Linköping University | Department of Science and Technology
Master's thesis, 30 ECTS | Computer Science
2026 | LiU-ITN-TEK-A–26/022–SE

Interactive visual mining of event sequence patterns at multiple levels of event granularity

Interaktiv visuell analys av händelsesekvensmönster på flera granularitetsnivåer

Caitlin Wu

Supervisor: Katerina Vrotsou
Examiner: Andreas Kerren

Linköpings universitet
SE-581 83 Linköping
013–28 10 00, www.liu.se

Abstract

Event-sequence data is increasingly used to understand complex processes in domains such as healthcare, manufacturing, user behaviour, and workflow analysis. However, a central challenge in event-sequence analysis is choosing an appropriate level of event granularity. Traditional process mining and sequential pattern mining approaches often require events to be represented at a fixed abstraction level before analysis, which can hide important patterns or make results difficult to interpret. This thesis addresses the lack of visual analytics systems that support interactive exploration of event-sequence patterns across multiple levels of granularity.

The aim of this thesis is to design, implement, and evaluate an interactive visual analytics prototype for exploring and refining event-sequence patterns through local granularity adjustment. The work follows an iterative design-study-inspired process, including literature review, problem characterisation, task abstraction, requirement definition, prototype development, and qualitative evaluation with domain experts. The resulting prototype, `SCOPE`, supports coarse-to-fine exploration of pre-mined event-sequence patterns. It enables users to selectively split individual event nodes into lower-level subcategories, preview the consequences of these changes, apply constraints, inspect matching raw event traces, and compare saved patterns at different abstraction levels.

The evaluation indicates that local granularity adjustment can support exploratory analysis by helping analysts understand how high-level patterns relate to more detailed behaviours. The prototype also demonstrates how coordinated graph, ontology, bar-chart, and timeline views can make changes in abstraction more transparent. While `SCOPE` is not optimised for scalability or real-time mining, it contributes design insights into how visual analytics systems can support mixed-granularity pattern exploration. The thesis concludes that interactive, local control of event granularity can improve the interpretability and flexibility of event-sequence analysis, and identifies opportunities for future work in richer interaction design and integration with mining algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Event sequence data is widely used in modern digital systems. An event sequence consists of individual *events*, which are observable activities occurring at a specific time. Examples of events in different areas include:

- **Healthcare:** taking medicine, taking a blood test, visiting a doctor.
- **Manufacturing:** starting a machine, assemble a part, performing quality check.
- **User behaviour:** clicking a button, scrolling a page, adding an item to a shopping cart.
- **Workflow optimisation:** assigning a task, completing a task, approving a request.

Across these areas, discovering patterns in event sequences can help analysts understand complex and interconnected processes.

A major challenge in event sequence analysis is the **level of granularity** used to describe events. Some events are detailed, fine-grained, while others are more general, coarse-grained. For example, in healthcare a coarse event such as *patient treatment* may consist of more detailed events like *taking medicine* or *having a blood test*. Traditional pattern-mining methods require a *fixed* level of granularity before analysis begins.

As shown in previous research, choosing this level during preprocessing is difficult and greatly affects what patterns can be found. Important details may be lost making it harder to explore data at different levels of detail. These limitations are especially clear in domains where behaviour naturally occurs across multiple levels [3].

1.1 Background

Process mining and event sequence analysis rely on logs that describe activities over time. Before analysis, these events are often abstracted using domain knowledge, hierarchies, or temporal rules. Sequential Pattern Mining (SPM) uses the same type of sequential event data, but focuses on discovering frequent subsequences that repeatedly occur across multiple cases. Although abstraction can simplify the data, it also introduces limits. Analysts rarely know in advance which level of granularity will be most useful, and the "right" level may change as the analysis develops. Choosing a fixed granularity too early can hide important patterns, create misleading results, or prevent certain analyses [3, 6].

Mining algorithms also struggle when logs contain very fine-grained or mixed levels of abstraction, often producing models that are difficult to read. This affects both process mining and SPM, where overly detailed logs may generate too many low-level patterns, while coarse abstractions may remove important behavioural information. After abstraction, coarse events usually cannot be traced back to their detailed origins, reducing transparency and making explanation harder. These problems highlight the need for methods that support exploration and pattern mining across multiple granularity levels.

1.2 Problem formulation

Existing event-sequence mining approaches require that all events be represented at a single level of granularity chosen during preprocessing. This fixed granularity constraint significantly impacts pattern discovery, because some patterns become visible only at specific abstraction levels, while others disappear entirely. Existing systems provide partial support for interactive or constraint-based granularity exploration, but limited support for direct, visual, node-level granularity adjustment within a selected pattern while preserving context and showing consequences before applying changes.

What is missing is a flexible, interactive approach that enables analysts to

- dynamically adjust event granularity during the mining process,
- explore and compare patterns at multiple abstraction levels,
- smoothly transition between fine-grained and coarse-grained representations within a single analytical workflow.

The problem addressed in this thesis is the lack of visual analytics systems that support these functionalities.

1.3 Aim

The aim of this thesis is to design, implement, and evaluate an *interactive visual analytics system* that supports the exploration and refinement of event sequence patterns at *multiple levels of granularity*, allowing analysts to dynamically adjust abstraction levels during the mining process.

1.4 Research questions

This thesis is guided by the following research questions:

RQ1 How is multi-level mining of events handled today in event sequence analytics approaches?

RQ2 How can event granularity be interactively and locally adjusted for individual nodes during the mining process, while ensuring that these changes are clearly communicated to the user?

RQ3 How can visualisations enable analysts to interactively explore and compare patterns across different levels of event granularity?

1.5 Objectives

To address the research questions, several objectives must be fulfilled. Each objective is directly linked to a specific research question and serves to motivate and structure the overall analysis.

1. Survey existing event sequence analytics literature to identify how event granularity and hierarchy are currently modelled. **(RQ1)**
2. Analyse strengths and limitations of current multi-level event mining approaches with respect to flexibility and scalability. **(RQ1, RQ2)**
3. Investigate existing algorithms and system designs that support incremental or interactive adjustment of event granularity during mining. **(RQ2)**
4. Survey existing visualisation techniques for multilevel temporal and sequential data analysis. **(RQ1, RQ3)**
5. Define requirements for dynamic granularity control from an analyst's exploratory workflow perspective. **(RQ2, RQ3)**
6. Identify interaction techniques that allow analysts to navigate, filter, and compare patterns across granularities. **(RQ3)**
7. Design and implement a visual analytics prototype for exploring mined patterns across multiple event levels. **(RQ2, RQ3)**
8. Evaluate how analysts understand and interpret changes in event granularity through interactive visualisations. **(RQ2, RQ3)**

1.6 Scope and delimitations

This thesis will focus on the **interactive exploration** and **visual mining aspects** of multi-granular event data. Therefore, several delimitations are set. The focus will not be on making the system optimised for scalability, domain-specific performance, or computational efficiency. Instead, the system will use existing mining techniques and previously mined patterns. Scalability will still be considered in the literature review because it affects how feasible different multi-level mining approaches are, but will not be treated as an implementation goal.

Chapter 2

Theory

Over the last decade, the amount of data collected and stored has increased rapidly as data storage technologies have improved. But this raw data increases faster than our ability to understand and use it. This creates what is called the *information overload problem*, where data is either not used at the moment or is shown in a way that is hard to interpret. In many organisations, data is collected as **event sequences**, where each event represents an activity in a real process. These events can be recorded at different **levels of detail (granularity)**, from high-level tasks to very small system actions. When examined properly, this data can reveal how processes actually run in practice.

Data mining focuses on extracting useful knowledge, interesting characteristics, and hidden patterns from raw data [7]. Within this field, **Sequential Pattern Mining (SPM)** identifies patterns as frequent subsequences that repeatedly occur across multiple event sequences, based on a user-defined minimum support threshold [8]. **Process Mining (PM)** also works with event sequence data, but focuses more specifically on analysing event logs to discover, monitor, and improve real processes [9]. While SPM emphasizes recurring sequential patterns, PM aims to capture and evaluate the overall process flow of the entire log.

Visual analytics (VA) helps turn large and complex data into something understandable through interactive visualisations [10]. When integrated with process-oriented analysis such as PM and SPM, VA helps transform complex event sequence data into interpretable insights about how processes behave in practice.

2.1 Event-sequence data

An **event** is typically defined as a discrete occurrence containing one or multiple attributes such as timestamp, duration, and a descriptive label [11]. Figure 2.1 illustrates an example of an event together with several commonly associated attributes, including identifiers, temporal information, and activity labels.

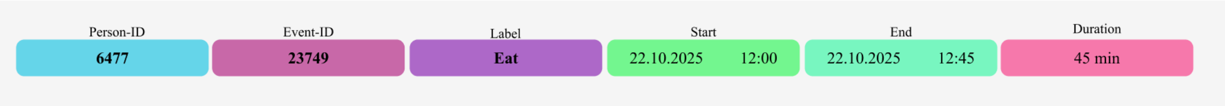


Figure 2.1: Example of an event and a selection of associated attributes commonly found in event-sequence data.

Case ID	Timestamp	Activity	Resource	Lifecycle
12373	11:00	Register request	Barbara	Start
12373	11:30	Register request	Barbara	Complete
12374	11:32	Review request	Jan	Start
12375	11:50	Review request	Jan	Complete

Table 2.1: Example event log table, where each row corresponds to an *instantaneous event*, adapted from [5].

Case ID	Activity	Resource	Start	Complete
12373	Register request	Barbara	11:00	11:30
12374	Register request	Jan	11:32	11:50

Table 2.2: Example event log table, where each row corresponds to an *activity interval* with a start and completion timestamp (duration).

In Business Management, events are captured in **event logs**, where each row represents one recorded event associated with a real-world process execution [12]. An **event sequence** describes a temporally ordered series of activities or actions that occur within a process. Collections of such sequences form **event-sequence data**, which is common in domains where real-world processes are logged over time, for example in healthcare, business operations, or information systems. In event-sequence analysis, the goal is often to understand how these processes unfold, identify recurring patterns, and detect deviations or unusual behaviour.

Analysis of event-sequence data is a relatively new field compared to the analysis of other data types [13]. In PM, event logs often describe activities performed as part of a process, where one real-world activity may correspond to several logged events depending on how the lifecycle is recorded. For example, an activity such as *Register request* may generate both a *Start* event and a *Complete* event [5]. This distinction is important, since event logs may represent either individual event occurrences or higher-level activity instances derived from multiple related events. The presence of mixed/overly fine granularity in event logging is a problem in many PM techniques [5].

In Table 2.1, each row corresponds to a single event. The first two rows illustrate how the same activity can be represented by multiple events occurring at different stages of its lifecycle (e.g., *Start* and *Complete*). Table 2.2 presents the same event log in an alternative format. In Figure 2.2, the two lifecycle events together represent a single activity instance, modelled as an event interval with a start and completion timestamp.

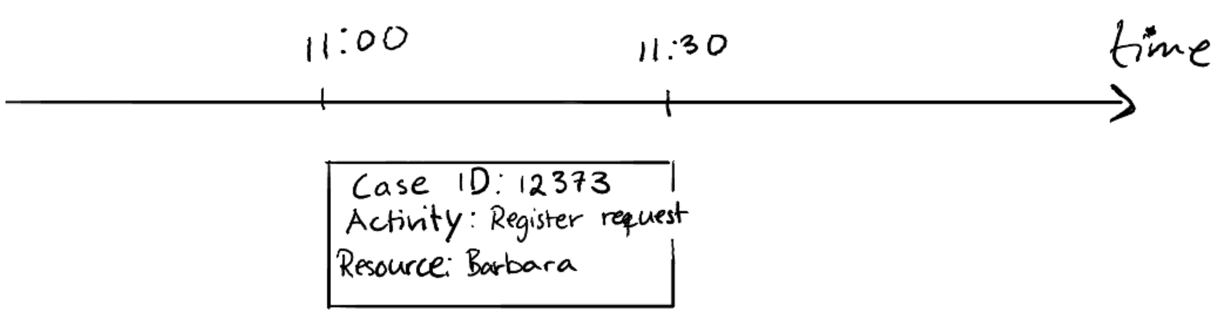


Figure 2.2: Example of an activity instance derived from Table 2.1 and 2.2, modelled as an event interval.

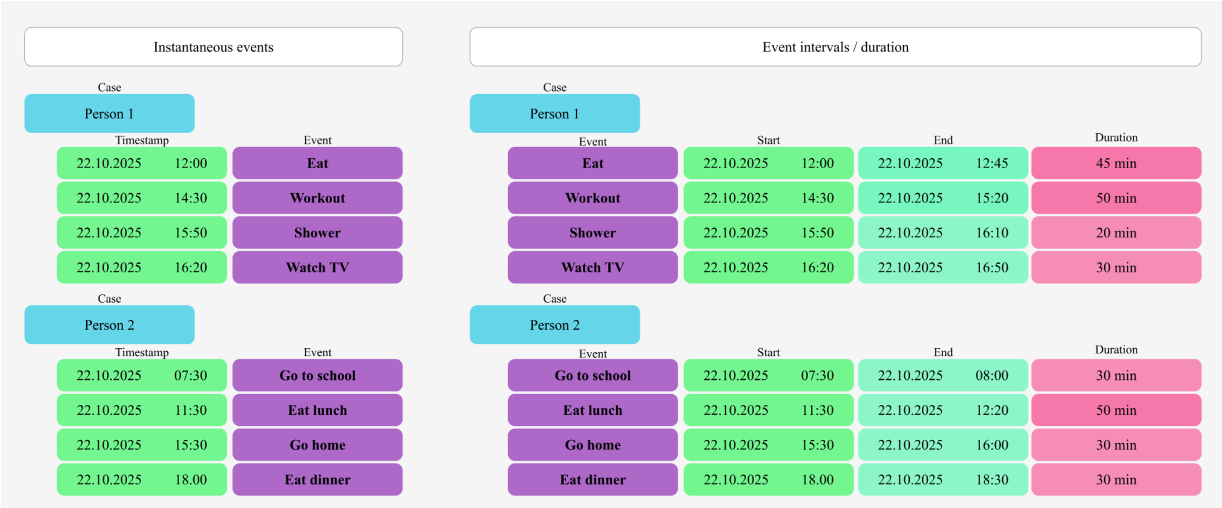


Figure 2.3: Example of instance-based case-event enumeration. Each person represents a unique case with an associated case ID and corresponding events. The same event data is illustrated using both instantaneous event representations and interval-based representations with duration. Inspired by [1].

Events that belong to the same process execution are linked through a shared identifier. In event-sequence analysis, this identifier is commonly referred to as a **sequence ID**, while in process mining it is more often called a **case ID**. In Figure 2.1, the person identifier acts as the shared identifier connecting events associated with the same process execution. The ordered set of events associated with one sequence or case forms an **event sequence**, also referred to as a **trace** in process mining. A trace can therefore be understood as a sequence of activities [14]. An event log therefore consists of multiple cases, each corresponding to one process execution. In addition, several sequences may be grouped into a **cohort**, for example, when they share a common attribute or belong to the same population [11].

2.1.1 Representations

There are different ways to represent event sequence data. Yeshchenko *et al.* distinguish between **instance-based** and **model-based** representations [1]. Instance-based representations describe the actual observed event sequences, preserving concrete events, their temporal order, and their association with individual cases. In contrast, model-based representations describe more abstract structures derived from the data, such as patterns, rules, or higher-level specifications that summarize regularities across multiple sequences. While instance-based representations focus on concrete *observed* behaviour, model-based representations focus on *generalized behaviour* inferred from multiple sequences.

Among instance-based representations, **case-event enumeration** links each event explicitly to a case while preserving the temporal order of events [1]. This representation is particularly suitable for event sequence analysis because it maintains the sequential structure necessary for both process mining and visualisation, while also allowing analysts to inspect concrete traces behind broader patterns. For example, a case may represent a patient, customer, or process instance, while the events describe the activities associated with that case over time. In Figure 2.3, the cases represented are persons, while the associated activities correspond to the recorded events within that case during a part of a day.

When referring to model-based representations, these focus on summarizing and generalizing behaviour across multiple event sequences. Figure 2.4 illustrates an example of a model-based representation, where activities and transitions are aggregated into a hierarchical graph structure. The nodes represent activities, while the edges represent transitions between them. Such representations provide a compact overview of behavioural structures that would be difficult to identify directly from raw event sequences.

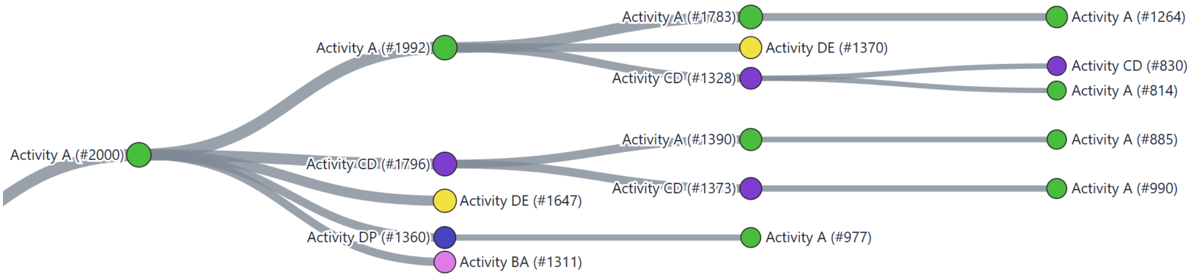


Figure 2.4: Example of a model-based representation of event-sequence data, where extracted sequential patterns are represented through aggregated activities and transitions in a hierarchical graph structure reflecting recurring behaviour within a dataset.

In this project, both representation types will be relevant and complementary. They will form the basis for the remainder of the thesis by supporting both abstraction across many sequences and inspection of individual process instances. These representations also influence how event sequence data can be visualised and explored interactively.

2.2 Visualisation and interaction design

When designing a visual interface, it is important to help users understand information quickly without overwhelming them. Ben Shneiderman describes this through his well-known Information-Seeking Mantra: “Overview first, zoom and filter, then details-on-demand” [15]. According to this principle, users should first be presented with an overview of the entire information space, followed by opportunities to zoom in and filter relevant content, and finally access detailed information when needed. This approach supports intuitive exploration and improves usability by making complex information easier to navigate and understand.

2.2.1 Visualising event sequences

Event sequence data can be visualised in many different ways depending on the analytical task, the scale of the data, and the level of abstraction being considered. Since event logs contain temporal information, time plays a central role in many representations [16]. At the same time, analysts often require both detailed inspection of individual traces and summarised views of larger collections of sequences.

One common way to represent instance-based event-sequence data is through **timelines**, where events are arranged according to time and encoded using position, colour, shape, or size [4]. Timeline-based views are especially useful for showing raw temporal behaviour and durations. In their survey of event sequence comparison techniques, van der Linden *et al.* note that timelines are frequently used when analysts need to compare one selected sequence against multiple others, often using **juxtaposed** layouts (where sequences are placed side by side for direct visual comparison) [11]. For visualising event types and event distributions in event sequences, charts are frequently used, such as bar charts and scatter plots, whereas matrix-based visualisations are usually the choice for presenting a summary of event frequency or frequent patterns [4].

In PM a commonly used abstraction, for event sequences, is the **variant view**, where cases sharing the same activity order are grouped together. Variant-based representations reduce complexity by summarising repeated behaviour while preserving sequential structure. Because they provide a manageable overview of process executions, they are often used as an entry point for process exploration [17].

When event data contains hierarchical relationships, other representations become relevant. Hierarchical event structures may be visualised using node-link trees, tree-maps, or icicle plots, all of which support the display of parent-child relationships and multiple levels of abstraction [4, 1]. Such visualisations are particularly relevant when low-level event types can be grouped into broader categories or when analysts need to move between detailed and abstract views of the same data.

2.2.2 Interactions

Often, one single view is not sufficient for all analytical tasks. When analysing event sequences that contains multiple levels it is often beneficial to combine several representations. Existing work suggests that analysts need both overview and detail views, and that relationships between these should be supported through interaction [11]. This is especially important when comparing patterns across granularities, where the challenge is not only to show different abstraction levels, but also to preserve context between them. Interaction therefore plays a central role in event sequence visualisation.

Yi *et al*. identify seven general categories of interaction intent in information visualisation: **select**, **explore**, **reconfigure**, **encode**, **abstract/elaborate**, **filter**, and **connect** [18]. These categories provide useful guidelines for describing what analysts seek to do when interacting with data. In the context of event sequence analysis, more domain-specific interaction techniques have also been identified. Guo *et al*. discuss techniques such as filtering, segmentation, scaling, emphasis, and aggregation [4], while van der Linden *et al*. highlight the importance of **alignment**, where sequences are aligned around one or more events or categories to support comparison, which is often done in juxtaposed visualisations [11].

To summarize, the literature suggests that multi-level analysis requires not only **multiple** representations, but also suitable interaction techniques that preserve context when moving between them. This is particularly important when abstraction levels are changed dynamically, since analysts must be able to understand both what changed and how the new view relates to the previous one.

2.3 Process mining

Process mining connects data science with how processes work in real life. In general PM takes event log data as input and outputs a model, often visualised in the form of event sequence graphs, that explains the link between two events [19]. By analysing these connections, organisations can learn from data created when their events are carried out, so they can better understand and improve the way they work [20]. Among many analytical methods, PM is special because it analyses and visualises data with a focus on how processes actually happen over time. It can look at the order of events for each case and by studying event logs using different tools, process analysts can clearly see how processes really work, find problems or delays, and suggest ways to make them better [21].

Figure 2.5, displays the pipeline of the three basic types of PM when it comes to input and output [2]. **Discovery** aims to build a process model directly from event log data. **Conformance checking** evaluates the fit between the created model and reality. **Enhancement** updates the model using event data to make it more accurate and useful, usually by adding information.

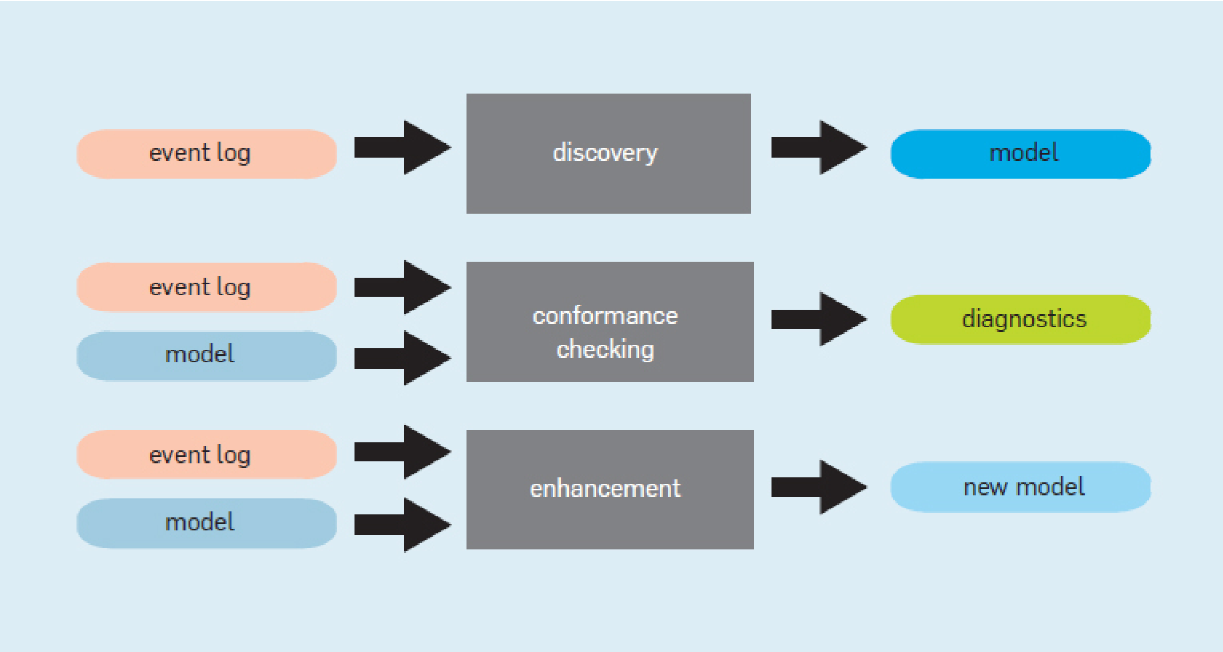


Figure 2.5: The three basic types of process mining in terms of input and output (*reproduced from* [2]).

The result of PM is often shown through a **process model**, which describes the intended or observed behaviour of a process [14]. Common process model representations include Petri nets, Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC), UML activity diagrams, directly-follows graphs (DFG), and process trees [22, 16, 14]. Process trees are particularly useful when hierarchical structure and multiple levels of abstraction are important [14]. Among these, DFGs and BPMN are especially common in practice, as they effectively visualise execution frequency, transitions, performance, and cost within processes [16].

Although PM has proven useful in many domains, several limitations remain, especially with respect to interaction and abstraction. The *Process Mining Manifesto* highlights challenges such as dealing with event logs of **varying granularity**, combining PM with other forms of analysis, and improving the understandability of results for non-expert users [22]. These challenges are directly relevant to this thesis. When event data is too detailed, discovered models may become overly complex and difficult to interpret. At the same time, simplifying or filtering the data too aggressively may hide important behaviour.

Another limitation is that many process discovery approaches remain largely automated and offer little user interaction during the mining process. Schuster *et al*. observe that most existing approaches provide limited opportunities for users to guide the discovery process or inject domain knowledge while a model is being built [23]. This black-box approach reduces transparency and makes it difficult for analysts to iteratively adjust the **level of detail** during exploration. Some work has begun to address this issue. For example, Schuster et al. propose an incremental approach for discovering hierarchical process models, where behaviour can be added trace by trace and the effects on the resulting model can be observed continuously [14]. Their method, based on the lowest common ancestor (LCA) in a process tree, illustrates how PM can move toward more incremental and interactive workflows. While their approach does not fully solve the problem of dynamic multi-level mining, they provide a useful foundation for systems in which analysts can steer the analysis process and inspect intermediate results.

2.4 Sequential Pattern Mining

Sequential Pattern Mining focuses on discovering recurring subsequences of events in ordered data. Since PM also aims to identify common behaviour in event logs, SPM can be seen as a complementary technique that formalizes the discovery of statistically significant sequential patterns across datasets [24]. In *Sequential Pattern Mining – Approaches and Algorithms* [8], SPM is described as the task of finding frequent subsequences that occur repeatedly across multiple event sequences. A pattern is considered significant if its support, that is, the number or proportion of sequences containing it, exceeds a user-defined minimum threshold.

One of the most widely used algorithms for this task is PrefixSpan, a pattern-growth approach that efficiently discovers frequent subsequences without generating a large number of candidate patterns [25]. Frequent Sequence Mining (FSM) can be viewed as a specific form of SPM, focusing explicitly on subsequences that occur frequently within temporal event data [26]. A major strength of SPM is its ability to summarize large and complex datasets by revealing recurring behavioural structures. However, most traditional SPM approaches assume a fixed event granularity before mining begins, which limits exploration across multiple levels of detail. In complex real-world data, this can reduce interpretability and make it harder to understand how patterns relate across abstraction levels [27].

2.5 Visual analytics

Visual analytics has been defined as the science of analytical reasoning supported by **interactive visual interfaces**, with the goal to aid the exploration of complex data [10]. By combining the strength of computers and humans, interactive visualisation methods can be used to automate analysis [16]. This approach goes beyond standard static charts to make data exploration more interactive and intuitive. Its goal is to help people explore, interpret, and reason about complex data by combining computational methods with human judgment. Rather than relying only on automated analysis or only on manual inspection, visual analytics seeks to integrate both into a single interactive workflow. This is particularly relevant for event sequence data, where the volume of data, the temporal dependencies between events, and the need to compare multiple levels of abstraction can make purely automatic analysis difficult to interpret. In these cases, visual analytics can help analysts move between overview and detail, inspect intermediate results, and steer analysis toward areas of interest. This aligns with knowledge-assisted visual analytics approaches, where human expertise is integrated into the analytical process in a **human-in-the-loop** manner to improve the quality and interpretability of results [28].

2.5.1 PVA and PDA

An important idea in this context is **Progressive Visual Analytics**, (PVA) where partial or intermediate computational results are shown while the analysis is still running [29]. PVA is closely related to Progressive Data Analysis (PDA), which is described as a human-in-the-loop process that supports data exploration at scale [30]. In traditional batch-oriented workflows, analysis is often performed first and only visualised once computation has finished. This creates a *compute-wait-visualise* workflow, which is often time-consuming for analysts. A few methods have been proposed which address such delays. One of these methods is **visualisation of partial results**, allowing analysts to gain early insights during the analysis process. This can provide meaningful guidance for the next steps in the exploration, or help filter out paths that are unlikely to be relevant, thereby saving both time and computational resources. This is a problem that Vrotsou *et al.* [25] address with their system, `ELOQUENCE`, which allows the user to decide what parameters to change locally, with the help of local constraints, thus incorporating expert knowledge in the mining process to prevent the compute-wait-visualise problem.

When designing a PVA/PDA system, an important consideration is how users are supported while the visualisation changes during analysis. This includes deciding **when** new results should be presented and **how** much these updates should be visually communicated [30, 31].

Regarding *when* to update, a simple approach would be to refresh the visualisation every time a change is made. However, this often creates rapidly changing views that are distracting, difficult to interpret, and challenging to interact with. A better approach is often to update only when requested, allowing users to stay in control of the analysis process [30].

Regarding *how* changes are presented, two main strategies used are *extension* and *overwrite*. Extension adds new visual elements while preserving previous results, whereas overwrite replaces previous results with a new visualisation state. Overwrite can be either partial, where only changed elements are updated, or full, where the entire previous view is replaced [30]. To improve interpretability, systems should also clearly highlight where changes occur, for example through differences in size, shape, or position.

The when and how are closely connected to guidance, since users need support in understanding what changed, why it changed, and what effect different analysis choices may have. In complex interactive analytical systems, good guidance improves transparency and helps users make informed decisions during exploration [31].

For this thesis, PVA/PDA will provide the foundation for combining exploratory event sequence analysis with interactive visualisation. In particular, it will provide a workflow in which analysts can iteratively adjust event granularity, inspect the consequences of those changes, and explore patterns across **multiple abstraction levels** rather than relying on one level of granularity for the chosen pattern.

2.6 Granularity and hierarchy in event sequences

A central challenge in event sequence analysis is the choice of **granularity**, i.e., the abstraction level at which activities and event sequences are represented and analysed. Real-world event logs often contain activities that can be described at multiple levels of granularity, where coarse-grained activities consist of several finer-grained activities that may be organised hierarchically [3]. For example, a high-level activity such as *Bake pizza* may be represented through finer-grained events such as *Make dough*, *Add toppings*, and *Put in oven*, as illustrated in Figure 2.6.

These hierarchical relationships are common in event-sequence data, where coarse-grained activities may consist of multiple lower-level actions organised across abstraction levels. Hierarchical structures therefore provide a foundation for abstraction, grouping, and multilevel exploration in both event-sequence analysis and process mining.

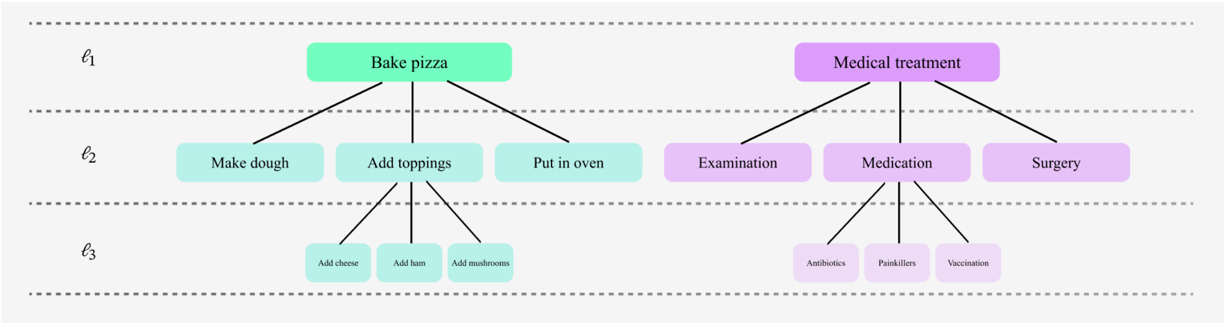


Figure 2.6: Examples of semantic event granularity represented through hierarchical activity structures across multiple abstraction levels. Inspired by [3].

In traditional process mining workflows, the granularity level is often fixed during preprocessing through event-abstraction techniques that aggregate fine-grained events into more coarse-grained business activities [3]. Aggregation may be based on temporal proximity, semantic similarity, domain hierarchies, or recurring execution patterns. While such abstractions reduce complexity and provide more compact representations of event sequences, they also introduce a trade-off, since increasing abstraction may hide important low-level behaviour or rare events [32].

2.6.1 Granularity of analysis and visualisation

Granularity does not only concern how activities are represented, but also the level at which event sequence data is analysed and visualised. Guo *et al.* describe four levels relevant to event-sequence analysis: events, subsequences, sequences, and sequence collections [4, 33]. As illustrated in Figure 2.7, analysis may focus on individual events (A), subsequences (B), complete sequences (C), or collections of sequences (D). Analysts therefore often need to move between overview and detail depending on the analytical task.

Existing approaches often focus on one of two extremes. Some techniques summarize large-scale event sequence data to reveal high-level structures and recurring patterns, while other methods focus on detailed temporal behaviour at the level of individual events [33]. Analysts therefore often need to move between overview and detail during exploration.

2.6.2 Fixed versus dynamic granularity

Choosing a suitable granularity level beforehand can be challenging, since the purpose of the analysis and the most meaningful abstraction level often evolve during exploration [3]. In exploratory analysis, analysts do not always begin with a clearly defined question or know in advance which level of abstraction will be most informative [34]. Instead it is during the inspection of the data from different abstractions and viewpoints that eventually lead to insights. Because fixating the granularity level during preprocessing also affects what patterns can be observed and analysed, patterns may exist at multiple abstraction levels simultaneously and, in some cases,

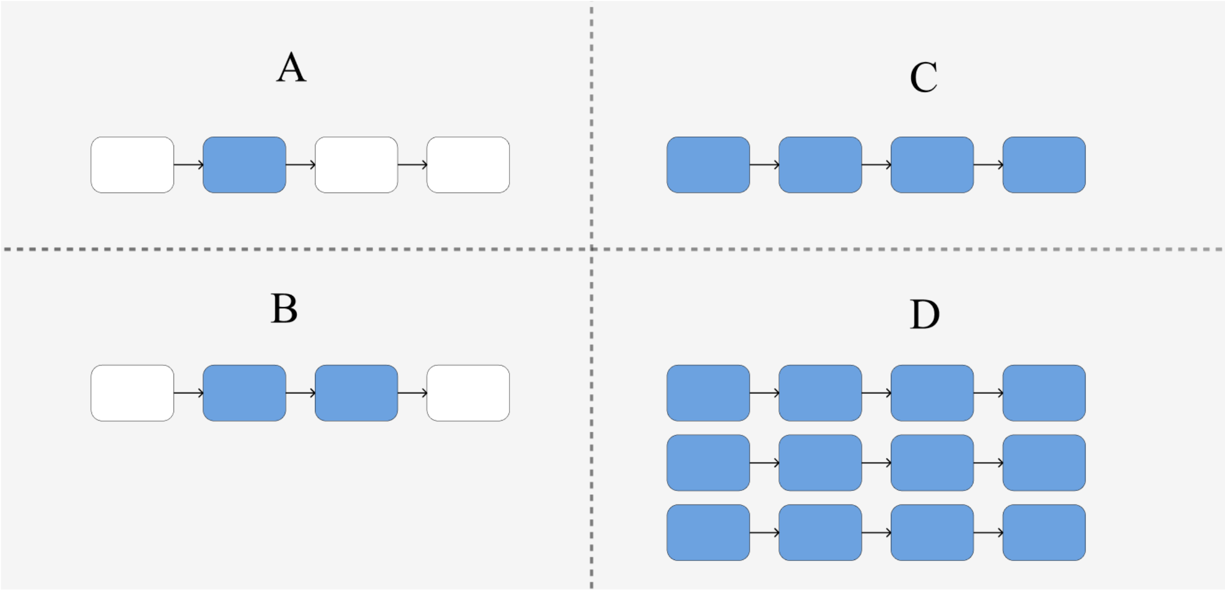


Figure 2.7: Different levels of granularity in event-sequence analysis: (A) individual events, (B) subsequences, (C) complete sequences, and (D) collection of sequences. Inspired by [4].

span across granularity levels [3]. A fixed granularity representation may therefore prevent analysts from identifying meaningful fine-grained behaviour, relationships between abstraction levels, or patterns **combining** both aggregated and detailed events.

This challenge is particularly relevant in process-oriented analysis, where analysts often begin with a high-level overview of frequent behaviour and progressively drill down into specific events, paths, or subgroups of cases. Fehrer *et al.* identify tasks such as drill-down and deep-dive as common workflows in event log analysis, highlighting the need for methods that support flexible granularity adjustment during exploration [35].

2.6.3 Local versus global granularity adjustment

A useful distinction can be made between **global** and **local** granularity adjustment. Global granularity adjustment changes the abstraction level for the entire dataset or visualisation. Local granularity adjustment instead changes the level of detail only for selected events, subsequences, or sub-patterns while preserving the surrounding context. Existing approaches based on SPM primarily support global summarization of event sequence data, while providing limited support for flexible local granularity adjustment. For example, `Frequence` enables users to iteratively refine and re-mine subsequences across multiple levels of detail [26]. However, the refinement operates at the level of complete patterns or filtered subsets of the data, rather than allowing independent adjustment of individual parts within a single sequence.

Similarly, `Sequen-C` supports multilevel exploration of event sequences through hierarchical aggregation and interactive control of overview granularity [27]. The level of detail can be adjusted across sequence clusters and cluster representations, but remains consistent within each representation rather than varying for specific events or sub-patterns.

`ELOQUENCE` extends this idea further by supporting local ontology-level constraints and progressive refinement during the mining process [25]. Different parts of the search space may therefore be explored at different abstraction levels. However, these adjustments remain constraint-driven and tied to the mining process itself, rather than enabling direct manipulation of individual nodes or events within a visual sequence representation.

Consequently, existing approaches generally do not support **local granularity adjustment** within the same pattern or subsequence. In particular, constructing mixed-granularity representations, where some activities remain aggregated while others are shown in fine detail, remains difficult.

2.7 Related work

The article *Visual Analytics Meets Process Mining: Challenges and Opportunities* highlights that visual analytics and process mining have largely progressed as separate research fields despite both focusing on the analysis of event data and processes [16]. The authors argue that combining interactive visual exploration from visual analytics with process-oriented analysis from process mining can provide deeper insights into complex event-sequence data. They further identify several open challenges related to interaction, abstraction, scalability, and interpretability when integrating visual analytics and process mining approaches.

Several systems have since explored different ways of combining interactive visualisation, process mining, and sequential pattern mining for exploratory event-sequence analysis.

2.7.1 ELOQUENCE

`ELOQUENCE` is an interactive visual sequence mining system proposed by Vrotsou and Nordman [25]. The system addresses two major challenges in sequential pattern mining. The first is the large number of extracted patterns and the second computational cost of mining them interactively. Instead of treating the mining process as a black box, `ELOQUENCE` allows users to progressively refine constraints while patterns are generated, thereby supporting a more transparent and interactive workflow. A key contribution of the system is its support for local ontology-based constraints and mixed-granularity pattern exploration during mining. Different parts of a pattern may therefore be explored at different abstraction levels. The system combines interactive filtering, progressive mining, and visual pattern exploration through a graph-based representation of extracted patterns. Figure 2.8 shows an example of a mixed-granularity pattern in `ELOQUENCE`.

2.7.2 Frequence

`Frequence` is a visual analytics system for exploratory frequent sequence mining [26]. The system combines sequential pattern mining with interactive visualisation to support iterative exploration of event-sequence patterns across multiple levels of detail. Users may progressively refine patterns, filter cohorts, and re-mine subsequences during exploration. The main contribution of `Frequence` lies in its integration of interactive visual exploration with iterative mining workflows, enabling analysts to move between overview and detail during sequence analysis. Exploration is driven through progressive refinement and re-mining of discovered subsequences, where abstraction changes are applied at the level of mined patterns, cohorts, or filtered sequence subsets. The system allows users to explore patterns at different abstraction levels, but each individual pattern is still represented using one consistent level of detail.

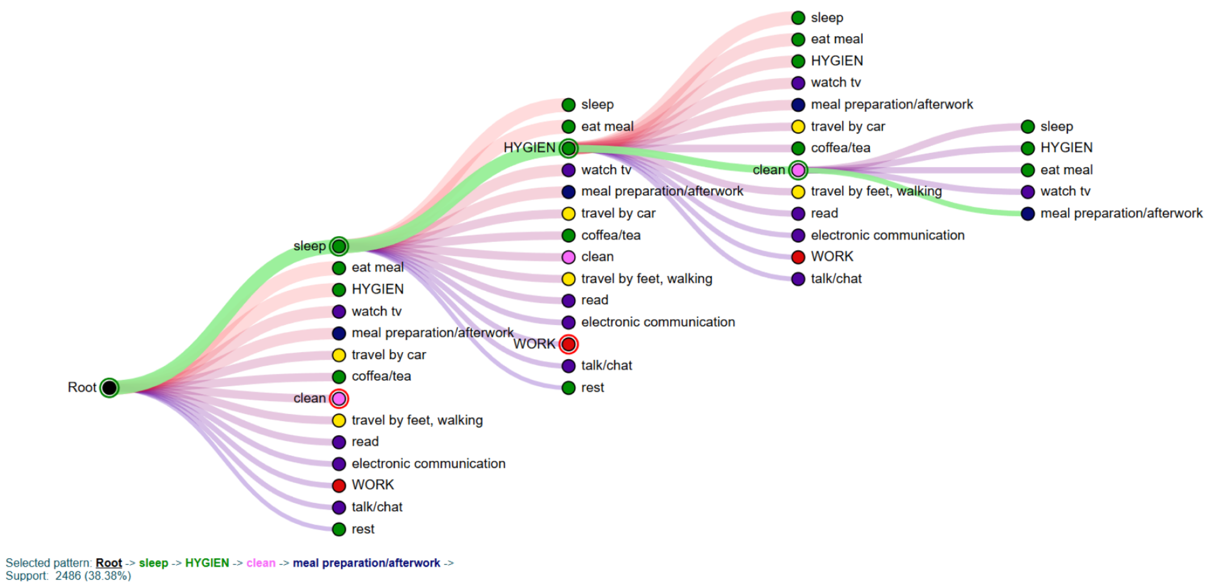


Figure 2.8: The pattern graph in the system ELOQUENCE.

2.7.3 Sequen-C

Sequen-C is a visual analytics approach for multilevel exploration of event-sequence data based on hierarchical aggregation and clustering [27]. The system supports exploration across different abstraction levels through adjustable overview granularity and sequence simplification mechanisms. A central contribution of Sequen-C is its support for multilevel representations of event sequences, enabling analysts to explore both large-scale behavioural structures and more detailed sequence patterns. Granularity adjustments are applied to cluster representations and overview levels, but not independently to individual events or sub-patterns within the same sequence.

2.7.4 EventBox

EventBox is a visual encoding and analysis approach for exploring temporal and multivariate attributes in event-sequence data [36]. The system extends Sequen-C with a dedicated representation for analysing groups of events through multiple attributes simultaneously, including temporal, categorical, and numerical attributes. The main contribution of EventBox is its support for multivariate event-sequence exploration through a visual encoding inspired by boxplots, scatterplots, histograms, and heatmaps. The system also includes interactive transformations such as aggregation, substitution, alignment, and sorting, as well as automatically generated statistical summaries. In contrast to systems that primarily focus on sequential structure or frequent pathways, EventBox emphasizes attribute relationships, temporal distributions, and outliers within event groups.

2.7.5 Cortado

Cortado is an open-source process mining tool designed for interactive and incremental process discovery [37]. The system combines multiple process mining techniques into an interactive workflow where analysts iteratively select observed behaviour and refine process models during exploration. A key feature of `Cortado` is its variant-based abstraction of event logs, where process executions are grouped into variants and explored through interactive visual representations supporting partially ordered behaviour and temporal overlap. The system contributes interactive exploration capabilities to process mining workflows, although abstraction changes primarily occur at the level of variants and process representations rather than through local mixed-granularity adjustments within individual patterns.

Together, these systems demonstrate the value of combining mining methods with interactive visual exploration. However, most existing approaches adjust abstraction at the level of complete patterns, clusters, cohorts, or process representations. This motivates the focus of this thesis on local granularity adjustment within individual events in event-sequence patterns.

Chapter 3

Method and design

This chapter describes the methodological approach used in the thesis. The theory chapter presented existing concepts, techniques, and systems related to event-sequence analysis, process mining, sequential pattern mining, visual analytics, and granularity. This chapter will describe how these concepts were translated into design decisions, interaction requirements, and finally a prototype system.

The work followed an iterative visual analytics design process inspired by the **Design Study Methodology**. The method consisted of analysing and characterising the problem, abstracting relevant terminology and analytical tasks, deriving design requirements from literature and expert input, designing and implementing a prototype, and evaluating the system with domain experts.

3.1 Research approach

A common approach for solving domain-specific visualisation problems is the **Design Study Methodology** (DSM). This method is particularly suitable when the goal is to design and evaluate a visualisation system for a specific analytical problem rather than to develop a general-purpose algorithm. In practice, DSM often consists of analysing the problem domain, abstracting data and tasks, designing and implementing a visualisation solution, evaluating the solution with users, and communicating the findings [38].

This thesis followed a workflow similar to DSM. The work focused on the iterative design and development of a prototype. Rather than optimising existing mining algorithms, the purpose of the project was to investigate how interactive visualisation and local granularity adjustment can support the user in an exploratory event-sequence analysis. Rapid software prototyping was used throughout this process, which Sedlmair *et al.* describe as a crucial skill in design studies, as ideas often need to be quickly tested, revised, or discarded [38].

The research process consisted of five main phases. First, the problem was identified and characterised through a literature review, analysis of existing systems, and discussions with domain experts. Second, terminology and analytical tasks were abstracted from related fields in order to bridge concepts from process mining, sequential pattern mining, and visual analytics. Third, design requirements were derived from theory, related work, and expert input. Fourth, a prototype was designed and implemented to demonstrate interactive local granularity adjustment. Finally, the prototype was evaluated qualitatively with domain experts from process mining and sequential pattern mining. The final evaluation procedure is described in detail in Section 5.

3.2 Design process

The overall research problem was transformed into a set of design considerations for the prototype. The design process consisted of problem identification and characterisation, domain and task abstraction, and literature-informed design. These steps were used to connect the theoretical background presented in the previous chapter with the practical development of the prototype.

3.2.1 Problem identification and characterisation

The first step in the design process was to analyse and define the research problem. As described in Section 1, event-sequence data is often recorded at different levels of detail, and choosing an appropriate granularity level before analysis can be difficult. Existing process mining and sequential pattern mining approaches often require the level of abstraction to be fixed before mining or preprocessing. This limits exploratory analysis, especially when analysts want to inspect one part of a pattern in detail while preserving the surrounding context at a higher abstraction level.

A literature review was conducted to understand how this problem has been addressed in previous work. The review focused on event-sequence visualisation, process mining, sequential pattern mining, event abstraction, hierarchical aggregation, and interactive visual analytics.

One important part of the problem characterisation was the analysis of the existing system `ELOQUENCE`. As described in Section 2.7.1, `ELOQUENCE` already supports interactive steering of the sequence mining process through local constraints. It also allows patterns to contain event types from different ontology levels, thereby enabling a form of mixed-granularity pattern exploration. However, the analysis of the system also revealed limitations. In particular, mixed-granularity patterns can be difficult to interpret, since the system provides limited visual guidance for understanding how categories at different levels relate to each other. This motivated the design of a system that more explicitly supports navigation between different levels and visually communicates the consequences of local abstraction choices.

The problem characterisation was further informed by discussions with process mining experts. During these discussions, the project idea and an early wireframe was presented and discussed. The experts emphasized that existing process mining tools mainly support global granularity changes, filtering, or preprocessing-based abstraction. They also noted that support for navigating between even two or three granularity levels locally would be a meaningful improvement over many existing workflows. This confirmed that local granularity adjustment was relevant not only from a visual analytics perspective, but also from the perspective of process mining practice.

3.2.2 Domain and task abstraction

A central part of the method was to abstract concepts and tasks from several related research areas. Process mining, sequential pattern mining, and visual analytics often use different terminology for similar concepts. For example, what is referred to as a *case* in process mining may correspond to a *temporal event sequence* in event-sequence analysis.

To support the design process, a unified terminology was created with the purpose of bridging concepts across the fields used in the thesis and to establish a consistent vocabulary. The terminology was derived from relevant literature on process mining, visual analytics and event-sequence analysis [21, 16, 1, 3, 36, 24, 39]. The resulting terminology is shown in Figure 3.1. In addition to terminology, analytical tasks were also abstracted from the literature. These tasks describe what users need to do when exploring event-sequence patterns across multiple levels of granularity. The unified task abstraction is shown in Figure 3.2.

Unified term	**Process mining**	**Sequential pattern mining**	**Visual event sequence analytics**	**Description**
Event	Event	Itemset element	Data element	A single occurrence representing an action within a sequence, which can be modeled either as: **Instantaneous event**: associated with one timestamp **Interval event**: associated with a start and end timestamp, representing a duration
Event dataset	Event log	Sequence database	Dataset	Collection of event sequences used as input for analysis across different domains. Events may be represented either as point events (single timestamp) or interval events (start and end timestamps).
Event type	Activity	Label	Event type	A property used to distinguish and group events by their meaning or function in the sequence (e.g., event type such as “purchase,” “login,” or “error”). It describes what kind of event occurred rather than the individual event instance itself.
Event sequence	Case	Sequence	Temporal event sequence	Ordered collection of events associated with a single process instance or entity.
Event sequence	Trace	Sequence	Temporal event sequence	Temporally ordered list of events representing the progression of behaviour over time. Ordering may be based on timestamps for point events or on intervals for duration based events.
Unique event sequence	Variant	Unique sequence	Unique temporal sequence	Recurring structure or regularity identified within event sequences.
Model	Process model	Pattern	Visual representation	Abstract representation that summarizes or explains the structure of event sequences.
Tine	Timestamp	Sequential order	Temporal dimension	Temporal information associated with events, represented either as: **Single timestamp** (instantaneous event) **Time interval** with start and end timestamps (interval event)
Event granularity	Event abstraction level	Sequence abstraction level	Level of detail	The level of detail at which events are represented, from fine-grained specific actions (e.g., eat bread, drink milk) to coarse-grained abstract activities (e.g., have meal). This helps balance detail and interpretability.

Figure 3.1: Unified terms used to connect concepts from process mining, sequential pattern mining, and visual event sequence analytics.

Unified task	**Granularity**	**Process mining**	**Sequence analysis**	**Visual event sequence analysis**	**Description**
Explore data	Low/high	Log exploration	Sequence exploration	Exploration	Analysis of data is often iterative and usually begins with open-ended exploration of event data to identify interesting directions
Discover patterns	Low/high	Subtrace mining	Frequent sequence mining	Pattern detection	The goal is to identify recurring structures or frequent sequences within event data.
Compare sequences	Low/high	Conformance checking	Sequence alignment through edit distances	Sequence comparison	Sequences are compared to identify similarities, differences, or deviations in behaviour.
Detect anomalies	Low/high	Deviation detection	Rare pattern mining	Outlier detection	Analysts aim to detect unusual or infrequent behaviours that deviate from expected patterns.
Filter data	Low/high	Event/trace filtering	Constraint-based mining	Filtering	Data is filtered or constrained to focus on relevant subsets during analysis
Abstract patterns	High	Process abstraction	Pattern generalization	Summarization	Sequences are aggregated or generalized into higher-level representations to reduce complexity.
Interpret findings	Low	Insight generation	Pattern interpretation	Sensemaking	Identified patterns are interpreted and contextualized to derive meaningful insights.
Validate hypothesis	Low/high	Evidence checking	Pattern validation	Analytical resoning	Hypotheses are tested by iteratively searching for supporting or contradicting evidence in the data.

Figure 3.2: Unified task abstraction for exploratory multilevel event-sequence analysis.

This abstraction step was important because the prototype was not designed for one isolated field only. Instead, it combines ideas in the event-sequence analysis from process mining, sequential pattern mining, and visual analytics. The unified terminology and task abstraction therefore acted as a bridge between the theoretical background and the concrete design of the prototype.

3.2.3 Literature-informed Design

Following the initial problem characterisation, a broader literature review was conducted to identify existing visualisation and interaction techniques for event-sequence analysis. Literature searches were performed using keywords such as *event sequence*, *event abstraction*, *dynamic granularity*, *hierarchical aggregation*, *process mining*, and *sequential pattern mining*. The collected literature provided both theoretical grounding and practical design inspiration for the prototype.

Several survey papers on event-sequence visualisation and visual analytics influenced the design decisions of the system [1, 4]. Timeline-based representations were identified as one of the most intuitive ways to represent raw temporal event sequences, especially when preserving event order and duration information. Therefore, timeline-inspired sequence views were used in the prototype to show matched event traces and subsequences in relation to time.

The literature review also highlighted the usefulness of chart-based visualisations for representing event distributions. Although charts do not directly show temporal ordering, they can support exploratory analysis by showing the frequency of events or categories. Based on this, bar charts were introduced to provide overview information about event type frequencies at different granularity levels. These views were intended to guide users toward interesting event types and support filtering and drill-down decisions.

Previous work also distinguishes between **instance-based** and **model-based** representations of event-sequence data. Since the prototype focuses both on mined patterns and the underlying event traces, both representation types were used. The main pattern representation is model-based, since it presents pre-mined subsequences and relationships between event types. The linked timeline view is instance-based, since it shows the event sequences in which the selected pattern occurs.

The design of the system also follows Shneiderman's Information-Seeking Mantra: overview first, zoom and filter, then details-on-demand [15]. This principle was well suited for this context, since beginning with a highest-level overview allows patterns to be matched at the **coarsest level** of the semantic description, thereby capturing broader support in the data before more specific distinctions are introduced. This motivated the use of an overview representation of mined patterns, followed by interactions for selecting patterns or individual nodes to access more detailed information. Filtering was supported through constraints and through drill-down interactions, where users could select specific subcategories for further analysis. Finally, the linked timeline view presented the raw event sequences from which the patterns were derived.

3.3 Design requirements

Based on the literature review, analysis of related systems and expert input, a set of design requirements was derived to guide the design and implementation of the prototype. The requirements focus on supporting exploratory analysis of event-sequence patterns where users can move between abstraction levels, inspect the consequences of their choices, and relate mined patterns back to the underlying data.

- **R1 Support multi-level pattern exploration.** The system should allow event-sequence patterns to be explored at multiple levels of granularity, since meaningful behaviour may appear at different abstraction levels.

R2 **Enable local granularity adjustment.** Users should be able to expand or collapse selected activities or nodes without changing the level of detail for the entire dataset or visualisation.

R3 **Communicate hierarchical relationships.** The system should visually indicate how detailed activities relate to higher-level categories, since local granularity adjustment depends on an underlying activity hierarchy or ontology.

R4 **Show the consequences of interaction.** When users drill down or constrain a node, the system should communicate what information will be preserved, expanded, removed, or hidden before or during the interaction.

R5 **Link mined patterns to raw event traces.** Users should be able to inspect the original event sequences behind a mined pattern in order to interpret, contextualize, and validate the abstraction.

R6 **Support exploratory analysis.** The system should support questions that emerge during interaction rather than only validating predefined hypotheses. In particular, users should be able to explore how different granularity choices affect pattern support, such as how much support is lost when selecting a specific support threshold, choosing a subcategory, or replacing a higher-level activity with a more detailed path.

Chapter 4

Results

The prototype was designed as an interactive visual analytics system for exploring pre-mined event-sequence patterns. Pre-mined patterns were used because the focus was not on developing or optimising a new mining algorithm, but on investigating how already extracted patterns can be visually explored, interpreted, and refined. This made it possible to isolate the design problem of local granularity adjustment from computational performance issues, such as mining time, scalability, and algorithmic optimisation. It was developed to demonstrate how local granularity adjustment could be supported visually and interactively, by allowing users to start from high-level patterns with broad support and progressively refine selected parts into lower-level, more descriptive patterns. This solution is not intended as a replacement for existing process mining or sequential pattern mining tools, but rather to investigate a specific design approach for coarse-to-fine, multilevel sequence exploration. To reflect this focus, the system was named `SCOPE`, which stands for **S**plit-based **CO**arse-to-fine **P**rocess **E**xploration. In the remainder of this thesis, the prototype is referred to as `SCOPE`.

4.1 SCOPE overview

`SCOPE` combines an overview of mined patterns, which starts at the highest level of abstraction, with linked detail views, as shown in Figure 4.1. The pre-mined patterns used in the prototype were derived from [25]. They were used as a fixed input dataset in order to focus the thesis on interaction design rather than mining performance. The main view consists of a graph-based overview of the mined pattern space, where nodes represent event types and edges represent sequential relationships between them. Since the patterns are derived from multiple sequences, this view is model-based. The graph starts from a root node and branches into alternative patterns, sorted by frequency, allowing the user to inspect common behavioural structures and their support in the data. This overview supports **R1** by presenting patterns at a high abstraction level while still allowing them to be explored at more detailed levels through interactions.

The side-panel provides details about the currently selected pattern. The upper view shows the selected pattern, while the lower view provides a timeline-based representation of the event sequences. The legend on the left shows the event ontology and the colour encoding used for the event types. Together, these coordinated views link the abstract pattern representation to both the hierarchical event ontology and the underlying event sequences, allowing users to move between overview, distributional detail, and temporal context. The exploration starts from the graph displaying patterns at the highest level of abstraction. In this initial representation, the raw event types have been mapped to the highest level of the ontology, so each node represents a broad activity category. As illustrated in Figure 4.2, the legend can be expanded to reveal the different hierarchical levels to which each event type belongs. The ontology legend supports **R3** by making the hierarchical relationships between event types visible during exploration.

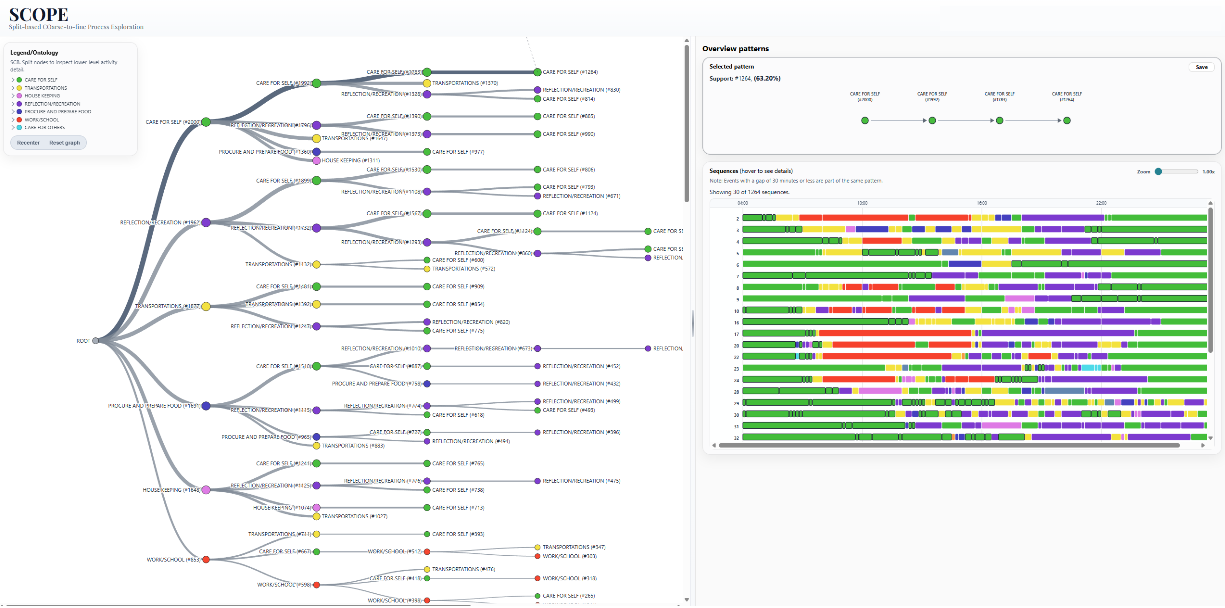


Figure 4.1: Overview of `SCOPE`. The left side shows the graph-based overview of the mined patterns, where darker edges indicate the currently selected pattern. The right side displays a simplified representation of the selected pattern together with a timeline-based view of the matching event sequences. In the timeline view, events that are part of the selected pattern are highlighted within their surrounding sequence context, making it possible to see where the pattern occurs in each sequence. The legend, top left, shows the event ontology and the colour encoding used for the event types.

4.1.1 Pop-up panel

The graph can be refined through the *Split* action, which allows users to inspect lower-level subcategories of a selected high-level node, as illustrated in Figure 4.2. These subcategories are shown together with frequency-based bar charts that indicate their relative contributions. This helps users evaluate whether a split is meaningful before applying it, for example by showing how much of the original support is retained by each more detailed subcategory. This addresses **R2** and **R6**, since users can locally refine a selected node and evaluate how the choice affects pattern support before committing to a split.

After inspecting the available subcategories, the user can choose whether to expand the selected node into a more detailed representation. This is done with the button *Split*, which will give a preview of how this node can be split into its subcategories and the relations with them, (see Figure 4.3). This preview allows the user to evaluate how the selected node would be replaced by its lower-level subcategories and how these subcategories connect to the following event types in the pattern. By displaying this, it can be explored how much support is lost when moving to a lower level representation. Two preview modes are provided, subtrees and compact. The *Subtrees* view shows each subcategory as a separate branch with its corresponding continuations. This supports detailed inspection of the alternatives created by the split. The *Compact* view instead

Figure 4.2: Drill-down exploration through the *Split* action. Selecting a high-level event type, such as *CARE FOR SELF*, opens a panel showing its lower-level subcategories and their local support. The expanded legend illustrates the hierarchical levels of the event ontology and the colour encoding used for the event types.

combines all the next-coming event, which gives a more condensed overview of the relationships between the split categories and the next event types. The split preview addresses **R4** by showing how the selected node would be expanded and how the surrounding pattern structure would change before the interaction is applied.

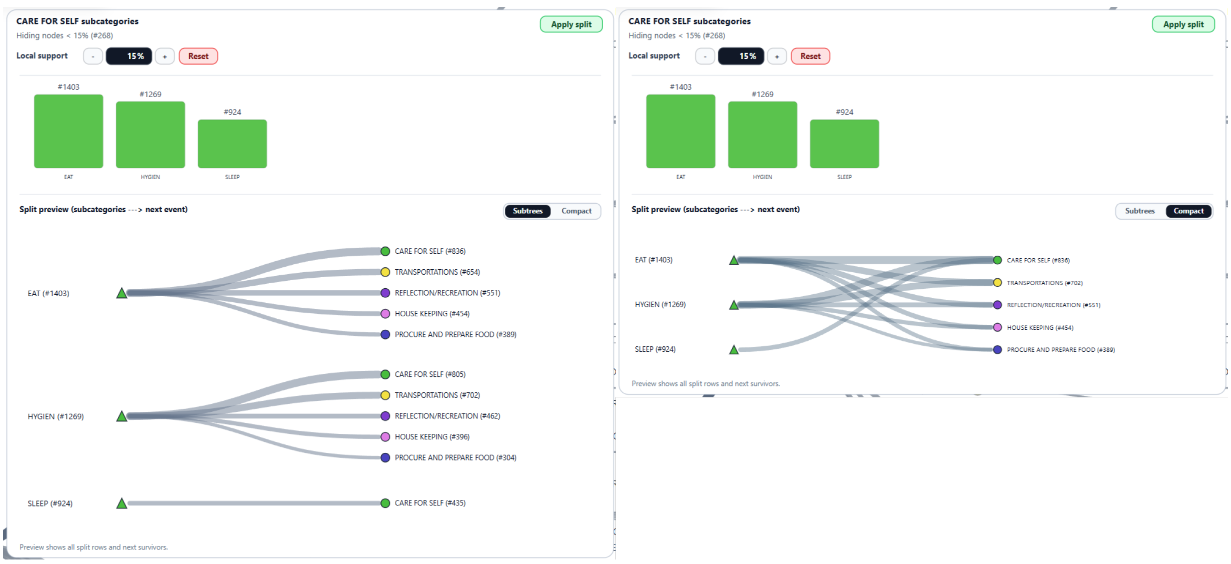


Figure 4.3: The two Split preview modes: subtree preview and compact preview.

By expanding only this node of the pattern, the user can investigate which detailed behaviours contribute to the high-level pattern without losing the broader sequential context. The user can then press *Apply Split* to expand the selected event type into a more detailed representation, causing the corresponding subcategories to be drawn in the graph. Because only the selected node is expanded, this supports **R2** by enabling local granularity adjustment.

4.1.2 Side panel

The side-panel provides detailed information about the currently selected pattern, as shown in Figure 4.1. In the upper part of the panel, the selected pattern is shown as a simplified sequence together with its support, allowing it to be inspected separately from the larger graph. The lower part of the panel contains a timeline-based view of the event sequences in which the selected pattern occurs. Because each row represents one matched sequence, this is an instance-based representation of the data. The coloured segments show the event types with duration encoded in width. The events that belong to the selected pattern are highlighted within their surrounding sequence context, making it possible to see where the pattern occurs in each matched sequence. Additional information, such as event type, start-time, end-time, and duration, is shown on hover. In this way, the timeline view links the abstract mined pattern back to the underlying event data, which addresses **R5**.

There is also a function to save the currently selected pattern in the side-panel, thus enabling for comparison between two or more patterns. In combination with the split-functionality it makes it possible to construct and compare mixed-granularity patterns, where some parts of the sequence remain on a coarse level while others are shown in greater detail (see Figure 4.4). By saving both **high-level** and **mixed-level** patterns, its possible to compare two or more patterns and examine how increased detail affects pattern support. This is also one way users can explore how much support is lost when moving from a broad, high-support representation to a more specific, lower-support pattern. This supports **R1** and **R6** by allowing users to compare patterns across different granularity levels and examine how increased detail affects support.

Figure 4.4: Saved patterns at different levels of granularity. The high-level pattern has broader support, while the more detailed variants have lower support due to selected event types being split into subcategories. This comparison helps users examine what is lost when moving from a coarse to a more specific pattern representation.

4.1.3 Constraints-handling

Constraints can be added by specifying a threshold value that determines which nodes and connections should remain visible. This supports **R6** by allowing users to explore how different support thresholds affect the visible pattern space. Rather than removing elements immediately, SCOPE first shows affected parts of the graph in a greyed-out state, providing a preview of the elements that would be hidden if the constraint were applied, which addresses **R4**.

A constraint is inherited from the selected parent node, meaning that it affects the active node and its subsequent nodes in the graph. When a constraint is applied to the root node, it affects the entire graph and therefore acts as a global constraint. This is illustrated in Figure 4.5, where the left example shows a constraint applied to the root node and the right example shows a constraint applied to another general node. Since constraints can affect either the whole graph or a selected part of it, this functionality relates to both **R1** and **R2**.

The constraint threshold is also reflected in the bar-charts and split preview, as shown in Figure 4.6, where lower-level subcategories below the selected support level are shown as inactive. The figure also shows both split preview modes: the subtree view on the left and the compact view on the right. Reflecting the constraint in the bar charts and split preview also supports **R4** and **R6**, since users can inspect the consequences of threshold choices before applying them.

Figure 4.5: Constraint preview in the graph. Nodes and connections that fall below the specified threshold are shown in a grayed-out state, indicating which parts of the graph would be hidden if the constraint was applied. The left example shows a constraint applied to the root node, affecting the whole graph, while the right example shows a constraint applied to a general node, affecting only the corresponding part of the graph.

4.2 SCOPE workflow

This section describes a typical exploration workflow in SCOPE. The purpose is to illustrate how a user can move from a high-level mined pattern to more detailed representations by selectively refining individual activities. Rather than presenting a dataset-specific demonstration, the workflow describes the main interaction sequence supported by the prototype.

A typical workflow starts with the user inspecting the graph-based overview of mined patterns at the highest level of abstraction. The user can then select an event type in the pattern to inspect its lower-level subcategories. The frequency-based bar chart shows how often each subcategory contributes to the selected high-level event type, while the timeline view shows the raw event sequences in which the selected pattern occurs. Based on this information, the user may choose to expand the selected event type into a lower granularity level.

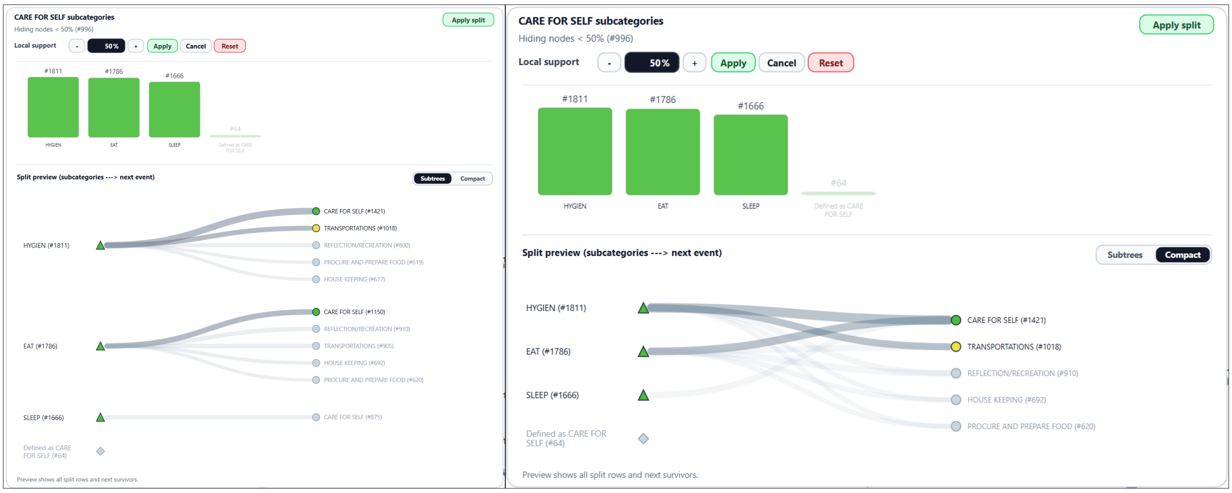


Figure 4.6: Constraint threshold reflected in the bar-charts and split preview. Lower-level subcategories that fall below the selected support threshold are shown as inactive. The left example shows the subtree preview mode, where each subcategory is shown separately, while the right example shows the compact preview mode, where the outgoing connections are aggregated.

After the expansion, the pattern becomes more descriptive but may also become less frequent, since a more detailed pattern usually matches fewer sequences. This trade-off between support and descriptive detail is central to SCOPE. The system allows users to explore this trade-off interactively by adjusting the granularity of selected nodes and observing how the pattern, its support, and its matched sequences change.

This workflow was used both during development and evaluation. During development, it helped assess whether the interaction flow was understandable and whether the visual components supported the intended analytical tasks. During evaluation, it provided a concrete basis for expert feedback on the prototype.

Chapter 5

Evaluation

The evaluation aimed to assess whether or not SCOPE supports an understandable way of exploring event-sequence patterns at multiple levels of event granularity. Since the prototype focuses on a specialised visual analytics problem, the evaluation was based on **qualitative expert feedback** rather than a controlled user study. This follows common practice in visualisation design studies, where iterative development and expert feedback are used to assess the usefulness, relevance, and validity of a proposed visual analytics approach [38, 40]. A qualitative expert evaluation was chosen because the prototype targets an exploratory and specialised visual analytics workflow, where expert interpretation, perceived usefulness, and conceptual clarity are more relevant than task-completion time alone. A controlled user study would have required a more mature prototype and a larger participant group, which was outside the scope of this thesis.

5.1 Method

The evaluation consisted of two parts. First, the prototype was evaluated iteratively throughout the development process during weekly meetings with a supervisor with expertise in visual analytics. These meetings served as formative evaluations, where design decisions, interaction techniques, and visual encodings were discussed and refined continuously. This iterative process is consistent with design study methodology, where visual analytics systems are developed and evaluated in close collaboration with domain and visualisation experts [38].

Second, qualitative feedback sessions were conducted with domain experts. Two experts had experience in process mining, and one expert had experience in **visual event sequence analytics (VESA)** using SPM. The sessions were conducted remotely, with one hour allocated for each session. The presentation used during the feedback sessions is included in Appendix A. Each session began with a short introduction to the thesis and the problem of event granularity. The experts were then asked about their current workflows, how they handle different levels of event granularity, and what challenges they experience when comparing patterns at different levels of abstraction.

After this, SCOPE was demonstrated in a live-demo. The demonstration covered all the functionalities described in Section 4.1. The experts were asked to comment on the understandability of the concept, the usefulness of local granularity adjustment, the clarity of the visual feedback, and the relevance of the workflow to their own analytical practice. These sessions were recorded and transcribed, with the consent of the participants.

5.2 Results

As the session consisted of two parts, the results are divided into a workflow and granularity challenges-section and a `SCOPE` feedback-section. The workflow section served as a recap for the experts, enabling them to more easily connect their granularity-related challenges to the proposed solution presented. It was further confirmed that the granularity problem addressed by `SCOPE` is relevant in current event-sequence analysis workflows.

5.2.1 Workflow and granularity challenges

The PM experts described event granularity as something that is typically handled **before** the data is imported into an analysis tool. One PM expert explained that "the way this is done is usually outside a process mining tool", and further noted that **changing** the granularity level often requires taking the log out of the process mining tool, applying preprocessing or abstraction steps, and then importing the modified log again. The other PM expert described the current state as having "no tool support" for handling granularity directly in the visualisation. Existing level-of-detail mechanisms were also described as mainly global, with one PM expert noting that "the granularity is only handled globally", and that it does not allow for users to zoom in on specific parts of a process. In the question for how to compare coarse vs fine patterns they said "no tool support" and then added "you can make a screenshot". Another aspect concerned the semantic nature of abstraction, where one expert emphasized that meaningful comparison is not only about changing the level of detail, but about choosing an abstraction level that preserves what is relevant for the analysis.

From the VESA & SPM perspective, granularity was described as a multi-level issue, where the low-level event data should be preserved while higher-level abstractions are created. They emphasized that they "always start with the raw data" and aim to "never lose the low level", but instead keep it linked to higher-level abstractions. The expert noted that finding the "best abstraction" is difficult, since there is no "one fits all approach", and mentioned challenges such as large data volumes, diverse attributes, missing data, and overly detailed event logs. For comparing coarse and fine-grained patterns, the expert described moving between overview and detail as a "deep dive". They emphasized the need for **coordinated and linked views**, where different levels of granularity can be seen at the same time in order to preserve context. The expert also noted that both frequent patterns and rare or anomalous situations are relevant, although infrequent patterns are harder to interpret because they are often numerous and diverse.

5.2.2 SCOPE feedback

The feedback session focused on how the experts perceived the main functionalities of `SCOPE`, as well as how these supported or limited the exploration of granularity in process data. Overall, several features were received positively, particularly those that made the consequences of granularity changes more explicit. At the same time, there were discussions that highlighted limitations in the proposed system.

- **Split preview:** The split preview was received positively, especially because it communicates the consequences of a granularity change before the graph is updated. The VESA expert commented that "the kind of preview that you get before applying the split... I like that very much". The two different preview modes, compact and subtrees, were also seen as complementary. The compact view was considered useful for gaining a higher-level overview, while the subtree view supported "the more detailed breakdown". One PM expert also stated "I like that you anticipate the effects of the filters".

- **Support-threshold preview:** The support-threshold preview was considered useful for understanding the trade-off between detail and support. During the discussion, the interaction was described as a way to show "how much information will get lost" when increasing the support threshold while moving to a more detailed level. The VESA expert responded positively to this, stating that "this is really great".

  One PM expert noted that currently `SCOPE` only uses a frequency threshold to filter activities, but that abstraction decisions should also be guided by semantic relevance, domain meaning, and the analyst's goals. For example, an analyst may want to preserve an activity such as "sleep" even if it is not among the most frequent activities.

- **Graph structure:** The graph structure was difficult for the PM experts to interpret, as it did not align with the usual "conceptual model" in process mining. Since the root of the graph does not represent the start of a process, one PM expert found it difficult to mentally map the visualisation to their typical way of analysing processes from start to end. It was explained as "the root here, it's the root of the pattern tree and it doesn't have a meaning".

- **Timelines:** The timelines were especially appreciated by one PM expert, who stated that "I do like that you can see where the pattern is actually occurring". The expert also highlighted that temporal information is important and frequently used in process mining. The VESA expert also responded positively to the timelines, explaining that "I need the coordinated views linked so that I can see both things at the same time". The highlighting of events in the timeline was also received positively, as the expert noted that "we are doing something similar", referring to a project they were currently working on.

- **Role in the PM workflow:** One PM expert described the system as feeling more like a preprocessing stage in process mining rather than an exploratory live-analysis tool. This was motivated by the fact that "pre-processing is anything you would do to make your data fit for process mining". At the same time, the expert noted that when abstractions involve domain knowledge and are related to the question the analyst wants to answer, "it's already an abstraction step you're doing for the analysis". Further it was noted that some abstraction decisions may belong within the analysis itself, even if they are currently treated as preprocessing because such interactions are not supported as analysis steps in existing PM tools.

- **Role in the VESA & SPM workflow:** The VESA expert expressed that they wanted to "play with it" referring to the system. They had a hard time evaluating if this would be useful in their area as they could not interact with the system on their own. It was further noted some limitations in the interaction flow. The expert noted that the system required "too many clicks" and that, when first encountering the interface, it was difficult to know "what do I do next?".

- **Other comments:**

  One PM expert noted that the "principles are really, really strong", referring to the underlying idea of the system and its potential to support more flexible exploration of granularity in process data.

The VESA expert sometimes found the splitting interaction difficult to follow, noting that the live-demo moved quickly and that it was not always clear where the focus was or what to look at.

It was also noted by the sequential pattern mining expert that parallels could be seen between `SCOPE` and their own ongoing work, particularly regarding the use of coordinated views, selective hiding/filtering strategies, and contextual highlighting in the timelines.

The experts within both areas had difficulties understanding the navigation and maintaining a clear mental model of the system. However, the source of confusion differed. The PM experts mainly struggled with conceptual interpretation, while the VESA expert had more confusion related to the interaction flow and knowing where to focus or what step to take next.

Chapter 6

Discussion

This chapter discusses the main conceptual and practical insights gained from the evaluation of `SCOPE`. The discussion focuses on the differences between visual event sequence mining and process mining, the role of abstraction in the analysis workflow, and possible directions for future work. Since `SCOPE` is designed for multi-level exploration, it assumes that the event-sequence data contains an underlying ontology or hierarchy that describes relationships between event types at different levels of granularity. Therefore, the system is most useful for datasets where such hierarchical structures are present. For event sequences without meaningful hierarchy, the benefits of `SCOPE` would be limited, since there would be no alternative levels of event granularity to explore.

6.1 Abstraction as preprocessing or live analysis

An important discussion concerned whether `SCOPE` should be understood as a preprocessing tool or as part of the live analysis process. One PM expert described the system as feeling closer to preprocessing, since abstraction is often something done to make data suitable for process mining. However, the discussion also highlighted that abstraction decisions are not always purely technical preprocessing steps. When abstraction is guided by domain knowledge, analyst goals, or a specific analysis question, it becomes closely connected to the analysis itself. This suggests that dynamic abstraction during analysis, as `SCOPE` does, could be an important future direction, especially for systems where analysts need to move between different levels of event granularity while exploring the data.

The evaluation also showed that frequency alone is not sufficient for deciding which activities should be abstracted, filtered, or preserved. Currently, `SCOPE` mainly uses frequency-based thresholds, but the experts emphasized that semantic meaning, domain relevance, and analyst intent should also influence abstraction decisions. For example, an activity may be important to preserve even if it is not frequent, because it has a specific meaning in the domain or is relevant to the analyst's current question. Future work should therefore explore ways of making this possibility more explicit in the interaction. This is partly supported by the current split functionality in `SCOPE`. Although frequency is used to sort and present possible splits, the user can still choose to follow a lower-frequency path if it is semantically relevant. However, this possibility may not have been communicated clearly enough during the demo, which could have contributed to the impression that the system was mainly frequency-driven.

6.2 PM and VESA

A central finding from the evaluation was the conceptual gap between the sequence mining perspective used in `SCOPE` and the process mining perspective of the PM experts. The PM experts primarily interpreted the system in relation to directly-follows graphs and event abstraction workflows. Their discussions therefore focused on aspects such as semantic abstraction, preprocessing versus analysis, process semantics, and the temporal interpretation of the tree structure.

In particular, the tree-based structure did not align with the experts mental model of process models, especially Directly-Follows Graphs (DFGs), where processes are typically interpreted from start to end. Since the root of the tree in `SCOPE` does not represent the start of a process, it was difficult for the PM experts to transfer their existing way of reasoning to the system. This created some conceptual friction between traditional process models and the sequence-oriented pattern tree representation used in `SCOPE`.

In contrast, the VESA expert more naturally interpreted the system as an exploratory tool for multilevel sequence analysis. The feedback from this perspective focused more on coordinated views, maintaining access to low-level data, contextual drill-down, and linked exploration across multiple levels of granularity. This suggests that the interaction model was more immediately compatible with existing exploratory practices in the VESA area.

Overall, this difference indicates that future versions of the system may need to better bridge the gap between process mining and sequential pattern mining perspectives. One possible direction is to adapt the interaction concept to DFGs, or to explore how similar abstractions and drill-down interactions could be applied directly to raw event sequences before a process model is generated.

6.3 Reflection on rapid prototyping

The development process also highlighted the value of rapid software prototyping, as described briefly in Section 3.1. Since design studies are iterative by nature, different ideas are often tested, improved, or discarded if they do not work well enough. This was relevant in this thesis, where several alternative designs for handling granularity, splitting, and drill-down interactions were explored before arriving at the final demo.

AI-based coding assistance was used as a support during this prototyping phase, primarily to generate and revise small code fragments for visual concepts and interaction behaviours. This made it possible to more quickly implement throw-away prototypes and test whether certain ideas were useful, understandable, or realistic for representing the data. The AI support was therefore used as an implementation aid during early concept development, while the design decisions, evaluation focus, and final selection of features were made by the author. Rather than spending too much time polishing early implementations, several exploratory prototypes could be created and compared, with only the most relevant parts carried forward into the final system.

6.4 Validity, reliability and ethical considerations

The evaluation is limited by the small number of experts, the remote demo format, and the use of one pre-mined dataset. Therefore, the results should be interpreted as qualitative design feedback rather than generalisable evidence of usability or performance. The remote format may also have made the interaction flow harder to follow, especially since the experts did not freely interact with the prototype themselves. Future evaluations should include hands-on tasks, more participants, and datasets from different domains. Ethical considerations mainly concern the origin and sensitivity of the event-sequence data. In this thesis, the prototype was evaluated on pre-mined patterns rather than on live user data; nevertheless, future versions should ensure that event logs are anonymised and that visualised patterns are not interpreted as complete or causal explanations of behaviour.

6.5 Future work

Future work could include an integration of `SCOPE` in `ELOQUENCE`, thereby providing an abstraction layer in the existing exploratory sequence analysis workflow. As `ELOQUENCE` is already a powerful tool for exploratory analysis, this addition could guide users in navigating between different levels of granularity.

During the evaluation sessions, it was also suggested that control-flow could be complemented with additional perspectives, such as metadata, context, or other attributes connected to the sequences in the timeline-view. This could help analysts understand not only which patterns occur, but also under what conditions they occur.

Another promising direction is to relate the concept of `SCOPE` to object-centric process mining [41], knowledge graphs, and dynamic modelling. These approaches often involve both high-level and low-level nodes, making them relevant for exploring abstraction across multiple levels of granularity. The idea of dynamically moving between abstraction levels during analysis therefore appears to fit well with future directions in process mining.

Chapter 7

Conclusion

This thesis addressed the lack of interactive visual analytics support for exploring event-sequence patterns across multiple levels of event granularity. The work was motivated by the observation that existing event-sequence analytics workflows often require the level of abstraction to be decided before analysis, which limits an analysts ability to explore how patterns change when moving between coarse and fine-grained event descriptions. To address this, the thesis designed, implemented, and evaluated `SCOPE`.

The research questions were answered through a combination of literature review, analysis of related systems, prototype design, and qualitative expert evaluation. The theoretical background and related work in Chapter 2 addressed how event granularity and hierarchy are currently handled in event-sequence analytics. The design process and implementation described in Section 3.3 and Chapter 4 addressed how local granularity adjustment can be supported interactively. Finally, the evaluation and discussion in Chapters 5 and 6 showed how experts understood the proposed workflow, what parts of the system were useful, and what limitations remain.

7.1 Completion of objectives

1. **Survey existing event sequence analytics literature to identify how event granularity and hierarchy are currently modelled. (RQ1)**

   This objective was addressed in Chapter 2. The literature showed that event granularity is commonly handled through abstraction, aggregation, ontologies, or hierarchical activity structures, but that these choices are usually made before or outside the main analysis workflow.

2. **Analyse strengths and limitations of current multi-level event mining approaches with respect to flexibility and scalability. (RQ1, RQ2)**

   This objective was addressed through the review of related systems in Section 2.7, as well as the expert input.

3. **Investigate existing algorithms and system designs that support incremental or interactive adjustment of event granularity during mining. (RQ2)**

   This objective was addressed in Sections 2.5, 2.6, and 2.7. Progressive visual analytics, progressive data analysis, and systems such as ELOQUENCE provided a foundation for understanding how users can interactively steer analysis.

4. **Survey existing visualisation techniques for multilevel temporal and sequential data analysis. (RQ1, RQ3)**

   This objective was addressed in the visualisation and interaction design section of Chapter 2. These techniques informed the choice of coordinated graph, bar-chart, ontology, and timeline views in `SCOPE`.

5. **Define requirements for dynamic granularity control from an analyst's exploratory workflow perspective. (RQ2, RQ3)**

   This objective was addressed in Section 3.3. The design requirements were derived from literature, related work, and expert input.

6. **Identify interaction techniques that allow analysts to navigate, filter, and compare patterns across granularities. (RQ3)**

   This objective was addressed through both the literature review and the prototype design.

7. **Design and implement a visual analytics prototype for exploring mined patterns across multiple event levels. (RQ2, RQ3)**

   `SCOPE` was implemented as a visual analytics prototype that combines a graph-based overview of mined patterns with linked detail views, a hierarchical ontology legend, split-based local refinement, support-threshold constraints, saved pattern comparison, and timeline-based inspection of matching event sequences.

8. **Evaluate how analysts understand and interpret changes in event granularity through interactive visualisations. (RQ2, RQ3)**

   The evaluation was done both iteratively and with a final expert user evaluation, described in Chapter 5. The evaluation showed that previews and linked views were useful, but that navigation and mental-model support need further refinement.

7.2 Completion of research questions

RQ1 How is multi-level mining of events handled today in event sequence analytics approaches?

Multi-level mining of events is currently handled mainly through preprocessing, abstraction, filtering, hierarchical aggregation, or system-specific representations of event sequences. In process mining, the event log is often transformed before it is imported into an analysis tool. This means that the analyst usually chooses one level of granularity before the actual analysis begins. If another level of detail is needed, the data often has to be transformed again outside the tool and then re-imported. In visual event-sequence analytics based on sequential pattern mining, some systems support exploration across multiple levels of detail, but the abstraction is often applied globally or at the level of complete patterns, clusters, cohorts, or representations.

The literature review and related work therefore show that existing systems provide partial support for multi-level analysis, but limited support for local, interactive, mixed-granularity exploration within individual event-sequence patterns. This was also confirmed by the expert feedback, where process mining experts described granularity as something typically handled before analysis rather than as part of an interactive workflow.

RQ2 How can event granularity be interactively and locally adjusted for individual nodes during the mining process, while ensuring that these changes are clearly communicated to the user?

This thesis shows that event granularity can be locally adjusted by allowing analysts to select individual nodes in a mined pattern and split them into lower-level subcategories based on an underlying event hierarchy or ontology. In `SCOPE`, this is supported through the *Split* interaction, where a selected high-level node can be expanded into more detailed event types while the surrounding pattern remains at a coarser level. This enables mixed-granularity patterns, where only analytically relevant parts of the sequence are refined.

To communicate these changes clearly, `SCOPE` uses several forms of visual feedback. Before a split is applied, the system shows a preview of the possible lower-level subcategories and their support. Two preview modes are provided, a subtree view for detailed inspection and a compact view for a more condensed overview. Support-threshold previews also show which nodes and connections would be hidden before the constraint is applied. These previews help the analyst understand what information will be preserved, expanded, filtered, or lost when moving to a more detailed level of granularity.

The evaluation showed that this approach was understandable and useful in principle, especially because experts appreciated being able to preview the consequences of a split or threshold before the graph was updated. However, the evaluation also showed that the interaction flow needs further refinement, since some users found it difficult to know where to focus or what action to take next.

RQ3 How can visualisations enable analysts to interactively explore and compare patterns across different levels of event granularity?

Visualisations can support exploration across granularities by combining overview, detail, hierarchy, comparison, and raw-data context in a coordinated workflow. In `SCOPE`, the graph-based overview gives access to mined patterns at a high abstraction level, while the ontology legend communicates the hierarchical structure behind the event types. The split panel and bar charts show the distribution of lower-level subcategories, helping analysts decide whether a node should be refined. The timeline view links the abstract pattern back to the original event sequences, making it possible to see where the selected pattern occurs and how it relates to the surrounding sequence context.

`SCOPE` also supports comparison across granularities through saved patterns. Analysts can save a high-level pattern and compare it with more detailed mixed-granularity variants. This makes it possible to examine how increased detail affects pattern support, and to reason about the trade-off between broad, high-support patterns and more specific, lower-support patterns. The evaluation showed that the timeline view, coordinated views, and previews were especially valuable for helping users interpret pattern changes.

At the same time, the evaluation showed that visualisations must align with the analysts mental model. The process mining experts found the tree-based pattern structure less intuitive because it did not correspond to a conventional process model such as a directly-follows graph. This suggests that future systems should further investigate how local granularity adjustment can be integrated into representations that are already familiar to process mining analysts.

7.3 Final remarks

The thesis demonstrates that local and interactive granularity adjustment is a relevant direction for event-sequence analytics. `SCOPE` shows how analysts can move from coarse to fine-grained pattern representations without losing the broader sequential context. The evaluation indicates that previews, coordinated views, timelines, and saved pattern comparison can help users understand the consequences of granularity changes.

However, the work also shows that dynamic granularity control is not only a technical or visual problem. It is also a conceptual challenge, especially when bridging process mining and sequential pattern mining perspectives. Future work should therefore investigate how the proposed interaction principles can be integrated into more familiar process mining representations, how semantic relevance can be combined with frequency-based filtering, and how the interaction flow can be improved to better guide analysts during exploration.

Bibliography

[1] A. Yeshchenko and J. Mendling, "A survey of approaches for event sequence analysis and visualization," *Information Systems*, vol. 120, p. 102283, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437923001199

[2] W. Van Der Aalst, "Process mining," *Communications of the ACM*, vol. 55, no. 8, pp. 76–83, 2012.

[3] F. Zerbato, R. Seiger, G. Di Federico, A. Burattin, and B. Weber, "Granularity in process mining: Can we fix it?" in *PROBLEMS 2021: BPM Problems to Solve Before We Die*, vol. 2938. CEUR Workshop Proceedings, 2021, pp. 40–44.

[4] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao, "Survey on visual analysis of event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 5091–5112, 2022.

[5] S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider, "Event abstraction in process mining: literature review and taxonomy," *Granular Computing*, vol. 6, no. 3, pp. 719–736, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s41066-020-00226-2

[6] X. Lu, A. Gal, and H. A. Reijers, "Discovering hierarchical processes using flexible activity trees for event abstraction," in *2020 2nd International Conference on Process Mining (ICPM)*, 2020, pp. 145–152.

[7] W. Wang and J. Yang, *Mining Sequential Patterns from Large Data Sets*, 1st ed., ser. Advances in Database Systems. New York, NY: Springer, 2005.

[8] C. H. Mooney and J. F. Roddick, "Sequential pattern mining – approaches and algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, Mar. 2013. [Online]. Available: https://doi.org/10.1145/2431211.2431218

[9] S. van der Linden, B. Cappers, A. Vilanova, and S. van den Elzen, "Long sequences with a lot of events (lolo): A visual analytics approach for analyzing long event sequences," *Information Visualization*, vol. 25, no. 2, pp. 121–138, 2026. [Online]. Available: https://doi.org/10.1177/14738716251372584

[10] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, *Visual Analytics: Definition, Process, and Challenges*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 154–175. [Online]. Available: https://doi.org/10.1007/978-3-540-70956-5_7

[11] S. van der Linden, E. de Fouw, S. van den Elzen, and A. Vilanova, "A survey of visualization techniques for comparing event sequences," *Computers Graphics*, vol. 115,

pp. 522–542, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849323000717

[12] S. van den Elzen, M. Jans, N. Martin, F. Pieters, C. Tominski, M.-C. Villa-Uriol, and S. J. van Zelst, "Towards multi-faceted visual process analytics," *Information Systems*, vol. 133, p. 102560, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437925000444

[13] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer, "Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 6, pp. 1636–1649, 2017.

[14] D. Schuster, S. J. van Zelst, and W. M. P. van der Aalst, "Incremental discovery of hierarchical process models," in *Research Challenges in Information Science*, F. Dalpiaz, J. Zdravkovic, and P. Loucopoulos, Eds. Cham: Springer International Publishing, 2020, pp. 417–433.

[15] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The Craft of Information Visualization*, ser. Interactive Technologies, B. B. BEDERSON and B. SHNEIDERMAN, Eds. San Francisco: Morgan Kaufmann, 2003, pp. 364–371. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558609150500469

[16] S. Miksch, "Visual analytics meets process mining: Challenges and opportunities," in *2021 3rd International Conference on Process Mining (ICPM)*, 2021, pp. xiv–xiv.

[17] F. Zerbato, P. Soffer, and B. Weber, "Initial insights into exploratory process mining practices," in *Business Process Management Forum*, A. Polyvyanyy, M. T. Wynn, A. Van Looy, and M. Reichert, Eds. Cham: Springer International Publishing, 2021, pp. 145–161.

[18] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, 2007.

[19] S. Knoblich, J. Mendling, and H. Jambor, "Review of Visual Encodings in Common Process Mining Tools," in *VIPRA 2024 - Visual Process Analytics Workshop*, A. Arleo, S. van den Elzen, T. von Landesberger, J.-R. Rehse, L. Pufahl, and F. Zerbato, Eds. The Eurographics Association, 2024.

[20] A. Berti, S. van Zelst, and D. Schuster, "Pm4py: A process mining library for python," *Software Impacts*, vol. 17, p. 100556, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665963823000933

[21] F. Zerbato, L. Zimmermann, K. Vrotsou, and B. Weber, "From analysis to findings: How do process mining analysts discover results?" *Information Systems*, vol. 135, p. 102596, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437925000808

[22] W. van der Aalst, A. Adriansyah, A. K. A. de Medeiros *et al.*, "Process mining manifesto," in *Business Process Management Workshops*, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Springer, 2012, pp. 169–194.

[23] D. Schuster, S. J. van Zelst, and W. M. van der Aalst, “Utilizing domain knowledge in data-driven process discovery: A literature review,” *Computers in Industry*, vol. 137, p. 103612, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522000070

[24] A. Bechini, A. Bondielli, P. Dell’Oglio, and F. Marcelloni, “From basic approaches to novel challenges and applications in sequential pattern mining,” *Applied Computing and Intelligence*, vol. 3, no. 1, pp. 44–78, 2023. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/aci.2023004

[25] K. Vrotsou and A. Nordman, “Exploratory visual sequence mining based on pattern-growth,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2597–2610, 2019.

[26] A. Perer and F. Wang, “Frequence: interactive mining and visualization of temporal frequent event sequences,” in *Proceedings of the 19th International Conference on Intelligent User Interfaces*, ser. IUI ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 153–162. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/2557500.2557508

[27] J. Magallanes, T. Stone, P. D. Morris, S. Mason, S. Wood, and M. Villa-Uriol, “Sequen-c: A multilevel overview of temporal event sequences,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 901–911, 2022.

[28] D. Schuster, W. Aigner, C. Di Francescomarino, C. Turkay, and F. Zerbato, “Kava-pm: Knowledge-assisted visual process mining,” *Information Systems*, vol. 137, p. 102638, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437925001243

[29] C. D. Stolper, A. Perer, and D. Gotz, “Progressive visual analytics: User-driven visual exploration of in-progress analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1653–1662, 2014.

[30] J.-D. Fekete, D. Fisher, and M. Sedlmair, “Progressive data analysis.” Eurographics, 2024.

[31] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, “Characterizing guidance in visual analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 111–120, 2017.

[32] K. T. Zinat, S. N. Sakhamuri, A. S. Chen, and Z. Liu, “A multi-level task framework for event sequence analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 1, pp. 842–852, 2025.

[33] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, “Eventthread: Visual summarization and stage analysis of event sequence data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 56–65, 2018.

[34] F. van Ham and A. Perer, “Search, show context, expand on demand: Supporting large graph exploration with degree-of-interest,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 953–960, 2009.

[35] T. Fehrer, L. Moder, and M. Röglinger, “An interactive approach for group-based event log exploration,” *Information Systems*, vol. 134, p. 102575, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437925000596

[36] L. Montana, J. Magallanes, M. Juarez, S. Mason, A. Narracott, L. van Gemeren, S. Wood, and M.-C. Villa-Uriol, “Eventbox: A novel visual encoding for interactive analysis of temporal and multivariate attributes in event sequences,” 2025. [Online]. Available: https://arxiv.org/abs/2507.14685

[37] D. Schuster, S. J. van Zelst, and W. M. van der Aalst, “Cortado: A dedicated process mining tool for interactive process discovery,” *SoftwareX*, vol. 22, p. 101373, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711023000699

[38] M. Sedlmair, M. Meyer, and T. Munzner, “Design study methodology: Reflections from the trenches and the stacks,” *IEEE transactions on visualization and computer graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[39] M.-C. Villa-Uriol, “Addressing directly-follows graphs limitations with visualization of event sequences.”

[40] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, “Empirical studies in information visualization: Seven scenarios,” *IEEE transactions on visualization and computer graphics*, vol. 18, no. 9, pp. 1520–1536, 2011.

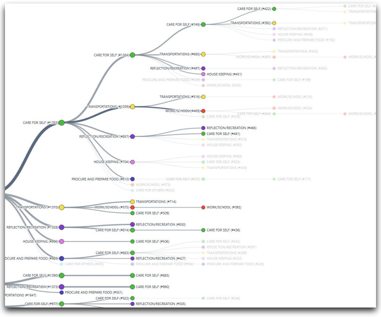
[41] S. Khayatbashi, N. Miri, and A. Jalali, *OLAP Operations for Object-Centric Process Mining*, 06 2025, pp. 111–118.

Appendix A

Evaluation presentation

Is it ok to record/transcript?

Interactive visual mining of event sequence patterns at multiple levels of event granularity



Caitlin Wu

My thesis

- Aim:
  The aim of this thesis is to design, implement, and evaluate an **interactive visual analytics** system that supports the discovery and exploration of event sequence patterns at **multiple levels of granularity**, allowing analysts to dynamically adjust abstraction levels during the mining process.

- Research questions

  1. How is multi-level mining of events handled today in event sequence analytics approaches?

  2. How can event granularity be interactively and locally adjusted for individual nodes during the mining process, while ensuring that these changes are clearly communicated to the user?

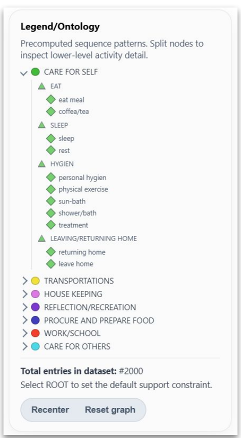
  3. How can visualizations enable analysts to interactively explore and compare patterns across different levels of event granularity?

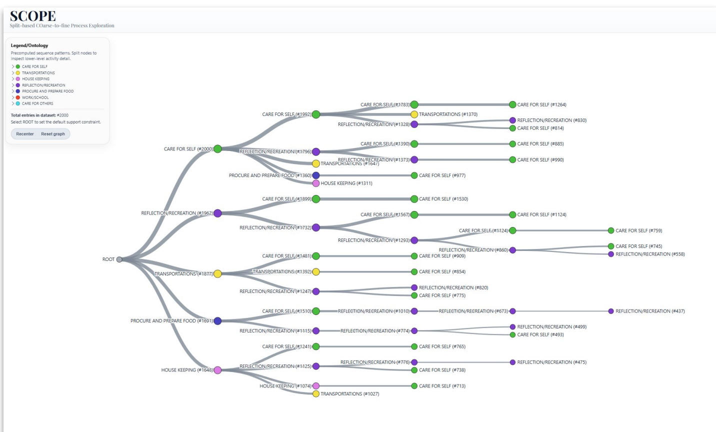
Your workflow

1. How are you handling granularity in event logs today?
2. What problems do you come across?
3. How is a coarse pattern VS fine-grained pattern compared?
   What tools, views. etc.
4. What are you usually looking for in sequences?
5. When do you usually zoom in/out on the data?

Short overview of **SCOPE**

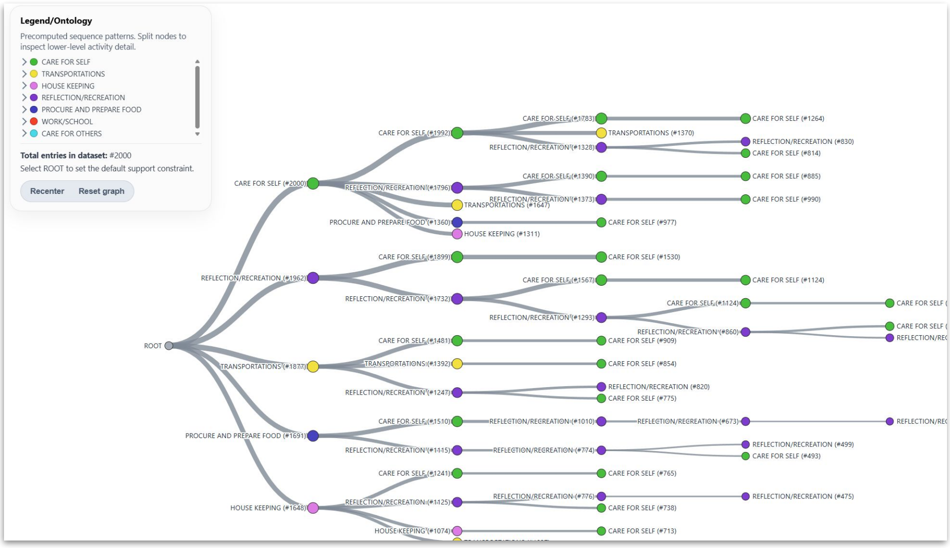
- **SCOPE** - **S**plit-based **CO**arse-to-fine **P**rocess **E**xploration
- Builds a graph of **precomputed patterns** at the highest level, exported from ELOQUENCE
- All raw event data are abstracted into the **highest level in the ontology,** in this way capturing more support of the data for each pattern.





Short overview of **SCOPE**

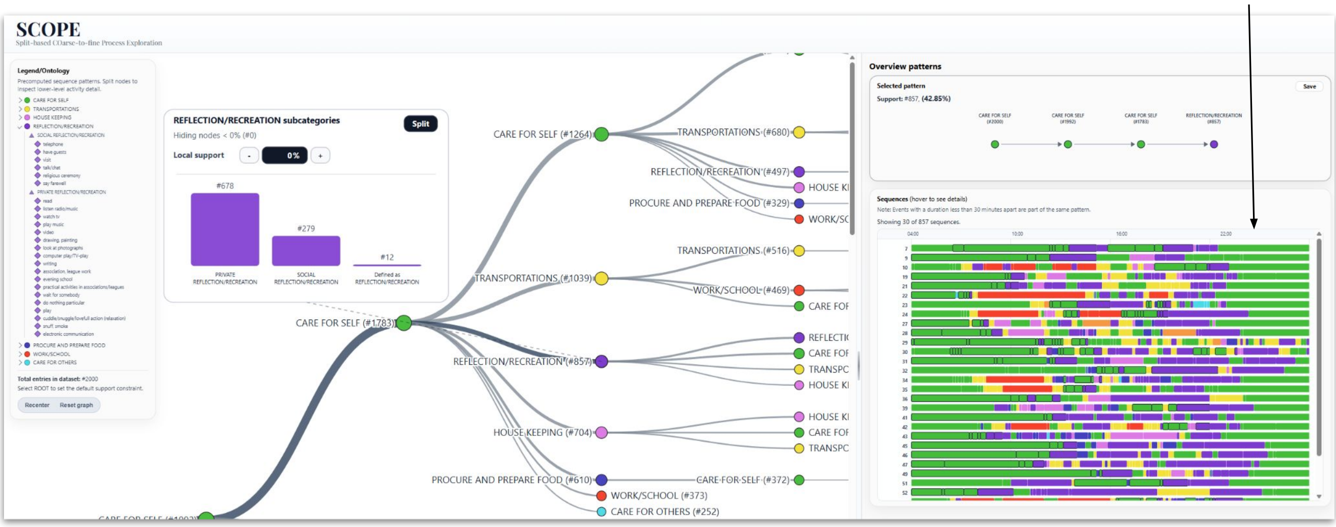
- The # indicates the **amount of sequences** that contain this pattern



Short overview of **SCOPE**

- Selection of a node will:
  - Show subcategories (lower levels) with barcharts that reveal the **frequency distribution**
  - Highlight the trace **(selected sub-pattern)**
  - Show timelines (raw data) in side panel for selected trace

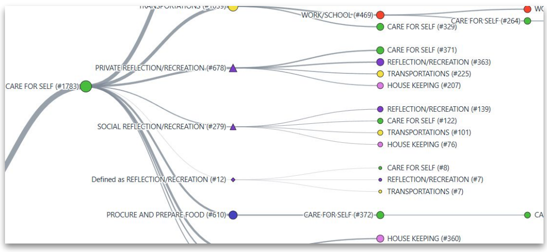
Hover over timeline to see event details

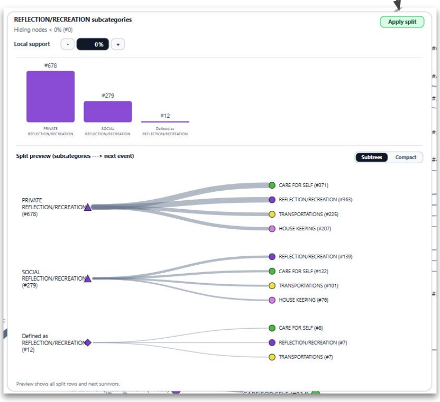


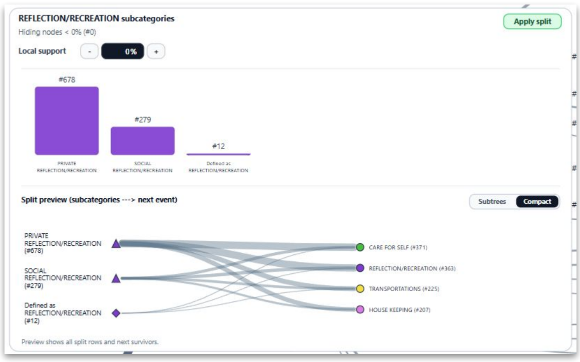
Short overview of **SCOPE**

- To see how these subcategories **contribute** to the structure of the remaining subtree → **"Split" = drill-down**
- Preview is shown **(with two different viewing-options)**

  and then **"apply split"** will **redraw the overview graph**



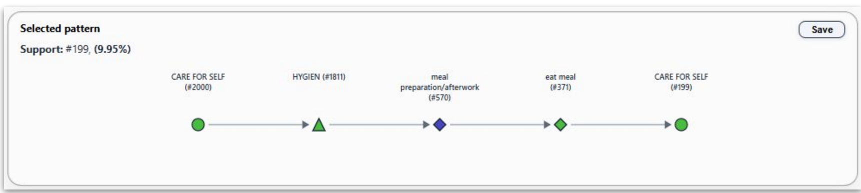




Short overview of **SCOPE**

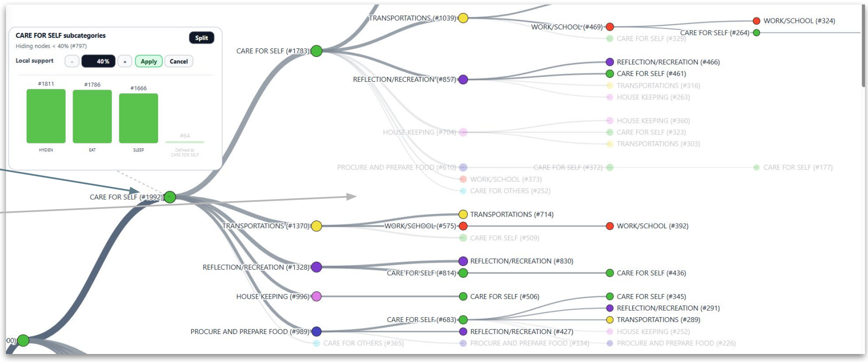
- Then the graph and thus a selected pattern can consist of **high**, **middle** and **low** level events
- A pattern can be **saved** and **compared** to other patterns..

  ...and then it can easily seen how much detail you get with how much support you lose.

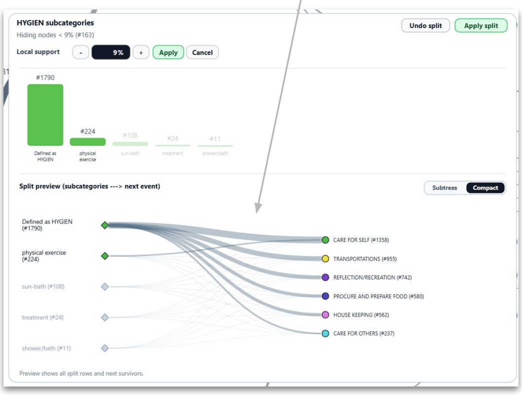


Short overview of **SCOPE**

(Images depicts the preview, before constraint is applied)

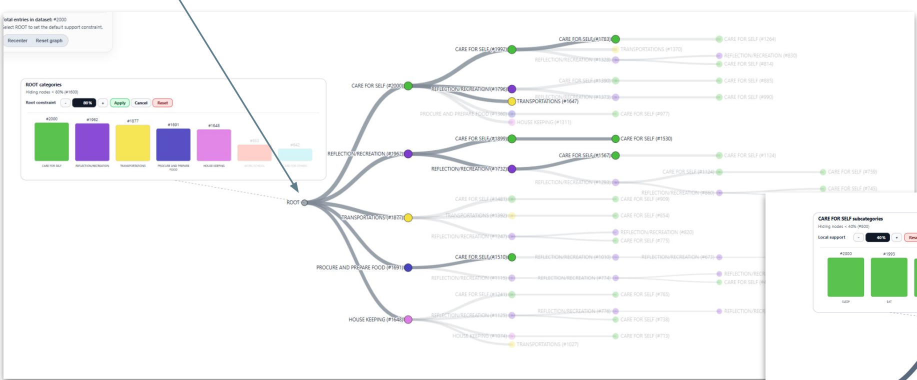


- A **local constraint** can be set on a node → filtering out nodes with a threshold
- It **“grays out”** the connections that will be lost, in graph and split preview.

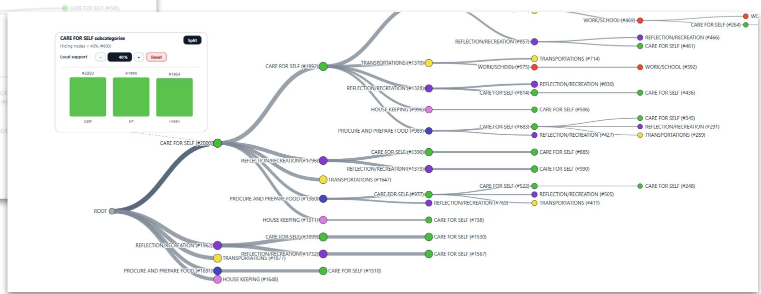


Short overview of **SCOPE**

- Constraints are inherited from parent → root constraint = **global constraint**

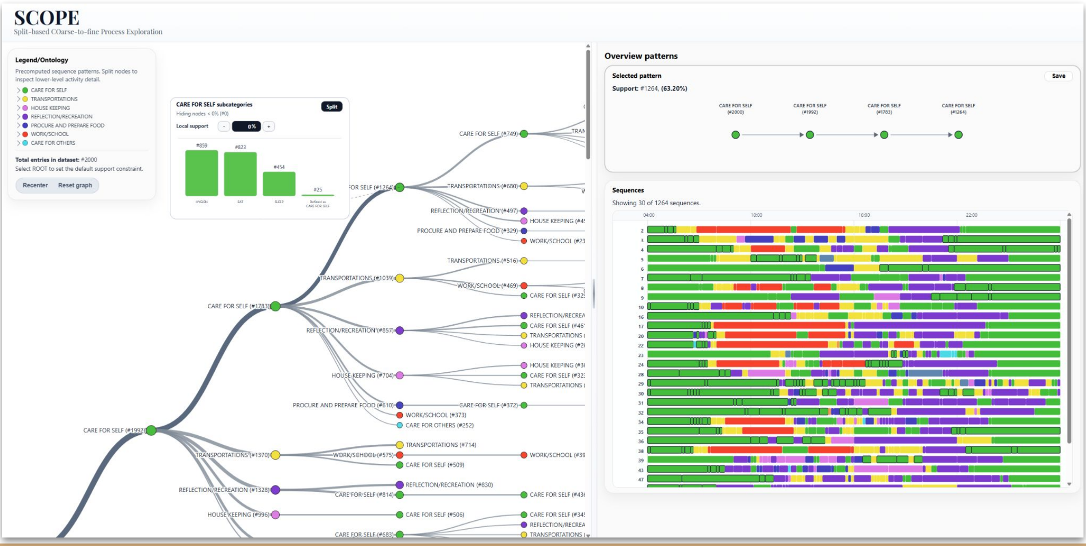


...but also possible to change the local again for the next coming nodes



Short overview of **SCOPE**

- Overview-graph for selection
- Pop-up panel for interactive drill-downs
- Side-panel for pattern trace + raw data in timelines



Demo/tasks

- **Task 1**
  Find and save a **detailed** pattern with at least 50% support that provides **more insight** than the higher level pattern alone.

- **Task 2**
  Identify the **second** most frequent **PRIVATE REFLECTION/RECREATION** activity,
  then identify which activity most follows it directly in sequence.
  What is the support for this pattern?

- **Task 3**
  Use split/drill-down to refine a pattern.
  Describe how **this** refined pattern differs from the original high-level pattern.

Reflection

1. What was hard to understand or follow?
2. Did the **split** (drill-down) contribute to more information in some case?
3. Did changing granularity levels help you discover **new insights**?
4. Was it clear how **different abstraction levels** were represented in the patterns?
5. Did the **local constraints** help narrow down meaningful patterns?
6. Were there moments where you **lost track** of what abstraction level you were viewing?
7. When **would** you use a system like this, and when would you **not**?
8. What is **missing** or can be **improved**?

Thank you for your time!