

Leveraging Digital Trace Data to Investigate and Support Human-Centered Work Processes

Barbara Weber, Amine Abbad-Andaloussi, Marco Franceschetti,
Ronny Seiger, Hagen Völzer, and Francesca Zerbato

Institute of Computer Science, University of St.Gallen, St. Gallen, Switzerland
{barbara.weber|amine.abbad-andaloussi|marco.franceschetti|ronny.seiger|
hagen.voelzer|francesca.zerbato}@unisg.ch

Abstract. The ongoing digitization of processes in all domains of everyday life driven by IT systems shows great potential for process automation, analysis, and optimization. In the last decade process mining has advanced to an important and mature discipline of computer science research and has been widely adopted in industry. More recently, —acknowledging the huge potential of digital trace data to study processes—process science has been introduced as an interdisciplinary field studying how processes unfold over time. This paper discusses the potential that arises when using digital trace data not only in the context of highly automated processes but also to investigate human-centered (work) processes and elaborates on associated challenges. Examples range from the semi-automated storage and production processes in a smart factory to healthcare processes to process analysts performing process mining tasks and software engineers reading software artifacts like source code and process models.

Keywords: Digital Trace Data, Process Science, Human-centered Work Processes

1 INTRODUCTION

Digital technology leaves an ever-increasing volume of digital traces. One major source of event data are information systems but also IoT devices that are equipped with sensors as well as a person’s interactions online produce large amounts of data. Digital traces typically come in the form of events where each event has a timestamp and refers to an occurrence of something that has happened in the past [41]. For example, an event could refer to a well-defined process step, i.e., an activity occurring at a specific moment in time, or to an unexpected happening, such as the breaking down of a machine. These events can also be related into an event sequence, which captures events happening over time [33]. In the context of process mining, events related to a specific process are recorded in an event log. Within

this event log, events referring to the same process instance are denoted as a trace [3]. An event sequence could also refer to a user interaction log including traces of interactions of a user with an IT system [9].

Digital trace data is at the heart of process mining (PM) [3]. The field of PM developed from a purely academic topic to an area with substantial industrial uptake and market potential. This is also reflected in a very dynamic PM market, where, especially in the last few years, several of the big tech players acquired PM tooling to extend their technology stack. Furthermore, Gartner started to publish a market guide in 2018 [39] and, in 2023, published for the first time a Magic Quadrant dedicated to PM tools [40].

The two key benefits of PM outlined by [22] can help explaining the industrial success of the discipline. One key driver for engaging with PM is that PM can help organizations to create a “current state” process that might be different from what is expected or known. Another pivotal factor for PM adoption in the industry is that many organizations have Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems in place to support their processes, such as Order-to-Pay (O2P) and Purchase-to-Pay (P2P) processes, whose execution is recorded in enterprise transaction systems (many of them already since decades). PM can help in a data-driven manner to access this data and provide insights into how these processes look like and how they are performing.

The starting point for PM is event data extracted from information systems with ERP and CRM systems being the most common event source. The extracted event data can then be used for process discovery, i.e., to mine the current process from the data [2]. This mined model can then be enriched with frequency or performance data to conduct performance analysis, or it can be used for conformance checking, i.e., to test whether the normative model of the process and log are aligned and to provide diagnostics. With the increasing uptake of Machine Learning, PM has evolved from taking a purely historic view to a more forward looking view, where the data is leveraged for predictions as well as improvements [2].

While PM has reached some great attention in industry, its application traditionally focuses on business processes (BPs). More recently, a paper called “Process science: the interdisciplinary study of continuous change” proposes a perspective which is substantially broader than the initial scope of PM [16]. While PM typically has focused on BPs with ERP and CRM systems as the main data source, process science is about leveraging trace data to study how phenomena evolve in terms of underlying sequences of events.

In this paper, we use this wider perspective and show how digital traces beyond classical BPs can be analyzed using a process lens. More specifically, we introduce four distinct cases and use them to provide concrete examples of process science in action. Sect. 2 introduces process science and process science activities. Sect. 3 then introduces four distinct process science

cases. Sect. 4, Sect. 5, and Sect. 6 then illustrate how the different cases approach the process science activities. Sect. 7 provides a process science case summary and distills lessons learned.

2 Process Science and Process Science Activities

Process Science is the “interdisciplinary study of continuous change” [16]. The notion of process refers to a “coherent series of changes that unfold over time and occur at multiple levels” [16]. Process Science leverages trace data to study how phenomena evolve in terms of underlying sequences of events. Process science activities can be divided into *discovery*, *explanation*, and *intervention*.

Discovery aims at capturing and describing processes [16]. This comprises techniques to *create representations of processes* using digital event data such as process discovery algorithms [1]. It also concerns *event-based architectures to organize data collection*—something which usually gets little attention in the scientific community, which often works with existing event logs and the assumption that event data can be extracted from a central data source. Moreover, it covers *computational methods to analyze the data and identify patterns* in processes, such as complex event processing (CEP) [21], or methods for analyzing event sequence data [64]. **Explanation** aims at understanding why, how and when a process unfolds [16]. This includes *methods for sense-making around processes in a specific context*, including research studying the context in which a pattern is situated. The goal in this stage is to identify cause-effect relationships embedded in a situational context. Finally, **interventions** aim at intervening and shaping the processes into a desired direction [16]. This includes *methods to develop and evaluate interventions to processes*. For example, design-oriented research could be used to develop interventions, which are then evaluated in terms of their effects in process event data.

A notable example of process science research is the work by Pentland et al. [47], which leverages digital trace data for analyzing, explaining, and predicting the dynamics of process change. An analysis of process structures over two years at four dermatology clinics reveals insights into process changes that were invisible so far. An overview of works studying change dynamics with digital trace data is provided in a recent systematic literature review by Franzoi et al. [29].

3 Introduction to Process Science Cases

This section introduces the four process science cases used throughout this paper with their underlying processes. Case 1 and Case 4 are processes we investigate as part of the ProAmbition [27] project funded by the Swiss

National Science Foundation (SNSF). In turn, Case 2 is in the focus of the SNSF funded project ProMiSE [70].

Case 1: Storage and Production Processes in a Smart Factory. Case 1 refers to a cyber-physical system and the storage and production processes in a smart factory [42,52]. For both of these processes the availability of process knowledge is high as the processes and activities are known. The processes are repetitive and well-structured and are therefore suitable candidates for process-driven execution achieving high degrees of automation [27]. Deviations of executions from the normative process models are usually only observed in case of exceptions (e.g., machine failures) [42].

Case 2: The Process of Process Mining. Case 2 refers to the *process of process mining* (PPM), i.e., the study of how process analysts act and think when engaging in a PM analysis. This process is typically conducted by a process analyst, who attempts to answer one or more analysis questions by interacting with PM software. Despite being often provided with analysis questions, process analysts engage in exploratory tasks to make sense of the data and discover insights that can help answer the question or generate new ones [66]. As such, the process of PM is a highly flexible and variable process characterized by analytic and cognitive activities that emerge during the analysis based on the question and the data at hand. The emergent and knowledge-intensive nature of this process makes it hard to know the relevant events and activities a-priori. Recently, the PEM4PPM model has been proposed [54] to analyze the PPM from a cognitive perspective, leveraging the theory of Prediction Error Minimization [23]. Still, our understanding of processes and activities happening both during the interaction with the software and in the minds of the analysts remains rather high-level.

Case 3: The Process of Comprehending Software Artifacts. Case 3 refers to the the attentional and cognitive processes of comprehending software artifacts (e.g., source code or process models). The former processes reflect the mental activities users perform when solving a task [13,43,46], while the latter processes reflect the way users' attention evolves over the different parts of the artifact throughout the task [37]. The processes and underlying activities are largely unknown a-priori. Similar to Case 2, this information emerges only as the process unfolds. Typically, these processes have high variability and low structuredness.

Case 4: Phlebotomy – The Process of Drawing Blood. Case 4 refers to the process of drawing blood in a healthcare context. For this process, guidelines defining process steps exist [62]. Additionally, guidelines for hand hygiene are relevant for the execution of this process [61]. The process is moderately flexible and variable and is, to a large extent, a manual process.

For example, the blood drawing guidelines prescribe to start with preliminary operations and then to perform the venipuncture; the hand hygiene guidelines require to perform an additional hand hygiene step before the venipuncture. Additionally, unanticipated events such as medical emergencies may lead to deviations from the prescribed flow and, in turn, the hand hygiene guidelines may trigger the execution of additional process steps. Similar to Case 2 and Case 3, this process is not suitable for being controlled by a process engine. Still, unlike the other two cases, we know process steps, guidelines, as well as relevant events and can monitor them.

4 Process Science in Action: Discovery

In this section, we focus on discovery, one of the process science activities, which is concerned with capturing and describing processes [16]. We illustrate different aspects of it along the four process science cases (cf. Sect. 3). More specifically, we cover the selection of data sources, data collection, the generation of event logs, and the analysis of event data.

4.1 Process Observability: Selection of Data Sources

To conduct process science activities, process observability is a key consideration. The extent to which we can investigate a certain phenomenon using process science largely depends on the extent to which relevant state changes can be observed and are reflected in the digital traces we use for analysis. Process observability can be often increased through the inclusion of additional data sources (e.g., placing additional sensors). Therefore, a well thought through selection of data sources plays a critical role toward better process observability. The four cases introduced earlier differ largely in the extent to which they can be observed and the nature of their underlying processes requires different ways to increase process observability.

Case 1: Processes of Storage and Production in a Smart Factory. The smart factory is equipped with sensors and actuators emitting events and allows observing state changes in the process to a large extent. Sensors include switches, light barriers, color sensors, environment sensor, a camera, and an NFC reader. Actuators include motors, compressors, and valves [42]. Events that are created by a process execution engine during process execution directly reflect state changes of the corresponding business process (e.g., a process instances has started, an activity has completed) (cf. Sect. 4.2). In turn, the events that are emitted by the smart factory are often low-level events, referring to IoT sensor readings or changes in machine states. An exemplary event emitted by a vacuum gripper robot (VGR) in the smart factory is shown in listing 1.1 [60]. The event has a list of attributes associated with

Listing 1.1. Example IoT event from smart factory.

```
{
  "id": "9ff5ad68-8771-4ea2-bff5-99dbb849e462",
  "station": "VGR_1",
  "timestamp": "2023-04-11 10:22:15.14",
  "i1_pos_switch": 0, "i2_pos_switch": 0,
  "i3_pos_switch": 0, "i7_light_barrier": 1,
  "i4_light_barrier": 1, "i8_color_sensor": 1038,
  "o7_compressor_level": 0, "o8_valve_open": 512,
  "m1_speed": 0, "m2_speed": -512, "m3_speed": 0,
  "current_pos_x": 421.87, "current_pos_y": 4.65,
  "current_pos_z": 25.81, "target_pos_x": 211.01,
  "target_pos_y": 0.0, "target_pos_z": 26.11}

```

a timestamp for the specific machine as payload. Each attribute represents a low-level IoT sensor reading as event that occurred at given timestamp.

When process execution is not driven by a dedicated process engine that already provides events at the process level, i.e., events reflecting relevant state changes in the process, the low-level nature of IoT events makes it necessary to bridge this abstraction gap. This entails detecting process events from the low-level events using event abstraction [57] and correlating them with the corresponding process instances [26].

Case 2: Process of Process Mining (PPM). For the PPM, the main goal is to better understand how process analysts address PM questions while interacting with different PM tools. Without additional sensor placement and instrumentation of such tools, the observability of the PPM is limited. In some cases, the interactions of the process analysts with the PM tools used to conduct different PM tasks might leave some traces (e.g., application logs depending on the tool). In other cases, screen recordings can be leveraged to derive tool interactions manually. The usage of task mining to monitor user interactions could be explored as a possible alternative approach to collect user interaction logs [3, Chapter 16]. Instead, if tool access is available, the respective tools could be instrumented (e.g., work on the Process of Process Modeling [49,48]). However, large parts of the process occur in the analyst's mind and require the collection of additional data to be understood. Think aloud data [56] can provide insights into the analyst's thinking processes and retrospective interviews can provide reflections after the analysis process, which can help to interpret and explain the observed behavior.

Case 3: Process of Comprehending Software Artifacts. In terms of process observability, the process of comprehending software artifacts highly resembles the PPM as large parts of the process occur in the developer's mind. When observing developers interacting with software artifacts (e.g., process

models or source code) in the context of comprehension tasks, observable tool interactions are restricted to navigation, scrolling, and zooming events leaving a limited amount of digital traces. Eye-tracking allows observing attentional processes and thus seems to be suitable to increase observability. Depending on the nature of the software artifact (i.e., static versus dynamic stimuli), additional instrumentation might be needed to be able to deal with zooming, scrolling, and file switching. Moreover, data collected from bio sensors such as Galvanic Skin Response (GSR) sensors or Electrocardiogram (ECG) sensors could be combined with pupillary response dilation collected via eye-tracking to measure cognitive load (CL), i.e., to measure how mentally demanding different parts of the artifact were perceived by the users.

Case 4: Phlebotomy – The Process of Drawing Blood. The process of drawing blood samples is a manual process characterized by a low degree of digitalization with most relevant actions being performed outside of an IT system. In this scenario, observability is low. For the blood drawing processes our goal is to support the user, i.e., the healthcare worker, to adhere to the existing hand hygiene guidelines. In order to do so, we need to be able to detect relevant state changes and events that might trigger hand hygiene activities during the execution of the process, for example hand contamination due to touching a non-sterile surface. To detect relevant state changes and events we envision using proximity, presence, pressure, touch, and flow sensors in combination to realize a non-invasive process monitoring system. We intentionally decided against video recordings for scene understanding, since this is likely to intrude the privacy of the involved process participants.

4.2 Data Collection

Apart from selecting the data sources that process observability is well supported, the data collection itself needs to be carefully planned to enable the linking of the collected data with the different elements of the process (e.g., process activities). In all of the settings introduced in Sect. 3 data is not collected by a single system with a centralized database but it originates from different distributed data sources. This section discusses three distinct data collection approaches that enable the above-mentioned linking:

- 1 the collection of data in the process context,
- 2 the collection of interaction logs linked to the artifact, and
- 3 the synchronized collection of data from different sources.

Collection of data in the process context When process execution is driven by a process engine, data can be collected in the process context. The

collection of process events (e.g., start and end of activity execution) are default capabilities of any process engine. However, data collection can go beyond process events and consider further event sources providing process-related data (e.g., IoT data), and embed it in the broader process context [45]. For example, for Case 1 IoT data could be collected in the process context. This results in IoT-enriched event logs, which associate sensor data with the corresponding process execution events [44]. The resulting event log can then be directly used by PM software. Moreover, IoT-enriched logs allow calculating metrics using IoT data and enriching the process model using that data (e.g., the average activity during a production activity).

Collection of interaction logs linked to the artifact For interactive scenarios where artifact usage is central, collecting interaction logs linked to the artifact usage (either via instrumentation or interaction logging) might help to deal with the lack of an execution engine. For Case 2 PM tools are the focal artifact and for Case 3 these are software artifacts such as process models or source code. For Case 2, studying the PPM across many different tools would require an enormous instrumentation effort to collect interaction logs linked to the PPM tools and was therefore practically not feasible for the specific setting of this case. Instead, for Case 3 we could instrument the process modeling tool and the IDE which were available as open-source to associate during data collection the user's gazes with the elements of the artifact that have been gazed (e.g., using iTrace [32] for source-code and EyeMind¹ for process models). In turn, this leaves us with enriched logs of gaze events which associate gaze data with corresponding elements of the artifact. As shown in Fig. 1a, each gaze data point (represented as a red circle) can be associated with the underlying process model element. During analysis time, we can then calculate eye-tracking metrics and enhance the artifact elements with these metrics. As depicted in Fig. 1b, the sub-process "Transport to next destination" was enhanced with a set of fixation, saccade and pupillary measures providing indications about users' attentional and cognitive states when reading this sub-process. This is similar to a performance analysis which can be done in classical PM tools where the discovered models are enhanced with time or performance information [2].

Synchronized data collection Whenever event data is collected from multiple distributed event sources, it needs to be collected in a synchronized manner to ensure that the timestamps of the different event sources are aligned. These are aspects which do not need any attention when data is extracted from a single event source, e.g., from an ERP system with a centralized database, that is typically assumed to be the case in classical PM settings. However, when data is collected from multiple and distributed event

¹ See <https://github.com/aminobest/EyeMind>

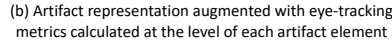


Fig. 1. Automated mapping of gazes to the model elements resulting in an enhanced artifact representation.

sources, synchronization becomes vital. All our cases leverage more than a single event source. For Case 1 and Case 4 we use a dedicated data collection application that adds the timestamps to the sensor readings. For Case 2, we set up a dedicated data collection platform to ensure that the timestamps of the different data sources, such as application logs, screen recordings and audio recordings of think aloud data, are synchronized. Synchronized data collection is also an important concern in Case 3 and is a requirement to later on align the bio signals (e.g., pupillary response data as well as GSR and ECG readings) with the data in the enriched log of gaze events. Literature has, for instance, shown that events associated with pupil dilation, increased skin conductivity (captured with GSR) or heart-rate variability, manifest in the signals when users' are experiencing increased CL. However, these events appear in the signal at different latencies [14,24,59]. In [50] for instance, authors use live event markers to synchronize different signals during the data collection, while in [7], an inductive approach was used to find the optimal alignment between pupillary and fixation data derived from gazes in a post-hoc manner (i.e., after the data collection).

4.3 Event Log Generation

In this section, we discuss how to generate event logs from the collected data. In the ideal scenario, data collection already provides process event logs. However, in most cases, additional processing is needed to generate event logs that are suitable for further analysis. The availability of process knowledge plays an important role in the choice of the approach. We describe the four distinct ways for event log generation we encountered in the context of our process science cases below:

- 1 Process event logs directly resulting from data collection,
- 2 Process and activity detection from sensor data,
- 3 Artifact elements as activity proxies, and
- 4 Screen recordings for generating interaction logs.

Process event logs directly resulting from data collection When process execution is process-driven (i.e., driven by a process execution engine like in Case 1) there is no need for explicit event log generation as this constitutes a default capability of the process engine. Moreover, when IoT data is collected in the process context (cf. Sect. 4.2) then event logs enriched with IoT data are obtained as a result of data collection [44]. To make the scenario amendable for online settings event streams can be used [17].

Process and activity detection from sensor data When data cannot be collected in the process context, sensor data can be leveraged for process and activity detection. This approach is especially promising for the process of drawing blood samples, where we can rely on existing guidelines and indications of how to conduct this process. IoT devices can be leveraged to detect relevant state changes like the start or completion of activities. Similar approaches toward the recognition of specific activities from specific sensors have been proposed before, e.g., [20,31]; however, here we aim at a general approach leveraging arbitrary IoT devices to detect arbitrary activities. Our idea is to apply an interactive approach where *activity signatures* are identified by the domain expert [51]. With activity signatures we essentially refer to multivariate time series of IoT data, which are associated with the execution of certain activities. These are automatically translated into CEP queries, which can be used to monitor the start and completion of process instances / activities [53], and emit corresponding log events. For example, we could associate the specific time series of IoT data from a proximity and a flow sensor positioned on a hand sanitizer dispenser with the execution of a hand hygiene activity, and have a CEP query emitting a hand hygiene log event whenever the corresponding activity signature is detected. Similarly, in [53] eye-tracking data together with user interaction logs is used to detect modeling phases (i.e., problem understanding, method finding, modeling, reconciliation, and validation) using supervised learning [53]. Taking a process science lens, we can consider these phases as process activities and define the scope of the process as a distinct modeling session.

Artifact elements as activity proxies For interactive scenarios where interactions can be linked to the artifact (cf. Sect. 4.2), artifact elements can be leveraged as activity proxies (event abstraction). For the software artifact comprehension setting (Case 3), we are interested in the attentional

processes of users when interacting with a process model during comprehension tasks. To this end, we consider each comprehension task as a process instance. We further consider each element of an artifact as an Area of Interest (AOI). When it comes to activities we can then consider each visit to an AOI (referring to a certain element of the artifact) as an activity proxy. Assuming, for example, that the gazes shown in Fig. 1 reflect one’s reading of the underlying process model from left to right, then our process instance would have the following prefix: “Enter invoice details” followed by “XOR split gateway 1”, “XOR merge gateway 1” and ‘issue invoice’. Note that the two consecutive fixations on “issue invoice” were merged into one visit to that AOI and, thus, correspond to one activity in our process instance. The resulting process instance represents a sequence of AOI visits that can be analyzed together with other process instances using PM techniques to infer common reading patterns in users’ eye-tracking data. A similar approach could also be pursued when users interact using eye-tracking glasses with physical objects rather than software artifacts. Object detection technologies can be used to detect the objects in a scene and fixations on different physical objects or parts thereof could serve as activity proxies [11,12].

Screen recordings for generating interaction logs In the absence of application logs, screen recordings capturing the process execution can be leveraged for event log generation. In the PPM scenario (Case 2) we rely on screen recordings that capture the user interactions with PM tools and artifacts. Since, in this context, processes and activities are largely unknown, we approach event log generation in a bottom-up manner. Hereby, decisions on what to consider as events are left to the analyst. This is quite challenging since it requires substantial data exploration and human decision-making to identify the events at a level of abstraction that is “useful” for the analysis and allows exploring user interactions from different viewpoints [67]. One possible way to look at the PPM is to describe relevant events from different perspectives [15], e.g., by considering the tool functions used or the target artifact of the analysis. Fig. 2 illustrates an example interaction log generated from screen recordings. From the recordings, an event sequence is first created by adding events that denote when the analyst started and stopped using a specific tool function. For example, “Case Filter START” is an event with timestamp 00:09:38, indicating that the analyst started using a filter that enables filtering event log traces. Then, start and end events about the same tool function are grouped together to demarcate time segments and create a sequence of activities, which can be enriched with additional attributes, such as the PM tool used. What makes such a segmentation approach interesting is that many different labels can be added to the same segment, e.g., to describe the data on which the tool function was applied. The main drawback of the event log generation approach illustrated

in Fig. 2 is that it is manual and time-consuming. While the generation of some event sequences could be automated by leveraging application logs, decisions about how to describe the observed interactions and at what level of abstraction require the involvement of a human analyst.

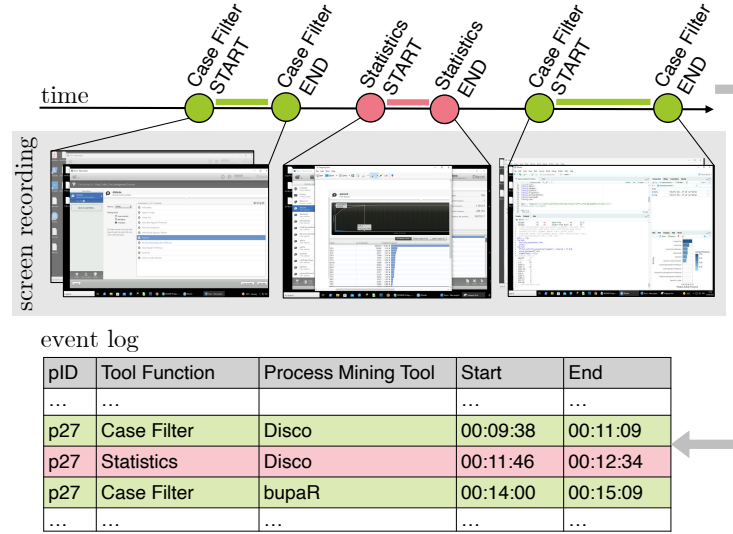


Fig. 2. Example of event log generation from screen recordings. In the log, attribute *pID* identifies a specific user. Rows are colored based on the *Tool Function* used.

When a model of user behavior is available, then the event log generation could be conducted in a top-down manner, i.e., considering known events to segment the screen recordings. For example, the PEM4PPM model presented in [54] allows identifying cognitive phases of the PPM that can be observed in screen recordings augmented with think aloud data.

4.4 Analysis of Event Data

Once we have event logs available, we can analyze the event data and describe how the process unfolds. This includes the application of state-of-the-art process discovery algorithms [1,10], techniques for visualizing and analysing event sequences [64,63], but also techniques for conformance checking [18,19] as well as process enhancements [1].

Process discovery for mining user behavior patterns We can leverage the collected data to mine different behavior patterns when interacting with the software artifact. Moreover, we can split the data (e.g., traces belonging

to different user groups) and apply process discovery techniques to the different splits and then explore differences between the different groups using the mined models. Related to Case 3 we developed a method based on eye-tracking to extract attention maps describing how developers interact with different artifacts while conducting comprehension tasks [38]. The method was demonstrated by investigating how developers engage with different software artifacts that are typically used as part of Behavior-Driven Development (BDD) (feature, step code, source code). We used the mined attention maps to compare developers' attentional processes at different granularity levels (i.e., using file types and files as activity proxies) depending on the type of comprehension tasks (i.e., domain understanding and code understanding tasks). The attention maps showed that the attentional processes unfold differently depending on the task type. To improve the interpretation of the results presented using attention maps, behavior data was complemented with retrospective think aloud data to analyze the benefits and the challenges associated with the usage of different artifacts.

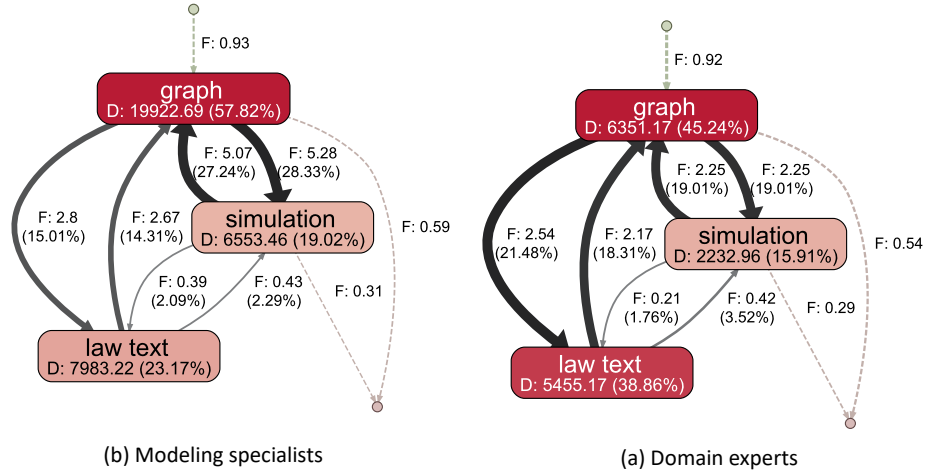


Fig. 3. Attention maps in form of Directly-Follow-Graphs comparing the attentional processes for modeling specialists and domain experts. Abbreviations: D: average fixation duration. F: average frequency of transitions between AOIs. Adapted from [8].

In turn, in [8], we used attentional processes to investigate how domain experts and modeling specialists use a hybrid process model representation [6] describing an article of the Danish law. The hybrid representation comprises (1) a declarative process model in the Dynamic Condition Response (DCR) language [36], (2) fragments of the law text associated to each relevant process model activity and (3) a guided simulation tool allowing us to

perceive the process behavior throughout the allowed execution scenarios. Following the approach described in Sect. 4.3, we have generated an event log leveraging the artifact elements (i.e., the declarative process model, the law text and the simulation) as activity proxies. Subsequently, we have used state-of-the art process discovery techniques to mine dependency graphs (i.e., attention maps) allowing us to study and compare the attentional processes of domain experts and modeling specialists. The resulting attention maps are illustrated in Fig. 3. The analysis of these maps [8] showed that domain experts relied more on the law text to make sense of the process and used the declarative process model for orientation purposes, while modeling specialists used primarily the declarative process model. In turn, based on this analysis, we suggested that providing a hybrid process model representation can help fitting the individual needs of user groups with different backgrounds and promote the citizen developer initiative by providing alternative approaches for non-modeling experts to understand process models.

Capturing processes from multi-perspectives Besides taking a process view (model-based analysis), we can look at individual traces (instance-based analysis). For example, we can visualize the event sequences for a single user and look at the data from multiple perspectives by creating not just a single event sequence but multiple ones. In Fig. 4, we can see the multi-perspective profile created from the event sequence of a process mining analyst with good PM expertise using Celonis². The profiles capture user behavior from the perspective of analysis tasks, i.e., the analyst visited multiple artifacts and tool functions, and its results, i.e., showing when the analyst arrived at finding points. The profile also illustrates to what extent the analysis was done based on filtered data. From user profiles, we can look at specific event subsequences, e.g., taking a window around finding points. Fig. 4 shows an example of a finding, which we can derive from screen recordings and we can look at the tool functions used and at the artifacts used in its surroundings to obtain a better understanding of the process.

Conformance checking and process monitoring In the smart manufacturing example (Case 1), event data is collected automatically and stored in an event log by the process engine executing the storage and production processes [52]. This data can be used for traditional PM activities (process discovery, process conformance, process optimization), both in an offline or an online setting. For our healthcare example (Case 4), we can leverage the collected data about an ongoing execution and monitor for hand hygiene indications, i.e., moments in the process where the healthcare worker should perform a hand hygiene activity as a pre- or postcondition for a certain activity or event according to the hand hygiene guidelines. For this, we plan

² Celonis: <https://www.celonis.com>

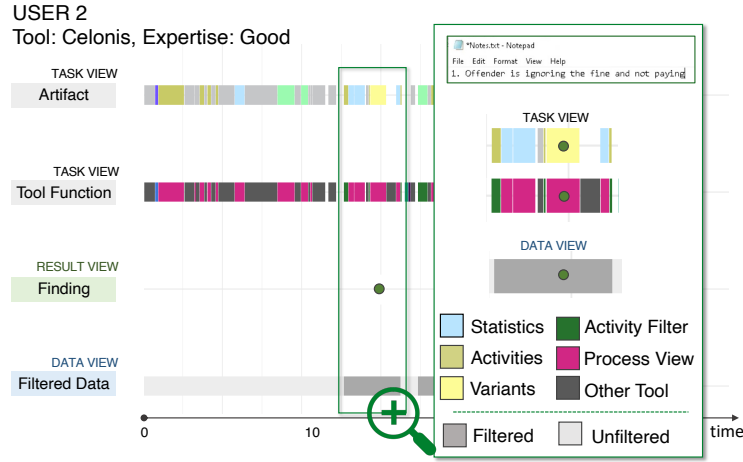


Fig. 4. Example of user profile capturing user behavior from multiple perspectives. Different colors in the scarf plots indicate that different *Artifacts* resp. *Tool Functions* were used during the PPM.

to develop a monitoring architecture leveraging the generated CEP queries for event detection [28], more specifically for the detection of indications of hand hygiene. This approach is related to leveraging a-priori knowledge for activity sequence prediction in predictive process monitoring [25], with the difference that here it is anchored to process knowledge in terms of (informal) specifications, i.e., guidelines, instead of past process executions. The availability of a process specification can be exploited also for conformance checking: we can retrospectively check for violations of the hand hygiene indications, which might be relevant for quality assurance purposes. Moreover, we plan to develop online conformance checking techniques to provide interpretable feedback at runtime to process participants (cf. Sect. 6.2).

5 Process Science in Action: Explanation

In this section, we focus on explanation which is about understanding, why, how, and when a process unfolds [16].

5.1 Providing Context to Activity Executions with IoT Event Data

As stated in Sect. 4.2, process event data collected by a process engine can be enriched with available IoT data recorded during activity execution [44]. This data cannot only be used for enhancements, but it can also provide additional contextual data regarding the execution of an activity (e.g., referring

to environmental conditions). The result of this correlation can then be leveraged for analyzing compliance aspects (e.g., in cold chain logistics [55]) or quality assurance (e.g., in smart manufacturing). In our previous work [52], we propose to apply CEP for analyzing IoT events produced by an environment sensor in parallel to the execution of an automated production activity to ensure that specific constraints are met during the execution.

5.2 Providing Context to User Interactions Logs

To explain user behavior, observations alone are often not sufficient. Indeed, every interaction with the PM tool is performed in the context of a specific analysis step or goal, which is usually hidden in the mind of the analyst and depends on many factors, such as previous analysis steps and findings or the personal experience of the analyst. In the PPM case, from the screen recordings alone, we can see what a PM analyst is doing in terms of what actions are performed on which objects. However, we lack information about why the analysis is conducted in a certain way. Understanding the rationale behind specific interactions is fundamental to explaining user behavior. To put the behavior captured by screen recordings into context, we leverage verbal data which we have collected in a synchronized manner (cf. Sect. 4.2). In the remainder, we will exemplify this for concrete user interactions using think aloud and interview data. Starting from a low level event sequence focusing on tool functions, as the one shown in the log of Fig. 2, we can use think aloud data to provide context to specific events. In this example, the user describes the event Case Filter starting at minute 09:38 as “I’m trying to filter the Payment activity to see all the cases that we don’t have a payment”, allowing us to understand what is the intention of the user behind the application of that specific case filter and how this relates to the question posed by the analysis. Similarly, interview data can be leveraged to add an additional layer of abstraction and contextualize the observed interactions within analysis strategies [68]. For example, the three events in the lower part of Fig. 2 can be explained as the strategy of verifying artifacts and findings, based on the user interview, which reports “Often I try to combine different tools to understand for sanity check if we have the same insights.”

5.3 Providing Context to the User’s Cognitive and Affective States

To infer the user’s cognitive and affective states when reading a software artifact, a set of bio-measures can be extracted from the recorded bio-signal. As this signal was collected in a synchronized manner (cf. Sect. 4.2), an event-based analysis or a temporal bracketing approach [34] can be used to study and correlate the bio-measures computed over different time bins to infer the user’s cognitive and affective states. Remember that the signal from which the states were derived was aligned with the gaze events that

were enriched with information about the gazed element of the artifact (cf. Sect. 4.2). Thus, it is possible to contextualize the user’s cognitive and affective states, by directly inferring the elements or parts of the artifact that were gazed by the user when a specific state occurred. Subsequently, We can leverage the characteristics (e.g., complexity or visual representation) of the respective artifact parts as potential explanations for the observed cognitive or affective state. Without this novel approach, we can assess the user’s cognitive and affective state associated with an entire source code snippet or process model. With the proposed contextualization of the user’s cognitive and affective states, we can delve further into the artifact, identify the problematic parts of it, investigate their characteristics and use them to explain the user’s cognitive and affective states. This step is also a pre-condition for providing interpretable feedback and for designing event-driven software systems that can provide users with context-specific support (cf. Sect. 6.2).

6 Process Science in Action: Interventions

Interventions are about intervening and shaping the process in a desired direction [16]. Insights obtained during discovery and explanation can inform the development of novel artifacts that better support the underlying process. Moreover, interpretable feedback to the user and event-driven software systems which adapt in response to detected events are possible ways to steer the process in the desired direction.

6.1 From Process Insights to Improved Tool Support

Insights about the underlying process can inform tool development. Concerning Case 2, better insights into the PPM led to the development of novel support for exploratory PM and for the rigorous formulation and testing of hypotheses.

Support for exploratory process mining While analyzing the PPM, we observed the knowledge-intensive and ad-hoc character of exploratory PM that develops step-by-step based on emerging insights. Based on such observations, we derived requirements for a system to support exploratory PM [65]. In particular, we found that there is a need to maintain analytic provenance information about the analysis steps, to enable analysts to trace their analysis goals and insights and to increase the awareness of what piece of the analyzed data led to a specific result. The system we proposed in [65] leverages a replayable history of interactions as well as integrated provenance and data views to address these requirements.

Support for the rigorous formulation and testing of hypotheses Our analysis of the PPM also showed that prominent analysis steps in the PPM are generating and testing hypotheses [69], for example about correlations in the event data that can explain desirable or undesirable process outcomes. However, tool support for rigorously formulating such hypotheses, testing them and for finding new hypotheses is still very limited. This is especially true for hypotheses across multiple process perspectives, i.e., control flow, data, time, and resources.

In recent work [58], we proposed a method that allows a process analyst to interactively find and test hypotheses that can explain a process outcome, improving support of multi-perspective hypotheses for process outcomes. A process outcome can be a normative deviation (e.g., the process does not arrive in a prescribed end state), a performance outlier (e.g., sometimes the process takes longer than 5 hours to complete) or an outlier of any other relevant metric (e.g., the utilization of a resource is below 50%). An explanation for an outcome O is a proposition of the form *if C then O* where C is a condition on the traces of the process that is meaningful to the process analyst. For example, if O represents that a normative end state of the process was not reached, e.g., “the invoice was not fully paid”, an explanation E could represent that a very similar but formally different end state was reached, e.g., “the invoice was paid apart from some negligible missing amount”. The quality of the explanation can be rigorously tested on the event log in terms of precision and recall. Multiple explanations might be needed to fully explain a specific process outcome, i.e., to achieve a high recall. The support for the analyst to find meaningful explanations is powered by automated rule induction [30] but must be carefully harnessed by techniques described in [58] to become effective in this use case.

We also proposed in [58] a novel way of visualizing how the main process design relates to the distribution of process outcomes. That novel visualization is based on Sankey diagrams and shows to what extent each global decision of the process contributes to each process outcome. The process analyst can thus pick up more easily which combination of process decisions contribute to bad outcomes for example. To limit the complexity of the Sankey diagram, the process analyst can select only those process decisions that they deem to be relevant for the process outcome distribution.

6.2 Interpretable Feedback

Interpretable Feedback is a form of intervention that can be provided in both offline and online settings. For instance, the ability to pinpoint the difficult parts of an artifact allows providing offline feedback to users that can be used to support review activities [35] and guide the attention of reviewers. As illustrated in Fig. 5a, when reviewing an artifact (e.g., source code), the reviewers can be pointed to the exact parts of the artifact (highlighted in red

in Fig. 5a) where the user (e.g., developer) was challenged and experienced and increasing cognitive load.

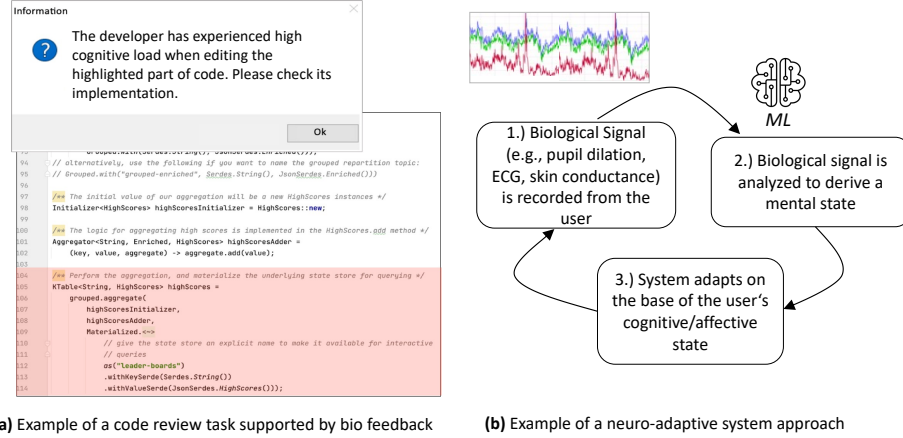


Fig. 5. Examples of different forms of interventions.

The provision of interpretable feedback to the users is also something we plan to pursue for Case 4 in an online setting. Let us assume that we have discovered non-conforming behavior in one execution of the blood drawing process. For example, we observed that a healthcare worker did not perform a hand hygiene activity before proceeding with the venipuncture activity, thus exposing the patient to the risk of infection. In this case, we could inform the user about nonconforming behavior (e.g., in a training setting), indicate the correct course of action, and leverage the process context to provide interpretable feedback and explain to the user that (s)he should have performed a hand hygiene step according to *indication #2: “before aseptic procedure”* [61] before the venipuncture activity. We envision that interpretable feedback could be provided not only in an online phase, during the process execution, but also in an offline phase, after the process has terminated. The modalities in which interpretable feedback is provided might differ, depending on the phase (online versus offline) and setting (e.g., training versus normal operation).

6.3 Event-driven Software Systems

Another form of intervention are event-driven software systems which adapt in response to detected events. For example, in the context of Case 3, we can build neuro-adaptive software systems (a specific form of event-driven system) following the approach summarized in Fig. 5b [50]. These systems can

monitor user’s bio signal and gaze locations, and then use machine learning techniques to infer their affective and cognitive states. Knowing when the user faces difficulties (from the bio signal) and knowing where these difficulties stem from (from the gaze location) is the basis for a system that can adapt or intervene to provide a targeted support. Our work in [7] makes a first stride toward this vision by developing machine learning models that can pinpoint the parts of code where difficulty originates from. Moreover, in the context of Case 1 we can monitor the production processes and respond to the detection of events. For example, depending on the evaluation of the context data, the product may need to undergo additional checks in the production process to ensure the desired quality [52].

7 Process Science Case Summary and Lessons Learned

In this paper, we discussed the different process science cases in Sect. 3 along the different process science activities, i.e., discovery (cf. Sect. 4), explanation (cf. Sect. 5), and intervention (cf. Sect. 6). Table 1 in Appendix A provides a summary of the different process science cases in terms of their characteristics and how the different process science activities were approached. It shows that digital traces come in many flavors as do the underlying processes. The table also highlights that process knowledge next to how the process is executed (i.e., fully automated, interactive, or manual) largely influences the way how we create event sequences and analyze the data. The four cases illustrate that there is value in leveraging digital trace data beyond traditional business processes, e.g., for interpretable feedback, the development of event-driven systems, and for data-driven tool support. At the same time, the four cases also show that there are challenges in terms of process observability, event correlation, and event abstraction.

Working on different process science cases has shown the **importance of a well planned data collection**. This becomes even more important when you plan to go beyond a single data source and centrally collected data. All our cases relied on multiple data sources, which we leveraged for the contextualization of events and for explanations. However, this requires that data is collected in a way that it can be linked with the different elements of the process. Based on our experience, we recommend to carefully evaluate whether data collection in the process context is feasible (similar to Case 1), since this approach provides already event logs in the desired form and thus dramatically decreases the effort for event log generation. Post-hoc linking the collected data with the different elements of a process is usually complex and may not be precise (e.g., due to a lack of synchronization between different data sources during data collection). For interactive scenarios, we recommend evaluating whether the software artifact (or alternatively the physical objects) can be leveraged. Similar to Case 3, the

structure of the software artifact can be exploited for a fine-grained mapping between gaze events and the underlying artifact elements serving as activity proxies in the construction of the event log. Alternatively, users interactions or actions on target objects can be associated with process activities following pre-defined reference models such as the one proposed in [5] to conceptualize user interactions and capture them in a process context.

The **insights** obtained from a process-oriented analysis are **highly dependent on the abstraction used to define events**. Therefore, events with different granularity can lead to the discovery of different patterns and yield different conclusions. Knowledge about process and activities can help to choose a proper abstraction level, i.e., where each event represents a relevant state change. For processes where this knowledge is limited, choosing the right level of granularity requires human decision-making. A multi-perspective analysis can mitigate this challenge to some extent (Case 2).

Concerning the analysis of event data, we also observed a **feedback loop** between discovered process knowledge from low-level events (e.g., IoT) and further analysis steps using the discovered process knowledge to analyze the low-level events (e.g., IoT data). For example, in the context of Case 1, identifying an activity from the IoT data helped to derive the process context which again informed the analysis of preceding or succeeding IoT data for identifying process activities. While analyzing and abstracting low-level events to higher level process levels events, the existing process knowledge and context should always be considered in addition to the low-level data since it helps to inform the next analysis steps and also to increase data quality (e.g., by resolving ambiguities originating from the low-level events [53]).

Case 2 and 3 deal with process data with high variability. Usually, traces are unique in these cases as they reflect users' unique behaviors. Therefore, some of the popular PM analyses (e.g., variant analysis) may not be applicable to this data without revisiting the existing approaches and algorithms, and adjusting them to support data with such characteristics (e.g., [4]).

ACKNOWLEDGEMENTS

The co-author Amine Abbad-Andalousi is supported by the International Postdoctoral Fellowship (IPF) Grant (Number: 1031574) from the University of St. Gallen in Switzerland. Moreover, this work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbition* project) and under Grant No. 200021_197032 (*ProMiSE* project).

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg, 2 edn. (2016)

2. van der Aalst, W.M.P.: Foundations of Process Discovery, pp. 37–75. Springer International Publishing, Cham (2022)
3. van der Aalst, W.M., Carmona, J.: Process mining handbook. Springer Nature (2022)
4. Abb, L., Bormann, C., van der Aa, H., Rehse, J.: Trace clustering for user behavior mining. In: 30th European Conference on Information Systems ECIS 2022 (2022)
5. Abb, L., Rehse, J.: A reference data model for process-related user interaction logs. In: Proc. BPM 2022. LNCS, vol. 13420, pp. 57–74. Springer (2022)
6. Abbad-Andaloussi, A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. *Information Systems* **91**, 101505 (2020)
7. Abbad-Andaloussi, A., Sorg, T., Weber, B.: Estimating developers’ cognitive load at a fine-grained level using eye-tracking measures. In: Proc. of IEEE/ACM International Conference on Program Comprehension. pp. 111–121 (2022)
8. Abbad-Andaloussi, A., Zerbato, F., Burattin, A., Slaats, T., Hildebrandt, T.T., Weber, B.: Exploring how users engage with hybrid process artifacts based on declarative process models: a behavioral analysis based on eye-tracking and think-aloud. *Software and Systems Modeling* **20**, 1437–1464 (2021)
9. Agosti, M., Crivellari, F., Di Nunzio, G.M.: Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. *Data Mining and Knowledge Discovery* **24**(3), 663–696 (2012)
10. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated discovery of process models from event logs: Review and benchmark. *IEEE TKDE* **31**(4), 686–705 (2018)
11. Barz, M., Kapp, S., Kuhn, J., Sonntag, D.: Automatic recognition and augmentation of attended objects in real-time using eye tracking and a head-mounted display. In: Proc. ETRA 2021. pp. 1–4 (2021)
12. Bektaş, K., Strecker, J., Mayer, S., Garcia, D.K., Hermann, J., Jenss, K.E., Antille, Y.S., Solèr, M.: Gear: Gaze-enabled augmented reality for human activity recognition. In: Proc. ETRA 2023. pp. 1–9 (2023)
13. Bera, P., Soffer, P., Parsons, J.: Using eye tracking to expose cognitive processes in understanding conceptual models. *MIS Quarterly* **43**(4), 1105–1126 (2019)
14. Bourdillon, N., Schmitt, L., Yazdani, S., Vesin, J.M., Millet, G.P.: Minimal window duration for accurate hrv recording in athletes. *Frontiers in neuroscience* **11**, 456 (2017)
15. Brehmer, M., Munzner, T.: A multi-level typology of abstract visualization tasks. *IEEE TVCG* **19**(12), 2376–2385 (2013)
16. vom Brocke, J., van der Aalst, W., Grisold, T., Kremser, W., Mendling, J., Pentland, B., Recker, J., Roeglinger, M., Rosemann, M., Weber, B.: Process science: the interdisciplinary study of continuous change. SSRN (September 2021)
17. Burattin, A., Eigenmann, M., Seiger, R., Weber, B.: Mqtt-xes: Real-time telemetry for process event data. BPM 2020 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track pp. 97–101 (2020)
18. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance checking. Switzerland: Springer.[Google Scholar] **56** (2018)
19. Carmona, J., van Dongen, B., Weidlich, M.: Conformance checking: foundations, milestones and challenges, pp. 155–190. Springer (2022)

20. Cornacchia, M., Ozcan, K., Zheng, Y., Velipasalar, S.: A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal* **17**(2), 386–403 (2017)
21. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)* **44**(3), 1–62 (2012)
22. Davenport, T.H., Spanyi, A.: What process mining is, and why companies should do it. *Harvard Business Review* (April 2019)
23. de Bruin, L., Michael, J.: Prediction error minimization: Implications for embodied cognition and the extended mind hypothesis. *Brain and Cognition* **112**, 58–63 (2017)
24. Denison, R.N., Parker, J.A., Carrasco, M.: Modeling pupil responses to rapid sequential events. *Behavior research methods* **52**, 1991–2007 (2020)
25. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An eye into the future: leveraging a-priori knowledge in predictive business process monitoring. In: *Proc. BPM 2017*. pp. 252–268. Springer (2017)
26. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10**(3), e1346 (2020)
27. Franceschetti, M., Seiger, R., González González, M.J., Garcia-Ceja, E., Rodríguez Flores, L.A., García-Bañuelos, L., Weber, B.: ProAmbitIon: Online Process Conformance Checking with Ambiguities Driven by the Internet of Things. In: *CEUR Workshop Proceedings*. vol. 3413, pp. 52–59 (2023)
28. Franceschetti, M., Seiger, R., Weber, B.: An event-centric metamodel for iot-driven process monitoring and conformance checking. In: *International Conference on Business Process Management Workshops*. Springer (2023)
29. Franzoi, S., Grisold, T., vom Brocke, J.: Studying dynamics and change with digital trace data: a systematic literature review. In: *Proc. ECIS 2023* (2023)
30. Fürnkranz, J., Gamberger, D., Lavrač, N.: *Foundations of Rule Learning*. Springer (2012)
31. Garcia-Ceja, E., Brena, R.F.: Activity recognition using community data to complement small amounts of labeled instances. *Sensors* **16**(6) (2016)
32. Guarnera, D.T., Bryant, C.A., Mishra, A., Maletic, J.I., Sharif, B.: Itrace: Eye tracking infrastructure for development environments. In: *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications. ETRA '18*, Association for Computing Machinery, New York, NY, USA (2018)
33. Guo, Y., Guo, S., Jin, Z., Kaul, S., Gotz, D., Cao, N.: Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics* **28**(12), 5091–5112 (2022)
34. Hartl, S., Franzoi, S., Grisold, T., Vom Brocke, J.: Explaining change with digital trace data: A framework for temporal bracketing (2023)
35. Hijazi, H., Cruz, J., Castelhana, J., Couceiro, R., Castelo-Branco, M., de Carvalho, P., Madeira, H.: ireview: an intelligent code review evaluation tool using biofeedback. In: *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*. pp. 476–485. IEEE (2021)
36. Hildebrandt, T.T., Mukkamala, R.R.: Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. *EPTCS* **69**, 59–73 (2011)
37. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., van de Weijer, J.: *Eye Tracking: A comprehensive guide to methods and measures*. OUP Oxford (2011)

38. Ioannou, C., Nurdiani, I., Burattin, A., Weber, B.: Mining reading patterns from eye-tracking data: method and demonstration. *Softw. Syst. Model.* **19**(2), 345–369 (2020)
39. Kerremans, M.: Market guide for process mining (2018), <https://www.gartner.com/en/documents/3870291/market-guide-for-process-mining>
40. Kerremans, M., K. Iijima, A.S., Duffy, N., Sugden, D.: Magic quadrant for process mining tools (2023), <https://www.gartner.com/doc/reprints?id=1-2CZI8XWU&ct=230320&st=sb>
41. Luckham, D.C.: Event processing for business: organizing the real-time enterprise. Wiley, [S.l.] (2011), the book can be consulted by contacting: PH/ADT: Magnoni, Luca
42. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using physical factory simulation models for business process management research. In: International Conference on Business Process Management. pp. 95–107. Springer (2020)
43. Malinova, M., Mendling, J.: Cognitive diagram understanding and task performance in systems analysis and design. *MIS Q.* **45**(4), 2101–2158 (2021)
44. Mangler, J., Grüger, J., Malburg, L., Ehrendorfer, M., Bertrand, Y., Benzin, J., Rinderle-Ma, S., Asensio, E.S., Bergmann, R.: Datastream XES extension: Embedding iot sensor data into extensible event stream logs. *Future Internet* **15**(3), 109 (2023)
45. Pauker, F., Mangler, J., Rinderle-Ma, S., Pollak, C.: centurio.work - modular secure manufacturing orchestration. In: 16th International Conference on Business Process Management 2018. pp. 164–171. Dissertation Award, Demonstration, and Industrial Track, BPM 2018 (September 2018)
46. Peitek, N., Siegmund, J., Apel, S., Kästner, C., Parnin, C., Bethmann, A., Leich, T., Saake, G., Brechmann, A.: A look into programmers’ heads. *IEEE Trans. Software Eng.* **46**(4), 442–462 (2020)
47. Pentland, B.T., Vaast, E., Wolf, J.R.: Theorizing process dynamics with directed graphs: A diachronic analysis of digital trace data. *MIS Q.* **45**(2) (2021)
48. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugall, S., Weber, B., Reijers, H.A., Mendling, J.: Styles in business process modeling: an exploration and a model. *Software & Systems Modeling* **14**, 1055–1080 (2015)
49. Pinggera, J., Zugall, S., Weber, B.: Investigating the process of process modeling with cheetah experimental platform. In: EMISA Forum: Vol. 30, No. 2. De Gruyter (2010)
50. Riedl, R., Léger, P.M.: Fundamentals of neurois. *Studies in neuroscience, psychology and behavioral economics* p. 127 (2016)
51. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from iot data. *Future Internet* **15**(2) (2023)
52. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: A systems architecture and use cases. *Journal of Manufacturing Systems* **63**, 575–592 (2022)
53. 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: 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW). pp. 20–26 (2020)
54. Sorokina, E., Soffer, P., Hadar, I., Leron, U., Zerbato, F., Weber, B.: PEM4PPM: A cognitive perspective on the process of process mining. In: Int. Conf. on Business Process Management (BPM). LNCS (2023), accepted for publication

55. Tsang, Y.P., Choy, K.L., Wu, C.H., Ho, G.T., Lam, C.H., Koo, P.: An internet of things (iot)-based risk monitoring system for managing cold supply chain risks. *Industrial Management & Data Systems* **118**(7), 1432–1462 (2018)
56. Van Someren, M., Barnard, Y.F., Sandberg, J.: The think aloud method: a practical approach to modelling cognitive. London: AcademicPress **11**, 29–41 (1994)
57. van Zelst, S., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. *Granular Computing* **6**(3), 719–736 (Jul 2021)
58. Völzer, H., Zerbato, F., Sulzer, T., Weber, B.: A fresh approach to analyze process outcomes. In: 5th International Conference on Process Mining, ICPM 2023, Rome, Italy, October 23–27, 2023. IEEE (2023), accepted for publication.
59. Weber, B., Fischer, T., Riedl, R.: Brain and autonomic nervous system activity measurement in software engineering: A systematic literature review. *Journal of Systems and Software* **178**, 110946 (2021)
60. Weyers, F., Seiger, R., Weber, B.: Method to identify process activities by visualizing sensor events. In: Business Process Management Workshops. pp. 76–88. Springer International Publishing, Cham (2023)
61. World Health Organization: Who guidelines on hand hygiene in health care. In: WHO guidelines on hand hygiene in health care, pp. 270–270. World Health Organization (2009)
62. World Health Organization: WHO guidelines on drawing blood: best practices in phlebotomy. World Health Organization (2010)
63. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Visual drift detection for event sequence data of business processes. *IEEE Transactions on Visualization and Computer Graphics* **28**(8), 3050–3068 (2021)
64. Yeshchenko, A., Mendling, J.: A survey of approaches for event sequence analysis and visualization using the ESeVis framework (2022)
65. Zerbato, F., Burattin, A., Völzer, H., Becker, P.N., Boscaini, E., Weber, B.: Supporting provenance and data awareness in exploratory process mining. In: Advanced Information Systems Engineering. pp. 454–470. Springer Nature Switzerland, Cham (2023)
66. Zerbato, F., Koorn, J.J., Beerepoot, I., Weber, B., Reijers, H.A.: On the Origin of Questions in Process Mining Projects. In: Enterprise Design, Operations, and Computing (EDOC). vol. 13585, pp. 165–181 (2022)
67. Zerbato, F., Seiger, R., Di Federico, G., Burattin, A., Weber, B.: Granularity in process mining: Can we fix it? In: CEUR Workshop Proceedings. vol. 2938, pp. 40–44 (2021)
68. Zerbato, F., Soffer, P., Weber, B.: Process mining practices: Evidence from interviews. In: Int. Conf. on Business Process Management. pp. 268–285. Springer (2022)
69. Zerbato, F., Soffer, P., Weber, B.: Process mining practices: Evidence from interviews. In: BPM. pp. 268–285. Springer (2022)
70. Zerbato, F., Zimmermann, L., Völzer, H., Weber, B.: ProMiSE: Process Mining Support for End-Users. In: CEUR Workshop Proceedings. vol. 3413, pp. 60–67 (2023)

Appendix A: Process Science Case Summary

Table 1: Process Science Case Summary

	Case 1: Processes of Storage and Production in a Smart Factory	Case 2: Process of Process Mining	Case 3: Process of Comprehend- ing Software Artifacts	Case 4: Process of Drawing Blood Samples
Nature of the process	Repetitive and highly structured; deviations only in case of exceptions	Highly flexible and variable	Highly flexible and variable	Moderately flexible and variable
Availability of process knowledge	Processes and activities are known	Processes and activities are largely unknown	Processes and activities are largely unknown	Guidelines including process steps are available
Process execution	Automated and potentially process-driven	Interactive	Interactive	Manual
Data collection	Process-driven execution and data collection resulting in an IoT-enriched event log	Synchronized data collection of interaction data and verbal data	Interaction data linked to the artifact resulting in a log of gaze events enriched with links to artifact elements; synchronized data collection to establish the link to further biosignals	Data from heterogeneous, distributed IoT devices (sensors, machines) deployed in the patient's and caretaker's surroundings

Continued on next page

Table 1 – Continued from previous page

Event log generation	IoT-enriched event log directly usable	Process scope is an analysis task; user interaction logs are created from screen recordings	Process scope is a comprehension task; artifact elements are used as activity proxies	Complex event processing is used to monitor the start and execution of process instances and activities, i.e., to generate process events, based on IoT data
Discovery	Processes have been designed and implemented by process engineers; they are known and do not need to be discovered; deviations from the normative process models have to be discovered (e.g., from machine data)	Mine patterns of behavior of individual users as they interact with process mining tools considering multiple perspectives e.g., tool functions, analysis artifacts and results	Mine attention maps in the form of dependency graphs allowing to study and compare the attentional processes of different user groups (e.g., domain experts versus modeling specialists)	Check process conformance with respect to guidelines
Explanations	Determine <i>why</i> exceptions and deviations from the normative process models occurred for specific executions, e.g., in cases of machine failures or equipment deterioration	Leverage verbal data to explain why certain patterns of behavior emerge and put them into the context of higher-level analysis and reasoning strategies	Use bio measures to continuously assess users' cognitive/affective states associated with different parts of an artifact	Inform about nonconforming behavior based on specified patterns/rules and sensor data; incorporate user feedback to explain nonconformance

Continued on next page

Table 1 – Continued from previous page

Interventions	Re-schedule activities, re-allocate resources, adapt the process instances	Develop data-driven tool support for exploratory PM and hypotheses generation and testing	Pinpoint the difficult parts of an artifact, support review tasks, build event-driven software systems	Provide interpretable feedback: inform about nonconforming behavior, indicate the right course of action, in both online and offline settings
----------------------	--	---	--	---