Project Description - Project Proposals

Henrik Leopold, Hamburg
Han van der Aa, Mannheim

Semantic Process Discovery from User Interaction Logs

Project Description

1 Starting Point

Process mining is widely used to discover, analyze, and improve business processes based on event data extracted from IT systems, stored in so-called event logs [6]. A key task in this regard is *process discovery*, which aims to reconstruct how a process was truly executed. To do so, process discovery strives to establish an accurate process model on the basis of the recorded behavior captured in an event log. Using such event logs as a basis for discovery has an important limitation, however: It limits the scope of analysis to *back-end events*, i.e., secondary, indirect events that were triggered by the actual user activity [23]. User activities that do not result in such back-end events or take place in productivity applications such as Excel and Outlook, are thus not recorded in event logs and, therefore, invisible to traditional process mining and discovery techniques.

To avoid this problem and be able to obtain a comprehensive view on business processes, the goal of this proposal is to enable process discovery based on *user interaction (UI) logs*, rather than on traditional event logs. In essence, a UI log is a collection of recorded interactions performed on GUI components, such as clicks on buttons or keyboard entries in text areas [7, 63]. The benefit of using UI logs is that they can be obtained for any business process of which the activities are performed on a computer, regardless of the specific applications required for it. Available *logging software* is then able to extract and store relevant data such as the interaction type (e.g. click or keyboard stroke), the time, and context (e.g., the GUI element and URL) in a UI log [35]. Figure 1 shows a simplified excerpt of such a UI log. Its events primarily show how a user receives orders via e-mail (cf., events 1 to 3) and proceeds to handle them in Salesforce's web application (cf., events 4 to 8).

ID	Timestamp	Event	Application	Element label	Element type	Element value	URL
1	08:35.2	click	Outlook	Customer X - O123	list	Please initiate an order ...	-
2	08:35.2	click	Outlook	Customer X - O234	list	Please initiate an order ...	-
3	08:35.2	click	Outlook	Customer Y - O789	list	Please initiate an order ...	-
4	08:39.7	click	Chrome	Log in	button	-	https://www.salesforce.com/
5	08:40.0	change	Chrome	Password	text field	-	https://login.salesforce.com/
6	08:40.5	click	Chrome	Submit	button	-	https://login.salesforce.com/
7	08:52.6	click	Chrome	New Account	button	-	https://com.lightning.force.com/home
8	08:53.2	change	Chrome	New Order	text field	Customer X	https://com.lightning.force.com/acc/
9	08:53.9	ctrl + c	Outlook	Customer X - O123	list	Please initiate an order ...	-
10	08:54.3	click	Chrome	Billing address	text field	-	https://com.lightning.force.com/acc/
11	08:54.4	ctrl + v	Chrome	Billing address	text field	Hofstraße 14, ...	https://com.lightning.force.com/acc/
12	08:54.9	click	Chrome	Save	button	-	https://com.lightning.force.com/acc/
13	08:40.0	change	Chrome	Password	text field	-	https://www.facebook.com/
14	08:42.9	click	Chrome	Log in	button	-	https://www.facebook.com/
15	08:42.9	click	Chrome	Messenger	button	-	https://www.facebook.com/
16	08:44.1	click	Chrome	New message	list	Hey, how are you? ...	https://www.facebook.com/
17	08:56.7	click	Outlook	Customer X - O234	list	Please initiate an order ...	-
18	08:58.2	change	Chrome	New Order	text field	Customer X	https://com.lightning.force.com/acc/
19	08:58.6	click	Chrome	Upload files	button	CustomerX-2021-O234.docx	https://com.lightning.force.com/acc/

Figure 1: Simplified excerpt of a UI log

Despite the obvious potential that UI logs have for process discovery, obtaining process information from them is a complex task, for which various problems need to be overcome. To tackle these, we structure our proposed project around two problem areas, each with its own specific challenges:

Problem area 1: Data transformation. A key problem is that UI logs do not meet the fundamental requirements that must be met by event logs used in process mining [6], i.e., (1) that events must

have a clear label, indicating the process steps to which they correspond, (2) that events in a log must all be related to a single process, and (3) that events must have a case identifier, allowing one to group together events related to the same process instance. Hence, to transform UI logs into event logs suitable for process mining, the following challenges need to be addressed:

Event annotation: Events need to be associated with an event label that defines the process step to which it corresponds. For instance, we observe that event 1 from Figure 1 was a "*click*" on a "*list*" in the application "*Outlook*". Yet from the perspective of a business process, the contents of the received e-mail reveal that this click relates to the receipt of a customer order. As such, to prepare a UI log for process mining, events need to be annotated with appropriate labels, such as "*Receive order*" for this particular case. How to infer such a label, however, strongly depends on the event's nature and its payload, and will, therefore, differ considerably from event to event.

Noise filtering: UI logs may contain events that do not relate to the process under investigation, so-called noise. To obtain proper process insights, such noisy events need to be identified and removed. Clear examples include events related to private activities, such as checking social media platforms (cf., events 13 to 16) or ordering things in a webshop. However, noisy events are not always so obvious, making their automatic recognition complex. A key reason it that an event that is considered irrelevant for one process, may be relevant for another one.

Case identification: Although case identifiers are fundamental to process mining, they are commonly missing from UI logs [34]. Yet, they may be derived by recognizing events that relate to the same process instance. This involves careful analysis of event attributes, combined with behavioral regularities. For example, next to event 1, also events 4 to 12 relate to order *O123* from Customer X. This, however, can only be properly inferred from event 11 where the billing address from Customer X is entered into the system. Similarly, by considering the filename uploaded to Salesforce in event 19, we can recognize that also the preceding events 17 and 18 relate to order *O234*.

Problem area 2: Process representation. Once a UI log has been transformed into an event log, the very low-level nature of the recorded UI events is still problematic. Particularly, applying process discovery directly on such logs will lead to so-called *spaghetti models*, which are too large and complex to provide useful insights into a process' execution. Therefore, this problem area focuses on the appropriate representation of low-level event data from UI systems, aiming to depict key process information at the right level of granularity. This again comes with three main challenges:

Event abstraction: To lift low-level UI data to the right level of granularity, a log's events need to be grouped together into higher-level business activities, e.g., by recognizing that events 8 to 12 jointly result in the creation of order *O123*. This abstraction task is highly challenging since business activities may be executed non-consecutively or comprise varying execution patterns.

Activity labeling: A useful process model requires clear and expressive labels. This means that the business activities, i.e., groups of events, resulting from event abstraction need to be labeled appropriately. For example, the group consisting of events 8 to 12 can be described by a "*Create order*" label, since this is the primary outcome its various low-level steps. Doing this automatically is very complex, however, and fundamentally differs from the annotation of individual events (cf., Problem area 1). In particular, it requires careful consideration of the meaning of individual events, as well as the role of the business activity in the larger context of the business process.

Process discovery and representation: Although various process discovery algorithms and visualizations have been developed, these are not tailored to the handling of abstracted, low-level events. Hence, they cannot provide users with an effective representation of the process captured in a UI log. The challenge, therefore, is to tackle both process discovery and subsequent process visualization in a manner that appropriately balances between both high and low-level information.

Project outcome. The proposed project will result in the development of approaches that address the aforementioned challenges in an automated manner, ultimately covering the entire pipeline from UI log to an informative process representation. We will achieve this by combining behavioral process analysis with a novel semantic angle, yielding approaches that overcome the limitations of existing works. In this way, the successful project will considerably advance state-of-the-art research in process mining, particularly for situations involving raw, low-level event data.

1.1 State of the art and preliminary work

1.1.1 State of the art

The proposed project primarily relates to research on traditional process mining (using event logs) and robotic process mining (using UI Logs). Below, we briefly review these streams and highlight the gaps that exist with respect to the challenges identified above.

Process mining. Process mining is a family of data analysis techniques that facilitate the discovery, analysis, and improvement of business processes [6]. The core idea of process mining techniques is to analyze so-called *event logs*. These event logs are extracted from information systems that support the execution of business processes and, therefore, capture how these processes are actually executed. Available process mining techniques serve a wide range of tasks including process discovery, conformance checking, and enhancement. Techniques for *process discovery* (see e.g. [28, 30, 70]) aim to provide the user with a visual representation of the process captured in the event log. *Conformance checking* techniques (see e.g. [8, 57]) detect differences between the actual and intended process execution by comparing the event log with a normative process model. Techniques for *enhancement* again address a variety of tasks such as predicting relevant aspects of the process execution [22] or repairing a given process model based on the event log [51]. Many of the challenges highlighted in the two problem areas above, also play a role in these "traditional" facets of process mining. Specifically, process mining techniques have been concerned with the challenges of noise filtering and case identification (problem area 1) as well as the challenges of event abstraction, event labeling, and model discovery (problem area 2):

Noise filtering: The removal of noise directly affects the quality of process models generated by process discovery techniques, as well as of other process mining results. Therefore, various discovery techniques take noise into account explicitly (e.g., [30, 71, 74]), whereas there are also dedicated techniques available for removing noise from events logs [19, 61]. However, one of the key assumptions of all these techniques is that noise is highly infrequent. This is problematic in our context, since if certain process-irrelevant actions (e.g., visiting social media websites) occur on a regular basis, they would not be recognized as noise by frequency-based techniques.

Case identification: The problem of case identification in event logs is addressed by various existing techniques (cf., [23] for an overview). However, they consider rather restrictive settings. The technique from Ferreira and Gillblad [26] provide a solution for processes that do not contain loops or activity repetitions, whereas the technique from Bayomie et al. [14] assumes that a process model is already available. Both are assumptions that will not be met in the context of our project.

Event abstraction: The issue of event abstraction has also been discussed in the context of traditional process mining [23, 75]. Recognizing that recorded events can differ widely in their granularity, several abstraction techniques have been proposed to obtain consistent and useful event logs or process models (e.g. [13, 36, 75]). Available techniques differ with respect to many aspects such as the type of supervision (supervised/ unsupervised), the handling of concurrency (yes/ no), and the type of output (probabilistic/ deterministic). What is currently still missing from the perspective of this project is an unsupervised technique that can can deal with the large degree of variability in UI logs and can reliably recognize respective higher-level business activities.

Event labeling: While the problem of labeling higher-level activities has been recognized and discussed in the context of process mining [24, 75], no dedicated techniques to overcome this problem currently exist. Instead, the currently proposed solution is to delegate this task to domain experts. For our goal of automated discovery of processes, this caveat must thus be addressed.

Model discovery: While various process discovery techniques exist (cf., [12] for an overview), the vast majority are designed to deal with higher-level event logs, rather than the low-level data found in UI logs. Closest to our goal is recent work on multi-level process discovery [31], which is the first to recognize the importance of granularity differences in discovery. Yet, the process models it yields lack intuitiveness, whereas it also imposes strong requirements on the input it can handle. As such, it provides a useful foundation for our project, but leaves a considerable gap to be addressed.

Robotic process mining. Robotic process automation (RPA) is a technology that aims to automate repetitive human work. The core idea is to let software robots (or bots) mimic the actions of a human directly in a GUI [60]. A key requirement of RPA is to actually identify automatable tasks. Recognizing this, the research domain of robotic process mining (RPM) has emerged in recent years [34]. The goal of RPM techniques is to automatically identify automatable routines based on UI logs. By doing so, RPM faces several challenges with the ones identified for the proposed project, primarily with respect to noise filtering and case identification from problem area 1:

Noise filtering: Also in the context of RPA, noisy events, such as social media visits, need to be removed from UI logs since they should not become part of automated procedures. However, effective solutions for noise removal from UI logs are missing. While noise removal techniques from traditional process mining (see above) are generally applicable, their limitations for removing noise from UI logs have also been recognized [34]. As a solution, some authors propose supervised noise removal based on an existing process model [10] or they suggest using rules [16, 33]. However, such process models cannot be expected to be available in the context of our project, whereas rule-based approaches are too rigid to deal with the flexibility of real-world UI logs.

Case identification: The identification of cases in RPM conceptually differs from case identification in traditional process mining. The underlying assumption in RPM is that cases do not overlap and, thus, that case identification can be achieved by splitting the log into segments. Researchers have proposed manual, supervised, and unsupervised approaches for this segmentation task. Urabe et al. [62] introduced a manual approach that visualizes the UI log using a graph and, in this way, supports the user in identifying segment boundaries. Agostinelli et al. [10] proposed a supervised approach, requiring a process model, that leverages trace alignments from conformance checking. Unsupervised approaches were introduced by Leno et al. [32, 33]. They construct a control-flow graph from the UI log and use back edges detection to identify segment boundaries. Another unsupervised approach from Urabe et al. [63] leverages the concept of co-occurrence from topic segmentation in natural language processing to segment the UI log. Despite the potential of these techniques, the assumption that cases are executed in a strictly sequential manner does not hold for many real-world settings, in which users may work concurrently on different cases (cf., Figure 1). Naturally, there also hybrid approaches available that combine supervised and unsupervised methods with human feedback. In a recent paper, Agostinelli et al. propose such a hybrid approach relying on frequent-pattern identification and human-in-the-loop interaction [9].

1.1.2 Preliminary work

The applicants, Prof. Leopold and Prof. Van der Aa, have published various works that relate to the proposed project. Both applicants have considerable expertise when its comes to the use of natural language processing (NLP) for the purposes of process analysis through individual as well as joint works, which forms a key component in the proposed solutions. Furthermore, the project relates to individual experience of the applicants with respect to the analysis of low-level event data (Prof. Van der Aa) and to the understandability of process representations (Prof. Leopold):

NLP for process analysis. The applicants have developed approaches for the extraction of semantic information, such as actions and business objects, from activity labels in process models [37, 39] and data attributes in event logs [53]. This expertise shall provide the foundation for event annotation in UI logs, a primary task in the proposed project. A key distinction, though, is that the existing works are designed to deal with short fragments (such as "*Create purchase order*"), whereas the project at hand shall also deal with larger texts, such as e-mails. In that regards the applicants can also build on their expertise when extracting process information from textual process descriptions, e.g., for the recognition of process constraints [3, 72] and for querying [38]. Here, a key distinction is that those approaches were designed for process-oriented texts, whereas the proposed project shall also deal with texts that are not structured in this manner.

Recently, the applicants showed the potential of employing semantic information in process mining when using extracted business objects and actions for the purposes of *semantic anomaly*

detection [5]. Specifically, the work demonstrates that NLP can be leveraged to detect process behavior that violates commonsense rules. This provides a novel angle for anomaly detection in comparison to existing works, which deem process behavior to be anomalous when it is infrequent. This work shall provide a starting point for the noise filtering that is required in the proposed project. However, next to the recognition of behavioral anomalies, the project also requires techniques that are specifically designed to recognize and remove non-business related events.

Analysis of low-level event data. Prof. Van der Aa has worked on several projects involving the analysis of low-level event data in both event logs and streams, which have clear similarities to the data granularity used in the proposed project. Existing work relates to the identification of key event patterns in streams [64], which can provide a basis for the recognition of behavioral regularities for case identification. Furthermore, experience on the efficient handling of large amounts of low-level events [76] can aid the computational efficiency of approaches developed for the proposed project, especially for tasks such as case identification and log abstraction that require global optimization. Finally, ongoing work on log abstraction with guarantees [54] can be used as a starting point for the meaningful abstraction of UI logs, since the characteristics of these logs can be used to guide the abstraction task, e.g., by ensuring that events from different systems are not grouped together.

Understandability of processes. Prof. Leopold has worked extensively on the topic of process model understandability, especially from a linguistic angle. Among others, he has developed techniques for automatically recognizing and correcting linguistic problems in process models that negatively affect the understandably of process models [39, 44, 50]. In this context, he has also proposed a technique that attempts to automatically determine names for (to be aggregated) process model fragments [42]. The challenge of labeling events that result from event abstraction in event logs is conceptually highly similar to labeling activities that result from activity abstraction in process models. However, the technique from [42] relies on the specifics of event-driven process chains (EPCs) and, therefore, cannot be transferred to labeling higher-level events. Nonetheless, these works provide important input for the challenges of problem area 2. The technical aspects represent an important basis for the challenges of event abstraction and activity labeling. The experience collected with user experiments, for instance in [50], will help us to successfully complete the challenge of process discovery and representation.

1.2 Project-related publications

Articles published by outlets with scientific quality assurance, book publications, and works accepted for publication but not yet published

Journal articles:

- A. Rebmann and **H. van der Aa**. “Enabling Semantics-aware Process Mining through the Automatic Annotation of Event Logs”. In: Information Systems 110 (2022), p. 102111. Reference [52].
- **H. van der Aa**, A. Rebmann, and **H. Leopold**. “Natural language-based detection of semantic execution anomalies in event logs”. In: Information Systems 102 (2021), p. 101824. Reference [5].
- **H. Leopold**, **H. van der Aa**, J. Offenberg, and H. A. Reijers. “Using Hidden Markov Models for the accurate linguistic analysis of process model activity labels”. In: Information Systems 83 (2019), pp. 30–39. Reference [37].
- F. Pittke, **H. Leopold**, J. Mendling: Automatic Detection and Resolution of Lexical Ambiguity in Process Models. In: IEEE Transactions on Software Engineering (2015), 41(6): 526-544. Reference [50].
- **H. van der Aa**, **H. Leopold**, and H. A. Reijers. “Checking process compliance against natural language specifications using behavioral spaces”. In: Information Systems 78 (2018), pp. 83–95. Reference [4].
- **H. Leopold**, J. Mendling, H. A. Reijers, and M. La Rosa. “Simplifying process model abstraction: Techniques for generating model names”. In: Information Systems 39 (2014), pp. 134–151. Reference [42].

Conference papers:

- A. Rebmann, M. Weidlich, and **H. van der Aa**. “GECCO: Constraint-driven Abstraction of Low-level Event Logs”. In: International Conference on Data Engineering. 2022. Reference [54].
- **H. van der Aa**, C. Di Ciccio, **H. Leopold**, and H. A. Reijers. “Extracting declarative process models from natural language”. In: International Conference on Advanced Information Systems Engineering. Springer.

2019, pp. 365– 382. Reference [3].

- **H. van der Aa**, J. Carmona Vargas, **H. Leopold**, J. Mendling, and L. Padró. “Challenges and opportunities of applying natural language processing in business process management”. In: International Conference on Computational Linguistics. Association for Computational Linguistics. 2018, pp. 2791–2801. Reference [2].
- **H. Leopold**, C. Meilicke, M. Fellmann, F. Pittke, H. Stuckenschmidt, and J. Mendling. “Towards the automated annotation of process models”. In: International Conference on Advanced Information Systems Engineering. Springer. 2015, pp. 401–416. Reference [40].

2 Objectives and work programme

2.1 Anticipated total duration of the project

The anticipated duration of the project is two years (24 months).

2.2 Objectives

The objective of this project is to develop approaches that address the six key challenges raised in Section 1. As shown in Figure 2, the first three approaches will focus on *data transformation*, jointly allowing us to automatically turn a UI log into an event log, through event annotation, noise removal, and case identification. The latter three approaches will focus on process representation, turning an event log into an appropriate process model, through abstraction, labeling, and visualization.

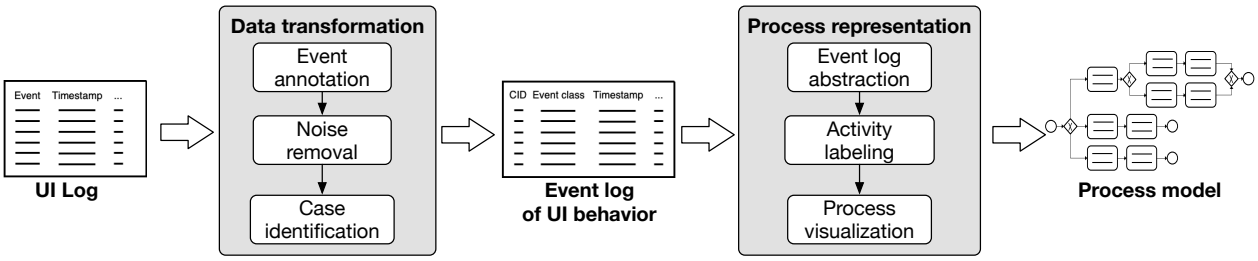

Figure 2: Overview of the proposed project

2.3 Work programme including proposed research methods

Package structure. In accordance with the outlined objectives, we divided the work programme into two work streams (WS1 and WS2), encompassing a total of six work packages (WP1 to WP6). Stream WS1, focusing on data transformation, will be led by the University of Mannheim and consists of WP1 to WP3. Stream WS2, targeting process representation, will be led by the Kühne Logistics University and consists of WP4 to WP6. We have designed the two work streams in such a way that they can be executed independently from each other. We achieved this by ensuring that WS2 can build on publicly available UI logs until the first results from WS1 are available. Therefore, both work streams can immediately start in parallel.

Research method. We will achieve our objectives through the design, implementation, and evaluation of novel process mining approaches. Their implementation will be conducted in Python, based on the *PM4Py* process mining library[1]. To evaluate the developed approaches, we will use publicly available real-world UI logs provided by the authors of [33][2]. Should we identify a need for evaluation data beyond the publicly available logs, we will use the publicly available *action logger* [35] tool to obtain additional UI logs. We will evaluate the approaches in WP1 to WP5 in terms of their *accuracy* and *efficiency*, e.g., with respect to manually established gold standards,

[1] https://pm4py.fit.fraunhofer.de/
[2] https://figshare.com/articles/dataset/UI_logs/12543587

whereas the *usefulness* of the visualization approach developed in WP6 shall also be assessed through a user study.

2.3.1 WP1: UI log enrichment (7 PM)

Although events in UI logs can be associated with a broad range of relevant attributes, they fundamentally lack the *event labels* that are required in process mining to indicate the meaning of events or to recognize equivalent ones (by using labels to define *event classes*). To illustrate this, consider the shortened version of our UI log in Figure 3. In an event log with proper labels, event 1 would carry a label such as "*Receive order [via e-mail]*", which would capture what exactly happened in the event (receiving an order) and, optionally, through which medium (e-mail). However, instead of providing such process-relevant information, the UI log only explicitly captures that the event was a "*click*" on a "*list*" in the application "*Outlook*".

ID	Timestamp	Event	Application	Element label	Element type	Element value	URL
1	08:35.2	click	Outlook	Customer X - O123	list	Please initiate an order ...	-
...	...	...	...	...	...	...	...
4	08:39.7	click	Chrome	Log in	button	-	https://www.salesforce.com/
5	08:40.0	change	Chrome	Password	text field	-	https://login.salesforce.com/
6	08:40.5	click	Chrome	Submit	button	-	https://login.salesforce.com/
7	08:52.6	click	Chrome	New Account	button	-	https://com.lightning.force.com/home
...	...	...	...	...	...	...	...
13	08:40.0	change	Chrome	Password	text field	-	https://www.facebook.com/
14	08:42.9	click	Chrome	Log in	button	-	https://www.facebook.com/
15	08:42.9	click	Chrome	Messenger	button	-	https://www.facebook.com/
...	...	...	...	...	...	...	...

Figure 3: Shortened version of UI log from Figure 1

Approach overview. To overcome this limitation in UI log data, this work package sets out to develop an approach for the semantic annotation of UI events. Specifically, our approach will enrich raw event data with dedicated attributes that capture *which action* was applied to *which object* and in *which context*. The former two aspects are common components of activity and event labels [37, 47], whereas the context (e.g., "*via e-mail*" or "*in Salesforce*") is incorporated because UI logs often span multiple applications, allowing us to distinguish events that appear to be similar, but occur in distinct parts of a process. Our approach will achieve this in two steps:

Step 1: Action and object identification. Our approach will use different strategies to identify the action and object to which an event applies. The proposed strategies differentiate between *button click*, *data entry*, and *data receipt* events, tailored to their structural differences:

Button click events. For events corresponding to a user clicking on a button, we will analyze the textual information associated with that button (typically the button's label, e.g., "*Submit*", "*New Account*"), or "*Upload files*". To do this, we will use a fine-tuned BERT [21] model to tag the tokens in a given text as relating to an action (`ACT`), a business object (`BO`), or a miscellaneous class (`X`), obtaining, e.g., *submit*\`ACT`, *new*\`BO` *account*\`BO`, and *upload*\`ACT` *files*\`BO`. Finally, to ensure that each event is associated with an action, if no `ACT` tag is assigned, we will associate the event with an action from a repository of standard actions (e.g., *create* or *modify*). To do this, we will use a recently developed classifier [52] that determines the most suitable action type based on the available textual information. For instance, "*new account*" best corresponds to a *create* action, which means that for event 7 our approach will return *action: "create"*, *object: "new account"*.

Data entry events. Data entry events correspond to events that set or change a value, e.g., in a free-text field, a drop-down menu, or a radio button. Our approach will use the information from the elemental label (e.g., "*Password*", "*order amount*", or "customer type"), if available, or otherwise the value that is set from a drop-down menu (e.g., "premium customer"). Given such a text, we will use the same tagging model as for button clicks to identify actions and objects. Afterwards, if no action was identified, we set the associated action to *enter*, *select*, or *set* depending on the element type, e.g., yielding *action: "enter"*, *object: "password"* or *action: "select"*, *object: "customer type"*.

Data receipt events. Finally, data receipt events correspond to any event that is about receiving or reading free-text information, such as opening an e-mail or reading a message. These events are the most challenging to annotate, given that an e-mail or message can relate to virtually any topic. To handle this diversity, our approach will first process the opened text using a recently proposed transformed-based approach for *e-mail intent mapping* [29], which will turn a larger message into a single phrase such as "*a meeting is being proposed*" or "*an order is being placed*". Next, we will use a fine-tuned BERT model to transform this phrase into a single object and adding a *receive* or *open* action, e.g., turning "*a meeting is being proposed*" into *action: "receive"*, *object: "meeting proposal"*.

Step 2: Context identification. Next, our approach identifies the specific application or sub application in which an event is performed, adding it as a *context* attribute to the event. For events performed by interacting with non-web applications, we directly derive the context based on the *application* attribute associated with an event. For browser-based events, instead, we set the context as the specific web application that an event relates to. To identify this, our approach will first apply string matching on the event's URL. For instance, after removing URL-specific prefixes and suffixes from event 4 in Figure 3, we obtain "*salesforce*", which can be easily verified as a business application using public resources. For events where this strategy does not deliver a conclusive result, we then turn to the event's broader context. For example, for event 7, the string matching strategy is unlikely to deduce that this event occurred in Salesforce. However, when taking the log's context into account, we can clearly identify a number of sub applications, such as Salesforce (from event 4) and Facebook (from event 13). Although the URL of event 7 is rather cryptic, comparing it against the two available options will clearly identify Salesforce as the most likely sub application. In this way, we enrich the UI log with the specific application in which each event occurred, enabling context-aware analysis of events in downstream tasks.

2.3.2 WP2: Noise removal (7 PM)

UI logs often contain events that do not relate to the business process under investigation, such as events related to private activities (e.g., checking Facebook) or to non-related business activities (e.g., filing a reimbursement form of a business trip). Given that process mining techniques assume that the events in a log relate to a single process, these irrelevant events, also referred to as *noise*, need to be identified and removed from a UI log.

While various noise detection techniques already exist (cf., Section 1.1.1), these inherently approach noise detection from a different angle. Particularly, they aim to detect process behavior that stands out in terms of frequency (such as rare occurrence of an order being accepted before it is created), i.e., they try to *detect events that are behavioral anomalies*, e.g., caused by recording errors. When dealing with UI logs, we need to *detect events that do not relate to the process at hand*.

Approach overview. To tackle this task, we propose to develop a semantic noise detection approach tailored to the specifics of UI logs. Our approach will capture noise detection in the form of a classification task, aiming to classify events as *process relevant* or not, based on the textual contents of their attributes. We will develop and compare several classifiers, to determine the best way to operationalize our approach. Specifically, we will test a traditional machine learning model using feature embeddings and standard classifiers, as well as a deep learning model using transformers.

Model 1: Embedding-based classification. We will assess the potential of traditional machine learning by first transforming the textual input data associated with an event into a high-dimensional feature vector using *embedding* techniques. In this manner, we encode both general event information in features, as well as process-specific information extracted in the previous step, i.e., the event's action, business object, and application. We will obtain (and compare) embeddings using established methods, such as through GloVe embeddings [49] and SentenceBERT [55].
Given such embeddings, noise identification can be approached using standard classifiers, for which we expect support vector machines (SVMs) to be well-suited. We will test and compare single-class and two-class classification strategies for this. Single-class classification involves training a classifier

that recognizes which events are related to a specific process by assuming that the majority of events in a UI log are related to that. While such an approach thus would not require any user input, the classification accuracy may be improved in a two-class setting, where users explicitly label some events as process relevant and irrelevant (i.e., ground-truth labels).

Model 2: Transformer-based classification. Aside from traditional machine learning techniques, we will also test state-of-the-art transformer-based models to classify events as noisy or not. We will achieve this by fine-tuning a transformer, such as BERT [21], directly on the text-classification task at hand. Again, we will test both single-class and two-class scenarios here.
Although the potential of such transformers is well-known, it will be interesting to see how well they perform in comparison to embedding-based models in situations with relatively small amounts of training data, which is important in the context of UI log analysis.

Depending on the outcome of our experiments, our final approach will either consist of one specific classification technique, or will provide recommendations on which technique to use in a given situation, e.g., depending on the size of the UI logs and the availability of ground-truth labels.

2.3.3 WP3: Case identification (10 PM)

Once irrelevant events have been filtered out from a UI log, we next need to recognize which events belong to the same process instance, e.g., to the same customer order or service ticket, a task referred to as *case identification*.

Existing techniques that address this task in general process mining settings primarily base the identification on co-occurrence statistics, which reveal behavioral regulations in an event log (cf., Section 1.1.1). Such techniques work well for relatively structured settings, but they are not sufficient when dealing with event data stemming from more flexible environments, in which the execution of several cases overlap. This is, e.g., seen in the example of Figure 1, where the first three events each start a new process instance, by initiating three different orders in a batch-like manner. This, therefore, calls for a new case-identification approach, tailored to the specifics of the UI logs.

Approach overview. This work package sets out to develop such a new approach for case identification in UI logs, which we achieve by tackling it as a *matching problem*. Following established practice in this regard [27], our approach will start by applying *first-line matchers* on the events in a UI log, which compute similarity scores between pairs of events, and then apply a *second-line matcher* that uses these computed similarity scores to find an optimal overall alignment among the events in a UI log. The resulting alignment will capture which events belong to each other, i.e., which are determined to be part of the same case.

Step 1: First-line matching. We will establish different first-line matchers, each dedicated to quantifying the similarity between two events from a particular perspective. Particularly, we will include matchers that consider: 1) *Behavioral relatedness*, which can recognize events that are observed to commonly co-occur or follow each other, similar to techniques used to capture behavioral regularities in event logs [23, 26] and those proposed in the context of task mining on structured UI logs [33, 63]. 2) *Semantic relatedness* between events, e.g., to recognize that an order cannot be *accepted before it is created*, which means that events appearing in that order cannot belong to the same case, for which we will exploit semantic regularities captured in our earlier work [5]. 3) *Shared identifiers*, which will look for mentions of specific identifiers (e.g., “Customer X” or “O123”) that can be helpful to recognize events that relate to the same object, when such information is available.

Step 2: Second-line matching. Having obtained multiple sets of similarity scores over the event-pairs in a UI log, our approach will then use these scores, in combination with relevant cardinality constraints (e.g., each event belongs to exactly or at most one case) to establish an optimization problem. We will represent and solve these problems using Markov logic networks [45, 56], which we have successfully used to solve other matching problems before [40]. In this manner, our approach will obtain an event grouping that maximizes the different kinds of similarity (according to learned parameter weights) in light of the applied constraints, thus yielding groups of events that

are determined to belong to the same case.

In this manner, the transformation from a UI log into an event log suitable for process mining is complete.

2.3.4 WP4: Event abstraction (6 PM)

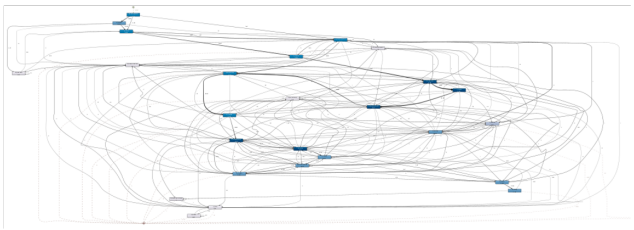

Figure 4: A so-called spaghetti process model

Even after transformation into a proper event log, the logs stemming from UI recordings comprise low-level event data, in which each event corresponds to a small action performed by a user. This fine-granular nature makes the data unsuitable for meaningful process analysis since the application of discovery techniques yields so-called *spaghetti models*, as e.g., depicted in Figure 4. To tackle this issue, various techniques for grouping low-level events into high-level activities have been defined (see Section 1.1.1). However, these approaches do not allow to define *what properties* the abstracted log shall satisfy. Hence, it is hard to ensure that the resulting abstraction is appropriate for a specific analysis goal.

Approach overview. To overcome this limitation, we will develop an approach for *constraint-driven event abstraction*. Based on 1) a number of distance functions that quantify the similarity between events and 2) a set of (user-defined) constraints, our approach identifies an optimal log abstraction using *constraint-based clustering*. That is, it groups together low-level event classes into high-level activities such that the abstracted log is meaningful for downstream analysis. The outcome of our approach, therefore, is an event log with respective high-level activities that provides clear insights into the high-level process behavior contained in the UI log.

Step 1: Distance computation. First, we need to determine which low-level event classes are similar to each other. Note that this is conceptually comparable to step 1 from WP3, where we identify which events belong to the same case. While some ideas from WP3 can be applied here as well, there is a need to identify groups on lower level of granularity and with a particular focus on cross-case behavior. To this end, we define three different *distance functions*: 1) We use a temporal distance function d_{tmp} to quantify how much time passes typically between instances of two event classes. Intuitively, event classes belonging to the same group should also commonly occur shortly after each other. 2) We will use a behavioral distance function d_{bhv} to quantify the correlation of event classes from a control-flow perspective. As explained in WP3, there are different ways to detect and quantify this and we will build on existing work from behavioral analysis [23, 26] and task mining [33, 63] to operationalize this. 3) We use a semantic distance function d_{sem} to quantify to what extent two event classes are concerned with the same task. To capture this, we will leverage available information about the event context, such as the application in which the event is executed or the data that is entered or processed during the event execution.

Step 2: Constraint generation. Second, we need to determine which constraints need to hold in order to obtain a meaningful event abstraction. An obvious and simple solution is to ask the user for input on that matter. However, while such user-defined constraints can lead to meaningful event abstraction, it may not always be obvious to users which constraints they shall request. Therefore, we will develop an algorithm that suggests potential constraints on the event data. To this end, we will identify which attribute values lead to a clear separation between different groups of event classes from a control-flow or temporal perspective, e.g., by considering aspects such as cohesion and coupling over the abstracted log [67]. A prime challenge here will be to obtain suggestions in a computationally efficient manner, given the high dimensionality of the task. We aim to deal with this by reducing the problem size through decomposition of the event data, e.g., on the directly-follows graph, and by considering the characteristics of data attributes in the log, e.g., through data profiling [48], allowing us to discard unpromising attributes in advance.

Step 3: Constraint-based Clustering. Third, we use the output from the distance computations from step 1 and the set of constraints from step 2 to determine the final clusters of event classes, which can then be used to abstract the cases in the event log. Intuitively, the distances determined via the three distance functions should be minimized, while still meeting the imposed constraints. To implement this, we will build on an existing approach for clustering with instance-level constraints [68].

2.3.5 WP5: Activity labeling (8 PM)

The value of a discovered process model highly depends on the quality of its activity labels, since these labels form the basis of what a human can understand about the process [47]. Recognizing this, the importance of clear and informative labels in process models has led to a large body of literature concerned with labeling guidelines [41, 46] and their automatic enforcement [15, 39]. These best practices indicate that process model activity labels should contain an action provided as an imperative verb (e.g. "*create*" or "*send*") and at least one business object provided as a noun (e.g. "*order*" or "*e-mail*"). Additional information can be provided at the end of the label if required. Following these rules results in labels such as "*Create order*" or "*Send e-mail to customer*".

The challenge in the context of this work package is to automatically generate meaningful labels for each group of UI events that has been identified in the event-abstraction step (WP4). To illustrate this, consider the events 4, 5, and 6 from the UI log in Figure 1, which, respectively, refer to the pressing of a log-in button, entering a password, and pressing the submit button. Recognizing the joint role of these events in the process, a proper label for the group could be "*Log into Salesforce*", which summarizes both the outcome (logging in) and the context (Salesforce) of the event group. However, automatically generating such *higher-level labels* is complex, since it requires to infer an overarching activity from the actions, business objects, and application contexts of multiple low-level events. Furthermore, how to derive such an overarching label depends on the nature of the events and, thus, differs per situation.

Approach overview. In light of these challenges, we propose a novel activity labeling approach. In a first step, our approach uses three different strategies to generate different *label candidates*. The rationale behind having different strategies is that a considered group of UI events can be described in different ways. In a second step, our approach then selects the most suitable labeling strategy based on behavioral and semantic analysis of the considered UI events. The outcome of our approach, therefore, is a higher-level activity label for a given set of UI events. These labels are then used as additional annotations for the log stemming from WP4, resulting in an abstracted and meaningful event log.

Step 1: Label candidate generation. To generate higher-level labels, we jointly analyze the behavioral and semantic perspectives of grouped events. Building on observations from earlier work on process model name generation [42], we define three specific strategies to generate label candidates:

Outcome-oriented labeling. This strategy builds on the observation that a *result* of a set of UI events can be used to describe the previous steps. As an example, consider the "*Log into Salesforce*" label for events 4, 5, and 6. In line with earlier work, we will derive such outcome-oriented labels by using the low-level labels from the first or the last event of a considered UI event group [42].

Decision-based labeling. This strategy will exploit the fact that many processes are driven by key decisions, such as rejecting or accepting an offer from a customer. If a considered group of events encompasses such a decision point, we extract the main object that is affected by the decision and capture it in a high-level label, such as "*Decide about customer offer*".

Holonym-based labeling. This final strategy considers a setting where low-level events jointly contribute to a higher-level activity. In linguistics, a *holonym* is a term that has a part-of relation with a number of *meronyms*, e.g., a "*finger*" is a meronym of the holonym "*hand*". By applying the notion of holonymy to events, we aim to recognize actions or business objects that are considered to be meronyms, allowing us to detect the holonym describing the higher-level activity. As an example,

consider events 11, 12, and 13 from Figure 1. While these events are essentially copy and paste operations, they all contribute to a higher-level event that could be described as "*Create customer account in Salesforce*". To detect such holonym relations, we will leverage techniques from ontology learning [11, 73]. In this way, we overcome the limitations of previous work [42] which built on the lexical database WordNet and, hence, was limited with respect to the scope.

Step 2: Strategy selection. The three strategies result in different labels for the same group of events. Therefore, the second step is to select the most appropriate labeling strategy. To this end, we again build on a behavioral and semantic analysis of the low-level events involved and implement respective selection rules. We pick outcome-oriented strategy if the group of events represents a synchronization point in a process, such as a *log-in step*, which is succeeded by various other paths. We pick the decision-based strategy if the behavioral relations indicate a choice in the process as signaled by mutually exclusive events or when low-level actions have clearly opposite meanings, e.g., *reject* versus *accept*. Finally, we pick the holonym-based strategy when holonym relations can actually be identified for the given low-level events. In unclear cases, we stick to the first strategy as this will always deliver a result.

2.3.6 WP6: Visualization (10 PM)

The final step is concerned with the visualization of the process captured in the UI log. In traditional process discovery, this is achieved by generating respective process models. Depending on the employed discovery algorithm, the resulting process model could be a Petri net [66], a BPMN model [20], or a process tree [30]. While each algorithm and output representation comes with advantages and disadvantages, there is no specific discovery technique available tailored to the kind of UI-based event logs resulting from WP1 to WP5. The specific challenge is that the original level of granularity of the UI log is not appropriate for visualization since the number of events is too high. Yet, the user might be interested in getting insights into this level to fully understand how the process is executed. What is required is a dedicated *visualization approach* that allows the user to adapt the process visualization in a flexible manner with respect to 1) the abstraction level and 2) the general scope. In the context of data warehousing and OLAP [18], the former corresponds to *drilling down* and *rolling up*, while the latter corresponds to *slicing*.

Approach overview. In line with the requirements outlined above, we will propose a novel interactive visualization approach that allows the user to adapt both the abstraction level and the scope of the discovered model. To adapt the abstraction level, we introduce a *slider-based* mechanism that allows the user to drill down or roll up. To adapt the scope, we introduce a *semantic scoping* mechanism which provides the user with the possibility to limit the scope with respect to certain objects or applications. The outcome, therefore, is a novel discovery and representation approach that can visualize UI logs in a flexible fashion. As such, it completes our overall pipeline for semantic process discovery from UI logs.

Slider-based abstraction. The general idea of adapting the level of abstraction using a slider approach is present in several commercial process mining tools, such as Celonis[3] and Disco[4]. However, they implement abstraction by removing arcs or nodes based on their frequency. Our idea is to merge events based on the abstraction mechanism introduced in WP4 and use the approach from WP5 to present meaningful labels to the user. In case the user is interested in more details, they can drill down on individual events or groups thereof and explore the process on several levels at the same time. To this end, we will build on the ideas on multi-level discovery from [31] and adapt them to the specific scenario of slider-based abstraction.

Semantic scoping. Besides adapting the level of abstraction, the user might also want to modify the general scope of the discovered model. It is well imaginable that, for instance, there are specific applications or business objects in the process the user wishes to investigate further or, in the opposite case, remove them from the visualization. We, therefore, introduce a scoping mechanism

[3] www.celonis.com

[4] https://fluxicon.com/disco/

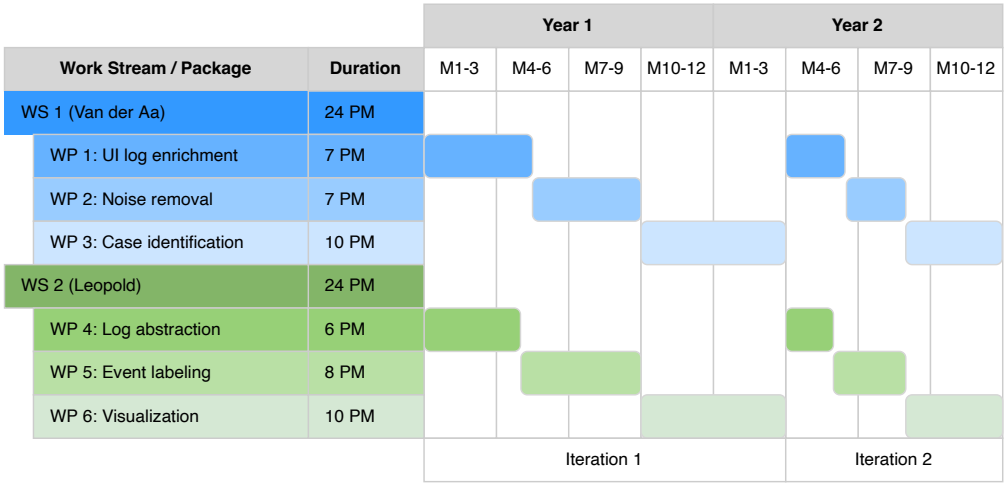

Figure 5: Work plan

that allows the user to select applications and business objects that should be included or removed. To accomplish this, we employ a fine-tuned version of the open-source language model BLOOM [59]. More specifically, we fine-tune BLOOM with a number of UI log traces such that BLOOM can learn about both semantic and behavioral aspects of typical UI events. Using the fine-tuned verion of BLOOM, we then determine which events should be included or removed from the presented model. Note that this is not trivial since the links between applications and business objects are complex and simply removing all events that do not mention the selected term "*invoice*" is certainly too simplistic. Hence, we use BLOOM to determine which other events are semantically related to selected business object(s) and respectively include them, while simultaneously ensuring the correctness of the scoped process from a behavioral perspective.

User evaluation. The two mechanisms introduced above are novel and require interaction with users. Therefore, we plan to complement the technical evaluation of our approach with a user study. Specifically, we aim to investigate the effectiveness (in terms of the ability of obtaining insights) and efficiency (in terms of how fast user can obtain insights). For this, we will build on our experience with user studies in the context of process understandability [50] and process modeling support [1].

2.3.7 Overview of the work plan

As explained above, the project consists of two work streams and six work packages. Figure 5 shows a simplified work plan that mostly abstracts from parallel and overlapping work within each work stream. It is important to highlight that the transitions between packages will not be as strict as depicted. We are aware of the various interdependencies that exist among the packages and will take them into account appropriately. The abstract view on the work plan, shown in Figure 5, highlights our general idea of having two main iterations:

- The first iteration ends after 15 months. At the end of this iteration, there will be a first implemented prototype available for the proposed end-to-end pipeline. The individual approaches have been evaluated both independently and as a whole. The main outcome from this iteration, therefore, are insights into the strengths and weaknesses of our work, which allows us to determine the required improvements and adaptations for the second iteration.
- The second iteration is slightly shorter and will be mainly used to address identified weaknesses and improve the quality and broaden the scope of the developed approaches.

Note that these two iterations help us to account for the aforementioned interdependencies between the work streams. In the first iteration, WS2 will build on manually refined, publicly available UI

logs[5], while in the second iteration WS2 can use UI logs that stem from the approaches developed in WS1.

3 Bibliography concerning the state of the art, the research objectives, and the work

[1] H. van der Aa, K. J. Balder, F. M. Maggi, and A. Nolte. “Say it in your own words: Defining declarative process models using speech recognition”. In: *BPM (Forum)*. Springer. 2020, pp. 51–67.

[2] H. van der Aa, J. Carmona Vargas, H. Leopold, J. Mendling, and L. Padró. “Challenges and opportunities of applying natural language processing in business process management”. In: *COLING*. Association for Computational Linguistics. 2018, pp. 2791–2801.

[3] H. van der Aa, C. Di Ciccio, H. Leopold, and H. A. Reijers. “Extracting declarative process models from natural language”. In: *CAiSE*. Springer. 2019, pp. 365–382.

[4] H. van der Aa, H. Leopold, and H. A. Reijers. “Checking process compliance against natural language specifications using behavioral spaces”. In: *Information Systems* 78 (2018), pp. 83–95.

[5] H. van der Aa, A. Rebmann, and H. Leopold. “Natural language-based detection of semantic execution anomalies in event logs”. In: *Information Systems* 102 (2021), p. 101824.

[6] W. van der Aalst. *Process Mining: Data science in action*. Springer, 2016, pp. 3–23.

[7] L. Abb and J.-R. Rehse. “A reference data model for process-related user interaction logs”. In: *International Conference on Business Process Management*. Springer. 2022, pp. 57–74.

[8] A. Adriansyah, B. F. van Dongen, and W. van der Aalst. “Conformance checking using cost-based fitness analysis”. In: *EDOC*. IEEE. 2011, pp. 55–64.

[9] S. Agostinelli, A. Marrella, L. Abb, and J.-R. Rehse. “Mastering robotic process automation with process mining”. In: *Business Process Management: 20th International Conference, BPM 2022, Münster, Germany, September 11–16, 2022, Proceedings*. Springer. 2022, pp. 47–53.

[10] S. Agostinelli, A. Marrella, and M. Mecella. “11 Automated segmentation of user interface logs”. In: *Robotic Process Automation*. De Gruyter Oldenbourg, 2021, pp. 201–222.

[11] F. N. Al-Aswadi, H. Y. Chan, and K. H. Gan. “Automatic ontology construction from text: a review from shallow to deep learning trend”. In: *Artificial Intelligence Review* 53.6 (2020), pp. 3901–3928.

[12] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo. “Automated discovery of process models from event logs: Review and benchmark”. In: *IEEE transactions on knowledge and data engineering* 31.4 (2018), pp. 686–705.

[13] T. Baier, J. Mendling, and M. Weske. “Bridging abstraction layers in process mining”. In: *Information Systems* 46 (2014), pp. 123–139.

[14] D. Bayomie, C. Di Ciccio, M. La Rosa, and J. Mendling. “A probabilistic approach to event-case correlation for process mining”. In: *ER*. Springer. 2019, pp. 136–152.

[15] J. Becker, P. Delfmann, S. Herwig, L. Lis, and A. Stein. “Towards increased comparability of conceptual models-enforcing naming conventions through domain thesauri and linguistic grammars”. In: (2009).

[16] A. Bosco, A. Augusto, M. Dumas, M. La Rosa, and G. Fortino. “Discovering automatable routines from user interaction logs”. In: *BPM*. Springer. 2019, pp. 144–162.

[17] N. Busany, H. v. d. Aa, A. Senderovich, A. Gal, and M. Weidlich. “Interval-based queries over lossy iot event streams”. In: *ACM Transactions on Data Science* 1.4 (2020), pp. 1–27.

[18] S. Chaudhuri and U. Dayal. “An overview of data warehousing and OLAP technology”. In: *ACM Sigmod record* 26.1 (1997), pp. 65–74.

[19] H.-J. Cheng and A. Kumar. “Process mining on noisy logs — Can log sanitization help to improve performance?” In: *Decision Support Systems* 79 (2015), pp. 138–149. ISSN: 0167-9236.

[20] R. Conforti, M. Dumas, L. Garcıa-Banuelos, and M. La Rosa. “BPMN Miner: Automated discovery of BPMN process models with hierarchical structure”. In: *Information Systems* 56 (2016), pp. 284–303.

[21] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In: *NAACL 2019* 1.Mlm (2019), pp. 4171–4186.

[22] C. Di Francescomarino, C. Ghidini, F. M. Maggi, and F. Milani. “Predictive process monitoring methods: Which one suits me best?” In: *BPM*. Springer. 2018, pp. 462–479.

[23] K. Diba, K. Batoulis, M. Weidlich, and M. Weske. “Extraction, correlation, and abstraction of event data for process mining”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.3 (2020), e1346.

[24] M. L. van Eck, N. Sidorova, and W. M. van der Aalst. “Enabling process mining on sensor data from smart products”. In: *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*. IEEE. 2016, pp. 1–12.

[25] S. A. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich. “PRIPEL: Privacy-preserving event log publishing including contextual information”. In: *BPM*. Springer. 2020, pp. 111–128.

[26] D. R. Ferreira and D. Gillblad. “Discovering process models from unlabelled event logs”. In: *BPM*. Springer. 2009, pp. 143–158.

[5] https://figshare.com/articles/dataset/UI_logs/12543587/4

[27] A. Gal. “Uncertain schema matching”. In: *Synthesis Lectures on Data Management* 3.1 (2011), pp. 1–97.

[28] C. W. Günther and W. M. Van Der Aalst. “Fuzzy mining–adaptive process simplification based on multi-perspective metrics”. In: *BPM*. Springer. 2007, pp. 328–343.

[29] F. Khandaker, A. Senderovich, E. Yu, S. Carbajales, and A. Chan. “Transformer Models for Activity Mining in Knowledge-Intensive Processes”. In: *AI4BPM*. Springer. 2022.

[30] S. J. Leemans, D. Fahland, and W. van der Aalst. “Discovering block-structured process models from event logs containing infrequent behaviour”. In: *BPM*. Springer. 2013, pp. 66–78.

[31] S. J. Leemans, K. Goel, and S. J. van Zelst. “Using multi-level information in hierarchical process mining: Balancing behavioural quality and model complexity”. In: *2020 2nd International Conference on Process Mining (ICPM)*. IEEE. 2020, pp. 137–144.

[32] V. Leno, A. Augusto, M. Dumas, M. La Rosa, F. M. Maggi, and A. Polyvyanyy. “Discovering data transfer routines from user interaction logs”. In: *Information Systems* 107 (2022), p. 101916.

[33] V. Leno, A. Augusto, M. Dumas, M. La Rosa, F. M. Maggi, and A. Polyvyanyy. “Identifying candidate routines for robotic process automation from unsegmented UI logs”. In: *2020 2nd International Conference on Process Mining (ICPM)*. IEEE. 2020, pp. 153–160.

[34] V. Leno, A. Polyvyanyy, M. Dumas, M. La Rosa, and F. Maggi. “Robotic process mining: vision and challenges”. In: *Business & Information Systems Engineering* 63.3 (2021), pp. 301–314.

[35] V. Leno, A. Polyvyanyy, M. La Rosa, M. Dumas, and F. M. Maggi. “Action logger: Enabling process mining for robotic process automation”. In: CEUR Workshop Proceedings. 2019.

[36] M. de Leoni and S. Dündar. “Event-log abstraction using batch session identification and clustering”. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020, pp. 36–44.

[37] H. Leopold, H. van der Aa, J. Offenberg, and H. A. Reijers. “Using Hidden Markov Models for the accurate linguistic analysis of process model activity labels”. In: *Information Systems* 83 (2019), pp. 30–39.

[38] H. Leopold, H. van der Aa, F. Pittke, M. Raffel, J. Mendling, and H. A. Reijers. “Searching textual and model-based process descriptions based on a unified data format”. In: *Software & Systems Modeling* 18.2 (2019), pp. 1179–1194.

[39] H. Leopold, R.-H. Eid-Sabbagh, J. Mendling, L. G. Azevedo, and F. A. Baião. “Detection of naming convention violations in process models for different languages”. In: *Decision Support Systems* 56 (2013), pp. 310–325.

[40] H. Leopold, C. Meilicke, M. Fellmann, F. Pittke, H. Stuckenschmidt, and J. Mendling. “Towards the automated annotation of process models”. In: *CAiSE*. Springer. 2015, pp. 401–416.

[41] H. Leopold, J. Mendling, and O. Günther. “Learning from quality issues of BPMN models from industry”. In: *IEEE software* 33.4 (2015), pp. 26–33.

[42] H. Leopold, J. Mendling, H. A. Reijers, and M. La Rosa. “Simplifying process model abstraction: Techniques for generating model names”. In: *Information Systems* 39 (2014), pp. 134–151.

[43] H. Leopold, M. Niepert, M. Weidlich, J. Mendling, R. Dijkman, and H. Stuckenschmidt. “Probabilistic optimization of semantic process model matching”. In: *BPM*. Springer. 2012, pp. 319–334.

[44] H. Leopold, S. Smirnov, and J. Mendling. “On the refactoring of activity labels in business process models”. In: *Information Systems* 37.5 (2012), pp. 443–459.

[45] D. Lowd and P. Domingos. “Efficient weight learning for Markov logic networks”. In: *PKDD*. Springer. 2007, pp. 200–211.

[46] J. Mendling, H. A. Reijers, and W. M. van der Aalst. “Seven process modeling guidelines (7PMG)”. In: *Information and Software Technology* 52.2 (2010), pp. 127–136.

[47] J. Mendling, H. A. Reijers, and J. Recker. “Activity labeling in process modeling: Empirical insights and recommendations”. In: *Information Systems* 35.4 (2010), pp. 467–482.

[48] T. Papenbrock, T. Bergmann, M. Finke, J. Zwiener, and F. Naumann. “Data profiling with metanome”. In: *Proceedings of the VLDB Endowment* 8.12 (2015), pp. 1860–1863.

[49] J. Pennington, R. Socher, and C. D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543.

[50] F. Pittke, H. Leopold, and J. Mendling. “Automatic detection and resolution of lexical ambiguity in process models”. In: *IEEE Transactions on Software Engineering* 41.6 (2015), pp. 526–544.

[51] A. Polyvyanyy, W. M. V. D. Aalst, A. H. T. Hofstede, and M. T. Wynn. “Impact-driven process model repair”. In: *ACM Transactions on Software Engineering and Methodology* 25.4 (2016), pp. 1–60.

[52] A. Rebmann and H. van der Aa. “Enabling semantics-aware process mining through the automatic annotation of event logs”. In: *Inf. Syst.* 110 (2022), p. 102111.

[53] A. Rebmann and H. van der Aa. “Extracting semantic process information from the natural language in event logs”. In: *CAiSE*. Springer. 2021, pp. 57–74.

[54] A. Rebmann, M. Weidlich, and H. van der Aa. “GECCO: Constraint-driven Abstraction of Low-level Event Logs”. In: *International Conference on Data Engineering*. 2022.

[55] N. Reimers and I. Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *EMNLP*. ACL, Nov. 2019.

[56] M. Richardson and P. Domingos. “Markov logic networks”. In: *Machine learning* 62 (2006), pp. 107–136.

[57] A. Rozinat and W. M. Van der Aalst. “Conformance checking of processes based on monitoring real behavior”. In: *Information Systems* 33.1 (2008), pp. 64–95.

[58] J. Sànchez-Ferreres, H. van der Aa, J. Carmona, and L. Padró. “Aligning textual and model-based process descriptions”. In: *Data & Knowledge Engineering* 118 (2018), pp. 25–40.
[59] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. “Bloom: A 176b-parameter open-access multilingual language model”. In: *arXiv preprint arXiv:2211.05100* (2022).
[60] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. Leemans, C. Ouyang, A. H. ter Hofstede, I. van de Weerd, M. T. Wynn, and H. A. Reijers. “Robotic Process Automation: Contemporary themes and challenges”. In: *Computers in Industry* 115 (2020), p. 103162. ISSN: 0166-3615.
[61] N. Tax, N. Sidorova, and W. M. van der Aalst. “Discovering more precise process models from event logs by filtering out chaotic activities”. In: *arXiv preprint arXiv:1711.01287* (2017).
[62] Y. Urabe, S. Yagi, K. Tsuchikawa, and T. Masuda. “Visualizing User Action Data to Discover Business Process”. In: *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE. 2019, pp. 1–4.
[63] Y. Urabe, S. Yagi, K. Tsuchikawa, and H. Oishi. “Task Clustering Method Using User Interaction Logs to Plan RPA Introduction”. In: *Business Process Management*. 2021.
[64] H. Van der Aa, A. Artikis, and M. Weidlich. “Complex Event Processing Methods for Process Querying”. In: *Process Querying*. 2021 (forthcoming).
[65] H. Van der Aa, H. Leopold, and M. Weidlich. “Partial order resolution of event logs for process conformance checking”. In: *Decision Support Systems* 136 (2020), p. 113347.
[66] W. Van der Aalst, T. Weijters, and L. Maruster. “Workflow mining: Discovering process models from event logs”. In: *IEEE transactions on knowledge and data engineering* 16.9 (2004), pp. 1128–1142.
[67] I. Vanderfeesten, J. Cardoso, J. Mendling, H. A. Reijers, and W. Van der Aalst. “Quality metrics for business process models”. In: *BPM and Workflow handbook* 144 (2007), pp. 179–190.
[68] K. Wagstaff and C. Cardie. “Clustering with instance-level constraints”. In: *AAAI/IAAI* 1097 (2000), pp. 577–584.
[69] M. Weidlich, T. Sagi, H. Leopold, A. Gal, and J. Mendling. “Predicting the quality of process model matching”. In: *BPM*. Springer, 2013, pp. 203–210.
[70] A. Weijters and J. Ribeiro. “Flexible heuristics miner (FHM)”. In: *2011 IEEE symposium on computational intelligence and data mining (CIDM)*. IEEE. 2011, pp. 310–317.
[71] A. J. Weijters and W. M. Van der Aalst. “Rediscovering workflow models from event-based data using little thumb”. In: *Integrated Computer-Aided Engineering* 10.2 (2003), pp. 151–162.
[72] K. Winter, H. van der Aa, S. Rinderle-Ma, and M. Weidlich. “Assessing the compliance of business process models with regulatory documents”. In: *International Conference on Conceptual Modeling*. Springer. 2020, pp. 189–203.
[73] W. Wong, W. Liu, and M. Bennamoun. “Ontology learning from text: A look back and into the future”. In: *ACM Computing Surveys (CSUR)* 44.4 (2012), pp. 1–36.
[74] S. J. van Zelst, B. F. van Dongen, and W. M. van der Aalst. “Avoiding over-fitting in ILP-based process discovery”. In: *BPM*. Springer. 2016, pp. 163–171.
[75] S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider. “Event abstraction in process mining: literature review and taxonomy”. In: *Granular Computing* (2020), pp. 1–18.
[76] B. Zhao, H. van der Aa, T. T. Nguyen, Q. V. H. Nguyen, and M. Weidlich. “EIRES: Efficient Integration of Remote Data in Event Stream Processing”. In: *Proceedings of the 2021 International Conference on Management of Data*. 2021, pp. 2128–2141.

4 Relevance of sex, gender and/or diversity

Neither sex, gender, nor diversity will play a role in the context of this research.

5 Supplementary information on the research context

5.1 Ethical and/or legal aspects of the project

Ethical and legal considerations are not applicable to the proposed project since we will not conduct experiments on humans, animals, or genetic resources, nor will we produce results that can be intentionally misused to cause harm.

We will conduct user studies in WP6 of the project to evaluate the the effectiveness and efficiency of the discovered process representation. Such user studies are highly common in the context of computer science research. Participation in these studies will be fully voluntarily, participants will have the possibility to opt out at any moment, and we will not collect any personal data as defined

by the Ethics Committee of the University of Mannheim[6]. Given that we will not collect any personal data, an ethics vote is not required according to the committee's guidelines.

5.2 Data handling

As explained in Section 2.3, we may create new synthetic UI log data sets in order to evaluate our approaches. Such data sets are not only relevant for this project, but also for other researchers in the area of process mining. In fact, the large interest in event logs has led to the creation of a central event log repository[7], which is hosted and managed by the IEEE Task Force on Process Mining. As members of the IEEE Task Force on Process Mining, we therefore plan to make our data sets publicly available via this repository.

5.3 Other information

All relevant information is discussed above.

6 People/collaborations/funding

6.1 Employment status information

Prof. Dr. Henrik Leopold, Associate Professor at the Kühne Logistics University, permanent position. Prof. Dr. Han van der Aa, Junior Professor (W1) at the University of Mannheim with a contract until March 2026.

6.2 First-time proposal data

Not applicable.

6.3 Composition of the project group

The project group is composed of Prof. Henrik Leopold and Prof. Han van der Aa:

Prof. Dr. Henrik Leopold - Henrik Leopold is a tenured Associate Professor at the Kühne Logistics University (KLU) and senior researcher at the Hasso Plattner Institute (HPI) at the Digital Engineering Faculty, University of Potsdam. Before joining KLU/HPI in February 2019, he held positions as an Assistant Professor at the Vrije Universiteit Amsterdam (February 2015 – January 2019) and WU Vienna (April 2014 – January 2015) as well as a postdoctoral research fellow at the Humboldt University of Berlin (July 2013 – March 2014). In July 2013, he obtained a PhD degree (Dr. rer. pol.) in Information Systems from the Humboldt University of Berlin. For his thesis he received the TARGION Dissertation Award 2014 for the best doctoral thesis in the field of Information Management and the runner-up of the McKinsey Business Technology Award 2013. Henrik Leopold's research is concerned with leveraging technology from the field of artificial intelligence to develop automated techniques for process analysis and process mining. The results of his research have been published in over 100 publications in books, book chapters, journals, conferences, workshops, and reports. Among others, his research has been published in the journals IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Software

[6] https://www.uni-mannheim.de/en/about/organization/bodies-and-committees/committees-and-councils/ethics-committee

[7] https://www.tf-pm.org/resources/logs

Engineering, ACM Transactions on Management Information Systems, Decision Support Systems, and Information Systems.

Prof. Dr. Han van der Aa - Han van der Aa is a Junior Professor (W1) in the School of Business Informatics at the University of Mannheim, where he heads the research group on process analytics since April 2020. Before that, he was an Alexander von Humboldt Fellow, working as a postdoctoral researcher in the Department of Computer Science at the Humboldt-Universität zu Berlin (May 2018 - March 2020). He obtained a PhD in computer science from the Vrije Universiteit Amsterdam in January 2018. His research interests include business process modeling, process mining, natural language processing, and complex event processing. The results of his research have so far resulted in close 50 publications, including articles in renowned international journals such as IEEE TKDE, DSS, and Information Systems, as well as at the CAISE, BPM, SIGMOD, and COLING conferences. He is a PC member of established conferences such as BPM, ICPM, and DEBS.

The applicants have a history of successful collaboration. While they share a similar background with respect to the application of natural language processing in process analysis and mining, they have complementary skills and knowledge, as described in the context of their preliminary work in Section 1.1.2.

6.4 Researchers in Germany with whom you have agreed to cooperate on this project

We agreed to cooperate on this project with three researchers from Germany: Prof. Dr. Stefanie Rinderle-Ma, Prof. Dr. Matthias Weidlich, and Prof. Dr. Simone Ponzetto.

Prof. Dr. Stefanie Rinderle-Ma: Stefanie Rinderle-Ma is a full professor for Information Systems and Business Process Management at the Department of Informatics, Technical University of Munich, Germany. Before Stefanie worked as full professor at the University of Vienna, Austria, where she led the Research Group on Workflow Systems and Technology. Stefanie‘s research interests include flexible and distributed process technology, production intelligence, process mining, and digitalized compliance management.

Prof. Dr. Matthias Weidlich: Matthias Weidlich is a full professor at the Department of Computer Science at Humboldt- Universiät zu Berlin and has been an Emmy Noether Research group leader at the same institute. Before that, he held positions at the Department of Computing at Imperial College London and at the Technion - Israel Institute of Technology. He holds a PhD from the Hasso Plattner Institute (HPI), University of Potsdam. His research focuses on process-oriented and event-driven systems and his results appear regularly in the premier conferences (VLDB, SIGMOD) and journals (TKDE, Inf. Sys., VLDBJ) in the field.

Prof. Dr. Simone Ponzetto: Simone Ponzetto is a professor of Information Systems at the University of Mannheim, where he leads the Natural Language Processing (NLP) and Information Retrieval group. His main research interests lie in the areas of knowledge acquisition, text understanding, and the application of NLP methods for research in the (digital) humanities and (computational) social sciences. Simone regularly serves as area chair and program committee member of *ACL and (IJC/AA)AI conferences and is an editorial board member of the Artificial Intelligence Journal and Journal of Natural Language Engineering. He is (co-)author and (co-)editor of over 100 refereed papers in scientific journals, books, and conference proceedings.

We have completed and/or ongoing research collaborations with all three researchers. With Prof. Dr. Stefanie Rinderle-Ma, we worked, among others, on extracting process information from textual resources in the context of compliance checking [72]. With Prof. Dr. Matthias Weidlich, we addressed various topics related to process analysis [43, 69], process mining [25, 65], and low level event data [17, 76]. With Prof. Dr. Simone Ponzetto, we have on ongoing cooperation on the extraction of process information from text. Based on this experience, we are confident that all three can provide valuable input for this project. With respect to the work packages, we aim to primarily collaborate with Prof. Rinderle-Ma on WP1 and WP6, with Prof. Weidlich on WP2 and WP3, and

with Prof. Ponzetto on WP1 and WP5.

6.5 Researchers abroad with whom you have agreed to cooperate on this project

Outside Germany, we agreed to cooperate with Prof. Dr. Josep Carmona and Dr. Chiara Ghidini.

Prof. Dr. Josep Carmona: Josep Carmona is an Associate Professor at Universitat Politècnica de Catalunya (UPC). He received a PhD at the same university in 2004, under the supervision of Prof. Jordi Cortadella. His research interests include formal methods and concurrent systems, data and process science, business intelligence and business process management, and natural language processing. He has co-authored numerous research papers and organized various conferences and workshops. He is a founder of the International Conference on Process Mining, where he is part of the Steering Committee, and a member of the IEEE Task Force on Process Mining.

Dr. Chiara Ghidini: Chiara Ghidini is a senior research scientist at Fondazione Bruno Kessler within the Data and Knowledge Management (DKM) research unit. Her work in the area of distributed knowledge representation is well known and internationally recognized and she has published more than 100 conference and journal papers on the topics. She has served on many organising and program committees for conferences and workshops in the areas of multi-agent systems and the semantic web. She has been involved in a number of international research projects, including the EU-funded projects APOSDLE and Organic Lingua and is currently leading the interdisciplinary SHELL project on procedural and ontological knowledge acquisition and evolution at the Fondazione Bruno Kessler.

We have already worked with Prof. Carmona in the past, primarily on the intersection between NLP and process analysis [2, 58]. With Dr. Ghidini, we have an ongoing collaboration on the use of deep learning for the analysis of textual process descriptions. Against this background, we believe that cooperations with Prof. Carmona and Dr. Ghidini will be a valuable addition to this project, primarily with respect to work packages WP1, WP2, and WP5.

6.6 Researchers with whom you have collaborated scientifically within the past three years

The researchers listed above are also ones we have collaborated with over the last three years. We plan to use this project to continue and deepen these collaborations. Other research collaborations in the context of process mining, process analysis, and natural language processing have been conducted with the researchers below.

Prof. Dr. Henrik Leopold

- Prof. Dr. Hajo Reijers (Utrecht University, NL): Process mining and analysis
- Prof. Dr. Mathias Weske (Hasso Plattner Institute, DE): Process mining and automation
- Prof. Dr. Flávia Maria Santoro (Universidade do Estado do Rio de Janeiro, BR): Process modeling and analysis

Prof. Dr. Han van der Aa

- Prof Dr. Avigdor Gal (Israel Institute of Technology, IL): Process mining
- Prof. Dr. Jan Mendling (Humboldt-Universität zu Berlin, DE): NLP-based process analysis
- Prof. Dr. Heiner Stuckenschmidt (University of Mannheim, DE): AI for process analysis
- Dr. Fabrizio Maggi (Free University of Bozen-Bolzano, IT): Declarative process analysis
- Dr. Adela del-Río-Ortega (Universidad de Sevilla, ES): Process performance analysis

6.7 Project-relevant cooperation with commercial enterprises

We do not plan to cooperate with a commercial enterprise in the context of this project.

6.8 Project-relevant participation in commercial enterprises

We do not have any project-relevant participation in commercial enterprises.

6.9 Scientific equipment

We will not need any further scientific equipment.

6.10 Other submissions

We currently have no other proposal submissions under review for this project.

7 Requested modules/funds

7.1 Basic Module

7.1.1 Funding for Staff

We apply for two doctoral researchers (TV-L 13) for the time of 2 years to conduct the work planned for this project (24 PM). Each position will be assigned to a PhD student who will be responsible for one work stream.

Doctoral researcher (WS1 - Van der Aa), TV-L 13, 24 months	$24 \times 6175.00\,€$
Doctoral researcher (WS2 - Leopold), TV-L 13, 24 months	$24 \times 6175.00\,€$
Total funding for staff	$296\,400.00\,€$

7.1.2 Direct Project Costs

7.1.2.1 Equipment up to €10,000, Software and Consumables

We will not require additional hardware or software. The required infrastructure will be provided by the Kühne Logistics University and the University of Mannheim.

7.1.2.2 Travel Expenses

The results of the project will be published and presented at national and international conferences. We plan to visit two conferences per year. Assuming an average of €1500 per conference, we will need 2 x 2 €1500 = €6000 per PhD student, i.e. €12000 in total. We plan to conduct two joint project meetings per year: one in Hamburg and one in Mannheim. This can be covered by €500 per trip, which means we need €2000 for project meetings. We further plan to visit two of our national and one of our international project partners over the course of the project. The trips to the national partners can be covered by €500 and the international partners by €1000. This sums up to €4000 for the entire project. A summary and the overall total required for traveling expenses are listed below.

Conference visits (2 years x 2 visits x 2 researchers)	$8 \times 1500.00\,€$
Project meetings (2 years x 2 visits)	$4 \times 500.00\,€$
Visit national project partner (2 partners x 1 visit x 2 researchers)	$4 \times 500.00\,€$
Visit international project partners (1 partner x 1 visit x 2 researchers)	$2 \times 1000.00\,€$
Total travel expenses	$18\,000.00\,€$

7.1.2.3 Visiting Researchers

As explained in Sections 6.4 and 6.5, we plan to cooperate with several national as well as international researchers. Costs of mutual visits are listed in above in Section 7.1.2.2.

7.1.2.4 Expenses for Laboratory Animals

Not applicable.

7.1.2.5 Other Costs

There will not be any other costs than the positions listed above.

7.1.2.6 Project-related publication expenses

We do not expect any publication expenses besides the conference fees, which we already included in the travel expenses in Section 7.1.2.2. The total for direct projects costs is listed below.

7.1.3 Instrumentation

Not applicable.

7.2 Module Temporary Position for Principal Investigator

Not applicable.

7.3 Module Replacement Funding

Not applicable.

Appendices

The appendix includes the applicant's CVs (*CV_PubList_Leopold.pdf* and *CV_PubList_VanDerAa.pdf*) with a list of their ten most important publications.