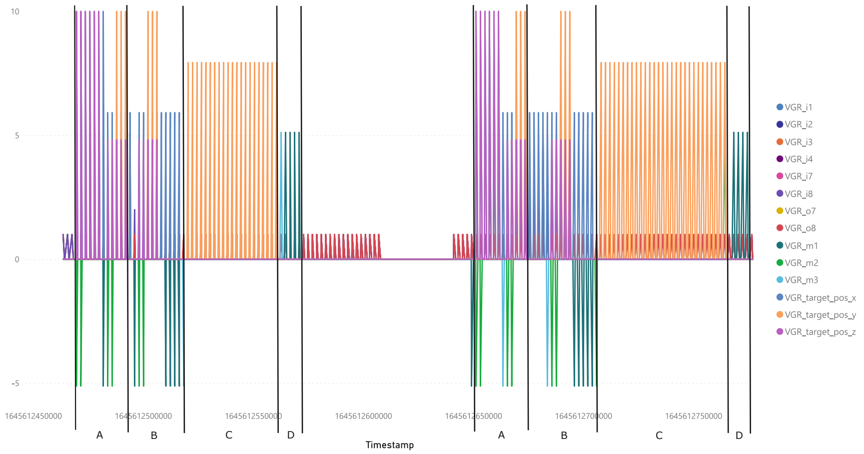

Fig. 9. All detected activities for the CPS component VGR over entire timeline.

After repeating Steps 3–6 in Fig. 5 for relevant CPS components and repeating Steps 4–6 identify new activities within unidentified segments, all detected activities are visualized for all CPS components over the entire timeline to identify process instances. Recurring patterns of activity sequences might be an indication for the execution of different process instances of the same process. However, we cannot fully say if these repeated sequences could also be part of the same instance. Moreover, we limit our approach based on the assumption that only one process instance can be executed at a time, which is reasonable for many discrete manufacturing settings [21].

Example: Figure 9 shows recurring sequences of activities "A", "B", "C" and "D" for the VGR in chronological order. This might indicate that two instances of the same process were executed in sequence. Unlabeled segments are either parts that we were not able to identify or classified as noise or inactivity by the analyst.

5 Proof of Concept and Discussion

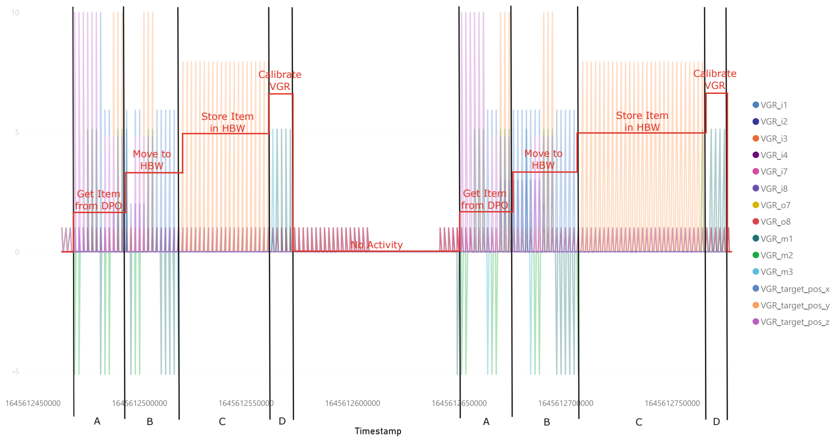

Fig. 10. Overlay of identified activities with event log data.

The software stack used for data recording also features a WfMS for process execution and generating an *Event Log* [14]. When developing the method (cf. Sect. 4), we assumed that the analyst does not have access to this event log. To provide a proof of concept showing the validity of our proposed method in this early development stage, we overlaid the event log data with the activities we identified from the sensor data (cf. Fig. 10). The red graph shows the mapped activities with their actual labels as executed by the WfMS. Apart from minor temporal delays resulting from different timestamp resolutions, we can see an almost exact match between the logged activities and the identified activities. The example also nicely shows that activity signatures for the same type of activity (cf. "Store Item in HBW") may differ as explained in Sect. 4.6.

Considering the research questions, we were able to answer RQ1 by proposing a visualization-based method for analyzing sensor data to identify activity executions while assuming minimal process knowledge and making reasonable assumptions for the domain of discrete manufacturing. Regarding RQ2, we are able to associate sensor data with identified activities based on the novel concept of *Activity Signatures* that can be used to automate the detection of similar activities. However, not all steps of the method can be completely automated. Especially the identification of relevant CPS components and the refinement of identified activities rely on the expertise of the analyst. Our proposed method follows a bottom-up approach to increase process awareness step-by-step (cf. Sect. 2.3). It is suited to identify activities as part of the *control flow* perspective as well as the process resources (i.e., CPS components) that performed them. We can only provide abstract activity labels without relying on further domain knowledge. Although it was not an explicit goal, we are able to make statements about the correlation of detected activities with process instances based on the assumption that no batch processing or parallel process execution is performed.

6 Conclusion and Future Work

In this work, we proposed a method to identify activities from sensor data following a bottom-up approach. Assuming a low degree of process awareness and limited knowledge about CPS, we apply various steps of *overview*, *filter and zoom*, and *details on demand* [17] to visually identify activity executions from sensor data. We also provide first approaches towards automating steps within the method (e.g., based on the new concept of *activity signatures*). A proof of concept evaluation with actual event log data from a WfMS has shown promising results regarding the applicability of our method in smart manufacturing.

In future work, we will relax some of the assumptions made for the initial version of the method (e.g., to also allow for parallel process executions). A larger case study with data from our smart factory will be next. With this, we will investigate the feasibility of the method when working with larger datasets. In this context, we will further develop the concept of *Activity Signatures* to automatically detect activities in unknown datasets based on similarity with known activities [16,23]. We will also investigate if we are able to generate stream processing applications from activity signatures to enable online activity detection [15].

References

1. Bauer, M., et al.: IoT reference model. In: Bassi, A., et al. (eds) Enabling Things to Talk, pp. 113–162. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40403-0_7

2. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. **10**(3), e1346 (2020)
3. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, vol. 1, p. 2. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33143-5
4. Folino, F., Guarascio, M., Pontieri, L.: Mining predictive process models out of low-level multidimensional logs. In: Jarke, M., et al. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 533–547. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_36
5. Hoppenstedt, B., et al.: Techniques and emerging trends for state of the art equipment maintenance systems—a bibliometric analysis. Appl. Sci. **8**(6), 916 (2018)
6. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020)
7. Jans, M., Soffer, P., Jouck, T.: Building a valuable event log for process mining: an experimental exploration of a guided process. Ent. Inf. Syst. **13**(5), 601–630 (2019)
8. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.J.: Process model discovery from sensor event data. In: Leemans, S., Leopold, H. (eds.) ICPM 2020. LNBIP, vol. 406, pp. 69–81. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72693-5_6
9. Kammerer, K., Pryss, R., Hoppenstedt, B., Sommer, K., Reichert, M.: Process-driven and flow-based processing of industrial sensor data. Sensors **20**(18), 5245 (2020)
10. Kramp, T., van Kranenburg, R., Lange, S.: Introduction to the internet of things. In: Bassi, A., et al. (eds) Enabling Things to Talk, pp 1–10. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40403-0_1
11. Lee, E.A.: Cyber physical systems: design challenges. In: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), pp. 363–369 (2008)
12. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using physical factory simulation models for business process management research. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 95–107. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_8
13. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work–Industry 4.0 integration assessment and evolution. In: 17th BPM Conf., pp. 106–117 (2019)
14. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: a systems architecture and use cases. J. Manuf. Syst. **63**, 575–592 (2022)
15. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards IoT-driven process event log generation for conformance checking in smart factories. In: 24th Intern. EDOC Workshop, pp. 20–26. IEEE (2020)
16. Serrà, J., Arcos, J.L.: An empirical evaluation of similarity measures for time series classification. Knowl.-Based Syst. **67**, 305–314 (2014)
17. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: The Craft of Information Visualization, pp. 364–371. Interactive Technologies, Morgan Kaufmann (2003)
18. Soffer, P., et al.: From event streams to process models and back: challenges and opportunities. Inf. Sys. **81**, 181–200 (2019)
19. Standard, O.: MQTT version 5.0 (2019)

20. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
21. Traganos, K., Grefen, P., Vanderfeesten, I., Erasmus, J., Boultadakis, G., Bouklis, P.: The HORSE framework: a reference architecture for cyber-physical systems in hybrid smart manufacturing. J. Manuf. Syst. **61**, 461–494 (2021)
22. Whitmore, A., Agarwal, A., Da Xu, L.: The internet of things—a survey of topics and trends. Inf. Syst. Front. **17**(2), 261–274 (2014). https://doi.org/10.1007/s10796-014-9489-2
23. Yeh, C.C.M., et al.: Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1317–1322 (2016)
24. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granul. Comput. **6**(3), 719–736 (2020). https://doi.org/10.1007/s41066-020-00226-2
25. Zerbato, F., Seiger, R., Di Federico, G., Burattin, A., Weber, B.: Granularity in process mining: can we fix it? In: CEUR WS Proc., vol. 2938, pp. 40–44 (2021)