EuroVis Workshop on Visual Analytics (2026)
D. Archambault and A. Vilanova (Editors)

Seeing and Understanding Process Ecosystems: A Case Study on Conference Submission Handling

N. Andrienko[†1] and G. Andrienko[1] and I. Beerepoot[2] and L. Cibulski[3] and C. Di Ciccio[2] and G. Meroni[4] and C. Turkay[5] and T. von Landesberger[6]

[1] Fraunhofer Institute IAIS, Germany, and City St George's University of London, UK
[2] Utrecht University, The Netherlands
[3] University of Rostock, Germany
[4] Technical University of Denmark, Denmark
[5] University of Warwick, UK
[6] University of Cologne, Köln, Germany

Abstract

Process mining traditionally focuses on individual processes, prioritizing case-level control flow and performance. Yet many organizations operate process ecosystems: collections of interrelated processes sharing resources, schedules and goals. This paper explores how visual analytics can support interactive analysis of such ecosystems. Using scientific conference management as a case study, we characterize submission handling as a multi-instance, multi-actor, multi-phase ecosystem. We derive representative analysis tasks and present a prototypical visual analytics system combining temporally aligned overview and detailed visualizations to reveal bottlenecks, actor workloads, and delays across concurrent processes. Using a realistic synthetic event log, we demonstrate ecosystem-level analysis and discuss design implications for visual process analytics.

CCS Concepts

• ***Human-centered computing*** → *Visual Analytics;* • ***Applied computing*** → *Business process monitoring;*

1. Introduction

Process mining (PM) [Aal16, vdAC22] has developed into a powerful methodology for discovering, monitoring, and improving business processes from event data. Most existing approaches, however, focus on a single, well-defined process, prioritizing case-level control flow and aggregate performance metrics. In many real-world settings, organizations operate *process ecosystems*: collections of interrelated process instances or component parts that run in parallel, share resources, and are constrained by common schedules and goals [vdEJM*25]. Understanding such ecosystems requires time-aware, multi-perspective analysis in which analysts monitor global progress, identify bottlenecks, assess workloads, and interpret delays in context across many concurrent process instances.

Existing PM tools provide limited support for ecosystem-level questions. Common abstractions like directly-follows graphs (DFGs) highlight aggregate control flow but suppress temporal structure, actor behavior, and cross-instance interactions [RKKM25]. Time is often reduced to average durations or simple Gantt charts with limited scalability and interactivity, making it difficult to inspect schedule alignments, shared actor effects, or coordination problems [AMST23]. With many process instances interacting in diverse ways, analysts require multi-faceted, temporally aligned visual representations that support seamless switching between case-, actor-, and action-centric views.

The management of submissions for a scientific conference is a representative example of such a process ecosystem. Each submission constitutes a process instance that progresses through multiple phases (e.g., reviewer assignment, external reviewer recruitment, reviewing, discussion, decision, and possibly revision). Multiple actors (chairs, program committee members, external reviewers) participate in several submissions at the same time, each conducting their role-specific thread of activities for each submission. The ecosystem is constrained by a global schedule with shared deadlines, yet subject to stochastic events such as declined invitations, non-responses, and late reviews. Delays caused by a single actor in one thread can affect many submissions, but these interdependencies remain largely invisible in traditional log views and DFG-based representations.

This paper explores how visual analytics (VA) can help make interdependencies in process ecosystems visible and analyzable. Guided by concrete analytical tasks of conference organizers, we

† This work was done with support of Lamarr Institute for Machine Learning and Artificial Intelligence. This paper is a result of the discussions that took place at the Dagstuhl Seminar 25152, "Multi-Faceted Visual Process Mining and Analytics".

design and implement a prototype that combines detailed process visualizations with temporal overviews to support ecosystem-level analysis. We make the following contributions:

- We characterize conference submission handling as a multi-instance, multi-actor, multi-thread, multi-phase process ecosystem with shared time constraints and actors, and derive a set of representative analytical tasks (Section 3).
- We present a coordinated visualization prototype comprising a *Process Timeline View* with case-centric and actor-centric modes and an *Overview Display* summarizing phase fulfillment and action distributions over time, designed to expose temporal structure, interdependencies, and delays, and to support multi-perspective, cross-case and cross-actor analysis (Section 4).
- Using a realistic synthetic event log derived from EuroVis submission management, we demonstrate how the prototype supports ecosystem-level tasks such as global progress monitoring, bottleneck identification, workload analysis, and delay interpretation (Section 4).
- We reflect on design implications for visual analytics of process ecosystems, emphasizing complementary perspectives, temporally aligned representations of planned versus actual progress, and interactive linking between detailed timelines and aggregate overviews (Section 5).

2. Related Work

Process mining (PM) provides methods for discovering, analyzing, and monitoring processes from event logs [Aal16]. Traditional PM tools and visualizations, such as Directly-Follows Graphs (DFGs), support single-process, case-centric analysis but lack time-aware, multi-perspective exploration of complex settings [MDSW24]. Recent work advocates *multi-faceted visual analytics* [vdEJM*25, AAB*24] and introduces timeline-based techniques making temporal structure more explicit [RKKM25]. Dotted charts [SA] show events over time, revealing temporal patterns. For multi-actor settings, there are approaches extracting social networks and interactions from event logs, revealing how actors collaborate across process instances [ARS05]. However, these do not directly target ecosystem-level coordination and phase-based analysis.

Visual analytics (VA) offers complementary techniques for time-oriented data [AMST23]. Extensive research addresses temporal event sequences [GGJ*22, Jen23]. Arc diagrams reveal structural patterns in sequences [Wat02]. Tools like LifeFlow [WGGP*11], EventFlow [MLL*13, DSP*17], Double Tree [KABB23], and IVESA [BBC*25] support alignment, simplification, and aggregation of event sequences to identify patterns and deviations. Time–space representations inspired by Marey's train schedules [Tuf83] have been used for managing air traffic [AAG*22] and for analyzing production workflows [XMRC17], emphasizing progression and throughput across parallel instances. Storyline visualizations [TM12, LWW*13, DGDL*21] depict actors as lines over time, encoding coordination via crossings, bundles, or proximity.

These techniques demonstrate visually encoding roles, interactions, and temporal coordination in multi-actor settings. Yet, they lack direct support for process analysis with explicit phases, deadlines, constraints, and activity types. Our work targets structured, multi-phase workflows with shared schedules and explicit roles. Our prototype integrates phase-centered, deadline-aware encodings and aggregated phase-fulfillment overviews with detailed multi-thread timelines, addressing ecosystem questions including global phase progress, bottlenecks from thread interdependencies, and actor workload patterns not well supported by existing tools alone.

3. Working Example: Conference Submission Management

The management of submissions to a scientific conference is a representative example of a high-concurrency, resource-constrained process ecosystem. It comprises hundreds of parallel submissions as *process instances*. Each instance consists of multiple interrelated *threads* of activity performed by different *actors* in different *roles*: paper chair, program committee (PC) member acting as primary or secondary reviewer, and external reviewer.

Each submission evolves through a strict sequence of time-constrained *phases*: 1) After submission, paper chairs assign a primary (coordinator) and a secondary PC member to each paper. 2) Assigned PC members invite external reviewers, often through multiple invitation attempts due to declines or non-responses. 3) PC members and external reviewers produce at least four reviews, monitored by coordinators and chairs to handle delays or substitutions. 4) Reviewers discuss and the primary PC member synthesizes the discussion into a summary review and recommendation. 5) Chairs issue first-round decisions based on recommendations. 6) Conditionally accepted papers are revised, checked by the primary PC member, and receive a final decision from the chairs.

In process ecosystems, interactions between process instances can take many forms, including parallel coordination, symbiosis, interdependency, merging or splitting of workflows, competition for shared resources, and the encapsulation of sub-processes within larger ones. While these types define the theoretical breadth of ecosystem interactions, our current case study and prototype focus on a specific subset relevant to conference management: competition for shared resources (namely, actors), encapsulation (threads nested within submission processes), and sequential dependencies, when delays in one activity can propagate through the ecosystem.

The primary goal we address is the **transfer of management knowledge** from experienced chairs to their successors. In this context, the following tasks must be supported:

T1: Understand the global temporal structure. Reconstruct the canonical sequence of phases, their durations, and key deadlines, and relate these to how submissions actually progress over time.

T2: Identify typical bottlenecks and long tails. Determine which phases tend to accumulate delays, which systematically finish late, and how long "tails" of late activities usually extend.

T3: Analyze behavioral patterns of actors and roles. Explore how paper chairs, PC members, and external reviewers typically act over time (e.g., response times, review completion, recruitment efforts) and how these behaviors differ across roles.

T4: Understand interactions and interdependencies between submissions. Examine how shared actors and shared deadlines connect otherwise independent submissions, how competition for reviewers creates delays, and how delays in one activity propagate to other parts of the ecosystem.

Proceedings published by Eurographics - The European Association for Computer Graphics.

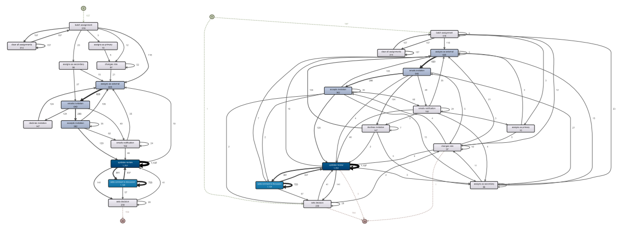

Figure 1: *DFGs of the submission handling created with Disco (fluxicon.com/disco). Left: 30% most frequent directly-follows links (simplified view). Right: 75% of links (denser but still incomplete).*

T5: Investigate recurrent problematic situations and their resolution. Identify common failure modes (e.g., repeated declines, non-responses, late assignments, missing reviews) and reconstruct how chairs and PC members intervened (e.g., reassignments, emergency reviewers, reminders) and when these interventions occurred.

T6: Assess workload distribution and role configurations. Analyze how submissions and responsibilities are distributed across chairs and PC members, identify recurring overload situations, and assess which assignment patterns lead to smoother or more problematic review cycles.

A secondary goal is **on-line process monitoring**, which involves the following tasks:

M1: Track current progress against the planned schedule. Continuously assess submissions that are on time or delayed in each phase, and identify phases at risk of missing their deadlines.

M2: Detect at-risk submissions and emerging bottlenecks. Identify submissions whose activities (e.g., external recruitment, review writing, discussion) are significantly delayed or stalled, and detect clusters that may indicate emerging bottlenecks.

M3: Monitor activity patterns for key actions. Observe when crucial actions (e.g., assignments, review completions, discussions) occur over the conference timeline, and detect unusual drops or surges that may require intervention.

M4: Monitor workload and balance responsibilities. Follow the distribution of ongoing tasks across chairs and PC members to identify overload situations and support timely redistribution or reassignment of work.

4. Looking Into Process Ecosystems: Visualization Prototype

To enable the exploration of conference submissions handling as a process ecosystem, we have implemented an interactive visualization prototype and applied it to a realistic synthetic action log reflecting the full papers track of the EuroVis as a representative scientific conference. This log captures the activities of paper chairs, PC members, and external reviewers but does not include actions of paper authors (such as submissions and revisions).

4.1. Limitations of Traditional Approaches

Directly-Follows Graphs (DFGs) face a completeness-readability trade-off: including more behavior yields dense "spaghetti" diagrams (Fig. 1, right); stronger filtering hides infrequent but relevant variants (Fig. 1, left). DFGs lack encoding for phases, actors, or inter-instance relationships and provide limited temporal structuring [RKKM25]. Our prototype addresses these limitations by offering temporally aligned, multi-faceted views: detailed multi-thread timelines and aggregate overviews summarizing phase fulfillment and activity distributions.

4.2. Process Timeline View: Multi-Threaded Interaction

The Process Timeline View (Fig. 2) represents each process instance (submission) as a collection of threads. A thread is a horizontal line representing the involvement of a single actor. The line is dashed before the actor's first action and solid thereafter. Actor roles are color-coded: paper chairs in red, PC members in blue, and external reviewers in purple. Actions are plotted as dots or abbreviations on these lines; delayed actions are highlighted in red.

To support multi-perspective analysis, the view provides two grouping modes. In the **case-centric mode**, threads are grouped by submission, revealing how all actors coordinate on a single paper. In the **actor-centric mode**, threads are grouped by actor, revealing an individual's workload and behavior across many submissions.

The background encodes the scheduled sequence of phases using pastel colors. A key visual feature is *phase extension*: if a submission fails to complete a phase-specific task by the official end of the phase, the background color of that phase extends horizontally under that submission's lines until the task is fulfilled (Fig. 2, left), making delays visually explicit. Submissions can be interactively *selected*. In case-centric mode, these submissions are highlighted with a bright yellow background; in actor-centric mode, highlighting is applied to threads belonging to selected submissions.

4.3. Overview Display: Temporal Aggregation

The Overview Display (Fig. 3) provides a high-level summary of the process ecosystem's state:

Phase Fulfillment View (top): Daily stacked bars show how many submissions have not yet completed specific phase requirements. Segment colors match the phase background colors in the timeline. Stacking reveals where tasks are carried over into later phases and where collective bottlenecks form.

Action Distribution View (bottom): A set of collapsible histograms shows, for each action type, how often it occurs over time. This allows users to identify bursts of activity or periods of inactivity for key actions. The bar scaling can be local per histogram or global across (non-collapsed) histograms.

4.4. Dynamic Linking and Coordination Between Views

All displays are dynamically linked through interactive selection and filtering to support cross-perspective analysis. Clicking on bar segments in the Phase Fulfillment View (Fig. 3) selects or deselects the corresponding submissions, which are then highlighted in the Process Timeline View (Fig. 2).

Both the Process Timeline View and the Action Distribution

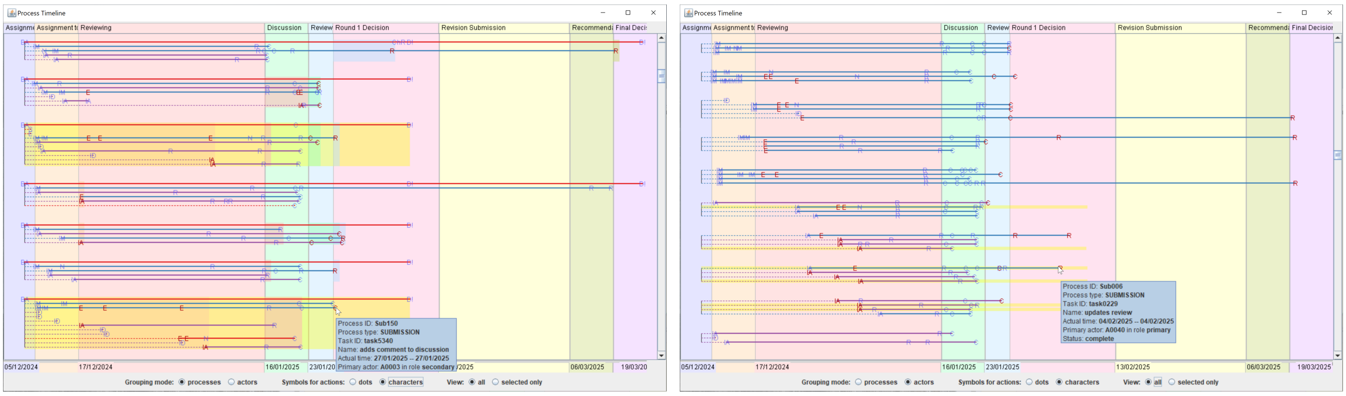

Figure 2: *Process timeline view in two modes: case-centric (left) and actor-centric (right).*

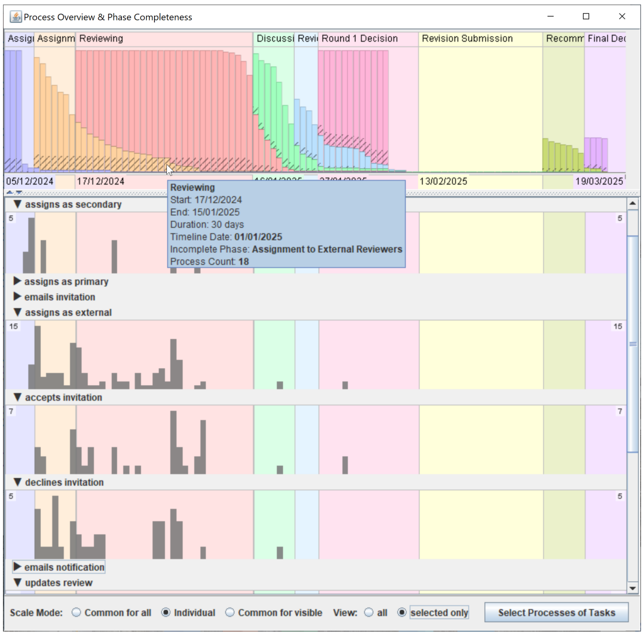

Figure 3: *An overview display including two panels: phase fulfillment progress (top) and action distribution over time (bottom).*

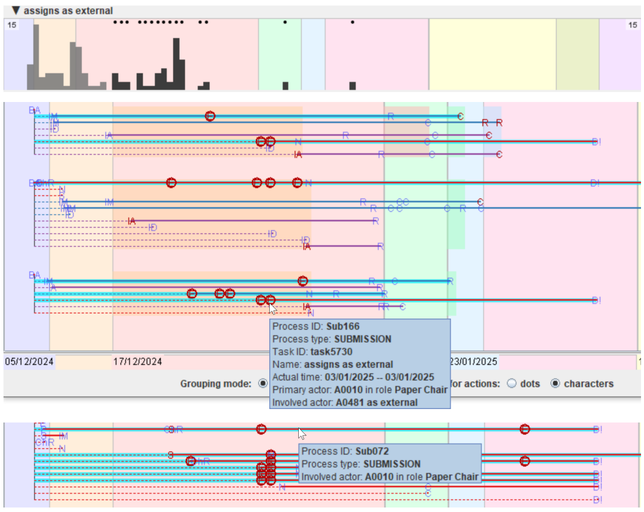

Figure 4: *Display coordination via action selection. Top: A sequence of bars selected in an action distribution histogram. Center and bottom: Representation of selected tasks and threads containing them in the Process Timeline View in the case-centric (center) and actor-centric (bottom) modes.*

View can be switched to a *filtering mode*. In this mode, the Process Timeline View shows only threads belonging to selected submissions (omitting the yellow highlighting), and the Action Distribution View counts only actions from those submissions when computing the histograms. Clicking on a bar or dragging across multiple bars in the Action Distribution View selects all actions of the corresponding type and time range. The Process Timeline View marks these actions with circles and highlights the containing threads in bright cyan (Fig. 4). Selection works in both directions: clicking an action symbol toggles its selection, and clicking a thread line (de)selects all actions in that thread. A button in the Action Distribution View (Fig. 3) selects all submissions containing the currently selected actions.

4.5. An Example Usage Scenario

We illustrate how the tool supports typical analysis tasks with a brief scenario. First, in the Phase Fulfillment View (Fig. 3, top), a chair selects submissions where the external reviewer assignment phase extends into the scheduled reviewing phase. With the Action Distribution View in filtering mode (Fig. 3, bottom), they inspect the histogram for the action *assigns as external* and select occurrences that happen more than one week after the start of the reviewing phase (Fig. 4, top).

Switching the Process Timeline View to filtering mode, the chair inspects the affected submissions (Fig. 4, center) to understand reasons for late assignments. They observe patterns such as repeated declines of invitations, missing follow-up on unanswered invitations, and last-minute emergency invitations by the chairs. In actor-centric mode, bundles of chair threads reveal how often and how late chairs intervened (Fig. 4, bottom).

Proceedings published by Eurographics - The European Association for Computer Graphics.

Next, the chair notices in the Action Distribution View that some *assigns as secondary* actions also occur very late. After canceling filtering, they select bars for this action type appearing during or after the reviewing phase and use the button to select the corresponding submissions. In the Phase Fulfillment View, delayed phase completion becomes visible; in the Process Timeline View, the chair examines when and by whom these late assignments were made, including cases where many secondary assignments by paper chairs were clustered around holiday periods.

4.6. Task Support

The coordinated views jointly support the analysis tasks from Section 3. While our implementation is based on a completed (synthetic) EuroVis log and thus primarily targets knowledge-transfer tasks (T1–T6), the same mechanisms could be applied to live data streams for monitoring tasks (M1–M4).

Global temporal structure and phase-level bottlenecks (T1, T2, M1, M2). The phase background in the Process Timeline View (Fig. 2) and the stacked bars in the Phase Fulfillment View (Fig. 3, top) make the temporal structure of the process visible and relate it to actual behavior across submissions (T1, M1). Bar height and composition reveal where delays accumulate and which phases exhibit long tails of incomplete instances (T2, M2). Selecting bar segments highlights the corresponding submissions in the timeline view for detailed inspection.

Temporal patterns of actions and critical events (T3, M3). The Action Distribution View (Fig. 3, bottom) shows when specific action types occur over the conference timeline, allowing users to identify typical scheduling patterns, bursts of activity, and unusually late actions (T3, M3). Selecting bars or ranges of bars (Fig. 4, top) selects the corresponding actions, which are then emphasized in the Process Timeline View, enabling chairs to examine their context and involved actors.

Interdependencies and multi-threaded dynamics (T4). The multi-thread representation in the Process Timeline View, combined with the two grouping modes, supports analysis of interactions between submissions through shared actors and shared deadlines (T4). In case-centric mode, users see how multiple actors coordinate on a single submission; in actor-centric mode, they see how a given actor participates in many submissions in parallel and how delays or interventions in one thread relate to others. Phase extensions and overlapping threads highlight where resource contention and temporal interdependencies occur.

Problematic situations and intervention strategies (T5). Selection and filtering across views directly support the investigation of problems and their resolution (T5). Starting from late or critical actions in the Action Distribution View, users filter to affected submissions and inspect their timelines to reconstruct typical failure modes and responses (Fig. 4, center). In actor-centric mode, they see how often and how late chairs intervened (e.g., inviting emergency reviewers) and how these interventions align with subsequent phases (Fig. 4, bottom). Users can also see how delayed actions manifest in phase completion patterns.

Workload distribution and role configurations (T6, M4). Grouping threads by actor and encoding roles by color allows users to inspect temporal workload patterns for individual chairs and PC members (T6, M4). Selections based on delays or late actions reveal which actors are repeatedly involved in problematic situations and how responsibilities are balanced across the PC.

5. Discussion and Conclusion

By focusing on conference submission handling as a concrete example of a process ecosystem, we have been able to make several observations about how such ecosystems can be analyzed and visualized. First, the case study confirms that even when only part of an ecosystem is captured in an event log (here: actions in the conference management system), it is still possible to address a meaningful set of ecosystem-level analysis tasks (Section 3) centered on temporal structure, bottlenecks, interactions, and workload. These tasks differ from traditional, single-process mining questions and explicitly reflect the multi-instance, multi-actor, and multi-phase nature of the ecosystem.

Second, the design and use of the prototype indicate that a small number of coordinated views can address many of these tasks. The combination of (i) temporally aligned, phase-aware timelines, (ii) aggregate views of phase fulfillment and action distributions, and (iii) interactive selection and filtering across case-centric and actor-centric perspectives proved sufficient to support all knowledge-transfer tasks T1–T6 and, conceptually, the monitoring tasks M1–M4. This suggests that temporal alignment and multi-perspective support are core design principles for visual analytics in process ecosystems.

Third, working with a specific ecosystem made some of the conceivable process interaction types (Section 3) more concrete. In the conference setting, *competition* for shared actors and *encapsulation* of sub-processes (threads) are prominent, as are *interdependencies* between sub-processes (e.g., reviewing and decision-making). Other interaction types, such as splitting and merging of instances or symbiotic processes, are less relevant. This suggests that real ecosystems may involve only a subset of theoretically possible relations, and that the design of visual analytics solutions should target the relations that are actually dominant in a given domain.

The scope of the present work is inevitably limited. We concentrate on a single, well-structured ecosystem and on the subset of interaction types manifested in the available log data, so we cannot claim that our task formulation or visual design generalize to ecosystems where other interactions play a central role. Applying the same methodology (starting from a concrete ecosystem, deriving task requirements, and iteratively designing coordinated visualizations) to other domains would allow researchers to test and refine both the task framework and the interaction typology. Ecosystems with richer merging/splitting behavior, stronger symbiotic relations, or more complex process hierarchies could motivate additional visual encodings and interaction techniques.

Overall, our work illustrates how considering a specific process ecosystem and building a prototype for its analysis can yield more general insights. It highlights the importance of ecosystem-level task formulations, the value of temporally aligned multi-perspective views, and the need to reason explicitly about inter-instance relations when designing visual analytics solutions.

Proceedings published by Eurographics - The European Association for Computer Graphics.

References

[AAB*24] Alman A., Arleo A., Beerepoot I., Burattin A., Di Ciccio C., Resinas M.: Tiramisù: making sense of multi-faceted process information through time and space. *Journal of Intelligent Information Systems* (2024), 1–23. doi:10.1007/s10844-024-00875-8. 2

[AAG*22] Andrienko G., Andrienko N., Garcia J. M. C., Hecker D., Vouros G. A.: Supporting visual exploration of iterative job scheduling. *IEEE Computer Graphics and Applications 42*, 3 (2022), 74–86. doi:10.1109/MCG.2022.3163437. 2

[Aal16] Aalst W.: *Process Mining: Data Science in Action*. 01 2016. doi:10.1007/978-3-662-49851-4. 1, 2

[AMST23] Aigner W., Miksch S., Schumann H., Tominski C.: *Visualization of Time-Oriented Data, Second Edition*. Human-Computer Interaction Series. Springer, 2023. doi:10.1007/978-1-4471-7527-8. 1, 2

[ARS05] Aalst van der W., Reijers H., Song M.: Discovering social networks from event logs. *Computer Supported Cooperative Work 14*, 6 (2005), 549–593. doi:10.1007/s10606-005-9005-9. 2

[BBC*25] Bernard J., Barth C.-M., Cuba E., Meier A., Peiris Y., Shneiderman B.: Ivesa – visual analysis of time-stamped event sequences. *IEEE Transactions on Visualization and Computer Graphics 31*, 4 (2025), 2235–2256. doi:10.1109/TVCG.2024.3382760. 2

[DGDL*21] Di Giacomo E., Didimo W., Liotta G., Montecchiani F., Tappini A.: *Storyline Visualizations with Ubiquitous Actors*. 02 2021, pp. 324–332. doi:10.1007/978-3-030-68766-3_25. 2

[DSP*17] Du F., Shneiderman B., Plaisant C., Malik S., Perer A.: Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (2017), 1636–1649. doi:10.1109/TVCG.2016.2539960. 2

[GGJ*22] Guo Y., Guo S., Jin Z., Kaul S., Gotz D., Cao N.: Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2022), 5091–5112. doi:10.1109/TVCG.2021.3100413. 2

[Jen23] Jentner W.: *Visual Pattern Analytics for Event Sequences*. PhD thesis, Universität Konstanz, Konstanz, 2023. 2

[KABB23] Krause C., Agarwal S., Burch M., Beck F.: Visually abstracting event sequences as double trees enriched with category-based comparison. *Computer Graphics Forum 42*, 6 (2023), e14805. doi:https://doi.org/10.1111/cgf.14805. 2

[LWW*13] Liu S., Wu Y., Wei E., Liu M., Liu Y.: Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec. 2013), 2436–2445. doi:10.1109/TVCG.2013.196. 2

[MDSW24] Miksch S., Di Ciccio C., Soffer P., Weber B.: Visual analytics meets process mining: Challenges and opportunities. *IEEE Computer Graphics and Applications 44*, 6 (2024), 132–141. doi:10.1109/MCG.2024.3456916. 2

[MLL*13] Monroe M., Lan R., Lee H., Plaisant C., Shneiderman B.: Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2227–2236. doi:10.1109/TVCG.2013.200. 2

[RKKM25] Rubensson C., Kaur H., Kampik T., Mendling J.: Timeline-based process discovery. *Information Systems 133* (2025), 102568. doi:10.1016/j.is.2025.102568. 1, 2, 3

[SA] Song M., Aalst W.: Supporting process mining by showing events at a glance. In *Seventeenth Annual Workshop on Information Technologies and Systems (WITS'07), pp.139–145, Montreal, Canada, December 8-9, 2007*, Chari K., Kumar A., (Eds.), pp. 139–145. 2

[TM12] Tanahashi Y., Ma K.-L.: Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2679–2688. doi:10.1109/TVCG.2012.212. 2

[Tuf83] Tufte E. R.: *The Visual Display of Quantitative Information*. Graphics Press, 1983. 2

[vdAC22] van der Aalst W. M., Carmona J.: *Process mining handbook*, vol. 448. Springer, 2022. doi:10.1007/978-3-031-08848-3. 1

[vdEJM*25] van den Elzen S., Jans M., Martin N., et al.: Towards multi-faceted visual process analytics. *Inf. Syst. 133* (2025), 102560. doi:10.1016/J.IS.2025.102560. 1, 2

[Wat02] Wattenberg M.: Arc diagrams: visualizing structure in strings. In *INFOVIS* (2002), pp. 110–116. doi:10.1109/INFVIS.2002.1173155. 2

[WGGP*11] Wongsuphasawat K., Guerra Gómez J. A., Plaisant C., Wang T., Taieb-Maimon M., Shneiderman B.: Lifeflow: Visualizing an overview of event sequences. pp. 507–510. doi:10.1145/1979742.1979557. 2

[XMRC17] Xu P., Mei H., Ren L., Chen W.: Vidx: Visual diagnostics of assembly line performance in smart factories. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 291–300. doi:10.1109/TVCG.2016.2598664. 2