Exploring Complex Group Dynamics

Visual Analysis of Overlapping Groups and Interactions Over Time

Shivam Agarwal

Exploring Complex Group Dynamics

Visual Analysis of Overlapping Groups and Interactions Over Time

Dissertation
zur Erlangung des Doktorgrades
Dr. rer. nat.
der Fakultät für Wirtschaftswissenschaften
der Universität Duisburg-Essen

vorgelegt
von

Shivam Agarwal
aus
Prayagraj (Allahabad), India

Betreuer: Prof. Dr. Fabian Beck

Institute of Computer Science and Business Information Systems
University of Duisburg-Essen, Germany

Essen, December 2023

DOI: 10.17185/duepublico/81425
URN: urn:nbn:de:hbz:465-20240215-102500-9

Examiners
Prof. Dr. Fabian Beck
Prof Dr. Torsten Brinda
Prof. Dr. Tatiana von Landesberger

Date of Oral Exam
12th October 2023

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

ABSTRACT

Analyzing group dynamics is crucial to understand the behaviors of members in groups. The analysis could help answer important questions: *How does the relationship between group members change over time? What is the effect of a member's action on others? How do group members coordinate their efforts to achieve a goal? What are the overall changes in group behaviors over time?* This thesis proposes novel techniques for visually exploring group dynamics, thereby aiding in answering such questions. The proposed techniques apply to diverse scenarios, as demonstrated through application examples.

This work focuses on the two characteristic features of group dynamics. First, since members can belong to multiple groups simultaneously, it results in overlaps between groups. We propose two novel visualizations to analyze dynamic memberships in overlapping groups. Their effectiveness is demonstrated by insights from application examples, e.g., authors' evolving research interests, classification models' performance in their training process, and developer contributions in software repositories. Second, the interactions among group members. A design and application space is proposed to explore user behaviors from mixed reality sessions. Three visualizations are presented to investigate collaborative and competitive interactions among members. The studied scenarios include humans interacting in mixed reality and autonomous agents collaborating and competing in simulated environments.

We propose an example of an integrated visual representation to show dynamic memberships in overlapping groups and entity interactions. The thesis discusses the future possibilities in encoding enriched interactions and describes a few works in progress. Finally, the thesis summarizes the contributions, highlights the limitations of the proposed visualizations, and presents a brief outlook toward the future.

Dedicated to my loving family for their support and sacrifices.

ACKNOWLEDGMENTS

I am extremely grateful to Fabian Beck for providing reliable, constructive, and consistent supervision during the PhD. I remember several engaging discussions with him on focused technical topics and academic research in general. Moreover, Fabian encouraged me to propose and pursue my ideas not only to steer the research projects in a specific direction but also extensively supported me to become a better teacher. Several thoughtful efforts by him, for instance, creating learning opportunities through workshops and establishing a healthy research environment, have played a critical role in nurturing me as a researcher. I am also thankful to Torsten Brinda for chairing the PhD committee and Tatiana von Landesberger for accepting the role of secondary examiner. I am especially grateful to Tatiana for providing feedback and accommodating the delay in the submission process.

I am blessed and humbled by the support of remarkable collaborators, particularly Günter Wallner, who not only lent an ear to my unconventional ideas as an inexperienced PhD candidate but also generously assisted in bringing them to fruition. I am glad to have Michel Wermelinger and Fabian Beck as co-authors in my first PhD project on dynamic set visualization. As the project faced multiple rejections, it tested my limits of perseverance. The co-authors helped me to navigate the complexities of the project and make improvements with each re-submission, finally receiving the best paper award at a conference. Additionally, I would like to mention and thank Maximilian Krug and Karola Pitsch for being very supportive and providing timely help during the student project and Jonas Auda and Stefan Schneegaß for their contributions in realizing the value of visualizations in mixed reality user sessions.

Of course, it would have been difficult to complete the projects without an optimistic and healthy work environment. I am lucky that Shahid Latif, Cedric Krause, Hagen Tarner, Sebastian Surminski, and Michael Rodler were my colleagues. The humorous conversations and lunch breaks helped to maintain my sanity level, especially while chasing challenging deadlines. Thanks to my German colleagues for patiently answering several informal queries (e.g., with Deutsche), which helped me to ease into life in Germany. I would especially like to thank Shahid Latif, who, besides being a wonderful co-author, gave valuable feedback on the projects and extensively discussed the technical aspects of the research. Thank you for the long philosophical discussions to help me think beyond the limited scope of the research topic and for accompanying me in sports activities to sometimes not think about research at all.

I am grateful to the students who saw the value in some of our ideas and accepted our supervision by taking them as a project or a thesis in their study

program. For instance, Carina Liebers did an excellent job by leading the project of visualizing multimodal interactions in a specific scenario as part of her Master thesis; Christian Herrmann, Aristide Rothweiler, Marc Drescher, and Florian Kunz demonstrated the value of visualizing behaviors of autonomous agents in games; while Mariana Pron designed and executed a wonderful user study for comparing dynamic set visualizations. I am immensely grateful to the students who bravely embraced challenging concepts, like Clara Kruse's exploration of using gamification to illustrate fairness and bias in machine learning or developing innovative approaches in enriched data comics (winter semester 2022/23 project course at the University of Bamberg). Since an exhaustive list would be infeasible to accommodate here, I would just like to thank all of them and mention that their dedication has been inspiring.

Reflecting back on the journey, I am thankful to Jaya Sreevalsan Nair and T. K. Srikanth for nourishing my research interests during the Master's studies at the International Institute of Information Technology Bangalore (IIITB), India. They introduced me to the world of data visualization and computer graphics, which sparked an interest in the topic. Their early mentorship and support have been critical for me to pursue a PhD in the field.

The social encouragement from friends has been immensely helpful in seeing the bigger picture. The meaningful discussions with Amit Tomar, Srinivas Vaidya, and Pankaj Agrawal have helped me to see the necessity of various aspects of education. These discussions have inspired me to improve continuously as an educator and researcher. I am also thankful for my ex-colleagues at IIITB, especially Reddy Rani, who helped me navigate into the exciting world of research.

Last but not least, I have the deepest gratitude for my family. Without their love, care, and sacrifices, completing the PhD would have been impossible. I am indebted to my parents (Rajeev and Abha Agarwal) for nurturing and encouraging me to become a researcher. A special thanks to my siblings Ashish Agarwal and Sonal Agarwal, who supported me in every possible manner and stood by my decision to pursue a PhD in a foreign land thousands of miles away. I am also grateful to my in-laws (Seema, Rajesh, and Shaurya Bajpai) for understanding and enabling me to pursue a career overseas for an extended duration. And lastly, none of this would have been possible without my best friend and wonderful wife, Ankita Bajpai. No amount of words will be enough to describe her contributions that have been critical during the PhD. I can just say that she has always been my steady rock, especially amidst the tides of turmoil. Thank you for everything!

PREFACE

The ideas presented in this work are submitted as a PhD dissertation at the Institute of Computer Science and Business Information Systems, University of Duisburg-Essen, Germany. The research has been supervised by Prof. Dr. Fabian Beck. As the secondary examiner, I suggest Prof. Dr. Tatiana von Landesberger.

The research conducted during the PhD has been published in scientific journals and presented at various conferences, including EuroVis (2020, 2022, 2023), IEEE VIS (2018, 2020), VMV (2020), PacificVis (2022), and IEEE CoG (2020). The parts of the content presented in this dissertation—including ideas, approaches, results, and figures—are based on the following peer-reviewed and already published research articles:

- **Shivam Agarwal** and Fabian Beck. "Set Streams: visual exploration of dynamic overlapping sets." In: *Computer Graphics Forum* 39.3 (2020), pp. 383–391. ISSN: 1467-8659. DOI: 10.1111/cgf.13988 - Chapter 3
- **Shivam Agarwal**, Gleb Tkachev, Michel Wermelinger, and Fabian Beck. "Visualizing sets and changes in membership using layered set intersection graphs." In: *Vision, Modeling, and Visualization*. 2020. DOI: 10.2312/vmv.20201189 (Best Paper Award) - Chapter 4
- **Shivam Agarwal**, Jonas Auda, Stefan Schneegaß, and Fabian Beck. "A design and application space for visualizing user sessions of virtual and mixed reality environments." In: *Vision, Modeling, and Visualization*. 2020. DOI: 10.2312/vmv.20201194 - Chapter 5
- **Shivam Agarwal**, Günter Wallner, and Fabian Beck. "Bombalytics: visualization of competition and collaboration strategies of players in a bomb laying game." In: *Computer Graphics Forum* 39.3 (2020), pp. 89–100. ISSN: 1467-8659. DOI: 10.1111/cgf.13965 - Chapter 6
- **Shivam Agarwal**, Günter Wallner, Jeremy Watson, and Fabian Beck. "Spatio-temporal analysis of multi-agent scheduling behaviors on fixed-track networks." In: *IEEE Pacific Visualization Symposium (PacificVis)*. 2022, pp. 21–30. DOI: 10.1109/PacificVis53943.2022.00011 - Chapter 7
- **Shivam Agarwal**. "Visualizing element interactions in dynamic overlapping sets." In: *EuroVis - Short Paper*. 2023. DOI: 10.2312/evs.20231050 - Chapter 8

Some approaches for future research directions that align with the dissertation topic have also been published (e.g., as work in progress), to which I contributed as supervisor of the student-led projects:

- **Shivam Agarwal**, Uttiya Ghosh, Fabian Beck, and Jaya Sreevalsan-Nair. "CiteVis: visual analysis of overlapping citation intents as dynamic sets." In: *IEEE Pacific Visualization Symposium (PacificVis - Poster)*. 2022 - Chapter 9

- **Shivam Agarwal**, Christian Herrmann, Günter Wallner, and Fabian Beck. "Visualizing AI playtesting data of 2D side-scrolling games." In: *IEEE Conference on Games - Short Paper)*. 2020, pp. 572–575. DOI: 10.1109/CoG47356.2020.9231915 - Chapter 9

- Carina Liebers, **Shivam Agarwal**, Maximilian Krug, Karola Pitsch, and Fabian Beck. "VisCoMET: visually analyzing team collaboration in medical emergency trainings." In: *Computer Graphics Forum* (2023). ISSN: 1467-8659. DOI: 10.1111/cgf.14819 - Chapter 9

- **Shivam Agarwal**, Shahid Latif, Aristide Rothweiler, and Fabian Beck. "Visualizing the evolution of multi-agent game-playing behaviors." In: *EuroVis - Poster*. 2022. DOI: 10.2312/evp.20221111 - Chapter 9

Other ideas have also been published during the PhD [1, 2, 3, 4, 5]. Since they are not in the main scope of the dissertation, they have not been included in the list above but referenced in the text wherever applicable.

The images used for the cover design and the three different parts of the thesis have been generated using DALL-E, an image generation model by OpenAI.

—Shivam Agarwal

CONTENTS

LIST OF FIGURES

LIST OF TABLES

"Group dynamics are the influential actions, processes, and changes that occur within and between groups. Groups come in all shapes and sizes and their functions are many and varied, but their influence is universal. The tendency to join with others in groups is perhaps the single most important characteristic of humans, and the processes that unfold within these groups leave an indelible imprint on their members and on society. To understand people, one must understand groups and their dynamics."

– Donelson R. Forsyth [6]

1 INTRODUCTION

Group dynamics broadly refers to the changes or processes that occur in a group. It includes the actions performed by the members of a group and their reactions under changing circumstances. For over a century, scholars from diverse disciplines have been intrigued by the complex human behavior in groups. They have attempted to answer important questions, such as *what is the influence of an individual's actions on the group?* and *how do members of a group coordinate their efforts to achieve a goal?* Thus, group dynamics has emerged as a research topic and is defined as *"the influential actions, processes, and changes that occur within and between groups"* [6]. Conventionally, the complexities in group dynamics have been scientifically analyzed by watching or recording the group activities, quantifying the occurrences of behaviors, conducting self-reported surveys for group members, and via qualitative or quantitative methods. However, there is a lack of exploratory visual analysis approaches that can be used to understand group dynamics. The thesis fills this gap and proposes visual exploration techniques that complement the existing analysis methods of group dynamics.

Generally, a group is a collection of entities that have something in common. However, the definition is a simplification that obscures the involved complexities. Groups can be defined based on different features. For the purpose of this thesis, we consider a group as a collection of entities or members that belong in the same category [7] (e.g., based on a categorical attribute), or *"individuals who stand in certain relations to each other ..."* [8], or as *"[entities] who work together interdependently on an agreed-upon activity or goal"* [9]. Groups have two characteristic aspects. First, due to connections, similarities, or joint memberships of entities, the groups exhibit a structure among members. This is captured through entity attributes. Second, entities interact with each other, both within a group and between groups. Hence, group behavior can be formally defined as a function of the entity attributes, the environment in which the entities act, and the interactions with others.

Time impacts the state and situation of an entity, affecting entity relationships, memberships in a group, or strategy of interacting with other entities. Hence, scenarios involving groups of entities can be considered complex systems with dynamic processes. To understand the group dynamics in specific scenarios, such temporal changes are necessary to visualize and explore insights about the cumulative outcomes, for instance, to understand the overall trend of entity interactions (e.g., cooperation), identify important timespans (e.g., abnormalities), and infer the patterns of evolving relationships among entities.

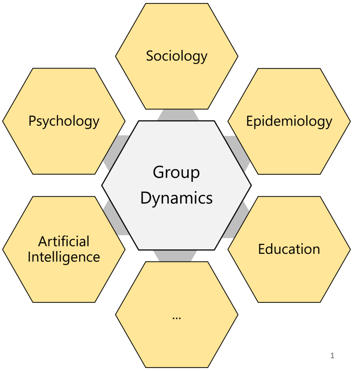


Figure 1.1: Research fields and application areas where the exploratory analysis of group dynamics is valuable.

The inclusion of temporal attribute in the exploratory analysis of group dynamics is valuable and applicable in several areas of research and application, as shown in Figure 1.1. For instance, analysis of group dynamics helps in understanding people's decision-making behavior as an individual and in a society. Hence, the analysis is valuable in psychology and sociology. Targeting a more specific use case, understanding group dynamics is crucial in epidemiology to study the spread of diseases due to interactions and physical contact. Similarly, understanding the interactions and influence of one's actions on others is helpful in a learning environment. Finally, analysis of group dynamics is necessary for training autonomous agents that can cooperate to achieve a common goal. Hence, exploratory analysis of group dynamics plays a pivotal role in understanding the current behavior of entities through their actions and interactions. However, the analysis involves some challenges.

The challenges in the visual exploration of group dynamics stem from the two aspects of group behavior. First, an entity may leave and join a different group due to a change in the group behavior of entities in a scenario (e.g., humans in social networks). Moreover, an entity may simultaneously be a member of multiple groups, resulting in overlapping groups. Hence, embedding temporal changes in memberships while showing the overlapping groups in an integrated visual representation becomes challenging due to the overlap. Second, the interactions between entities in the same or different groups. Since interactions potentially influence other entities, it marks a concrete and vital aspect of behavioral analysis. The interactions have a cascading effect on other entities individually and the groups as a whole. Hence, encoding entity interactions in visualization to efficiently analyze their sequences and influence on other entities to understand the group dynamics becomes a challenge.

Addressing the challenges in visualizing group dynamics is valuable. Take an academic research dataset as an example, where research fields are modeled as groups, while researchers publishing articles in a field determines their group membership. Analyzing the dynamics in this example could help understand the inter-disciplinary research trends (e.g., between visualization and psychology sciences) or between overlapping research fields (e.g., robotics, artificial intelligence, and visualization in computer science). Likewise, exploratory analysis of interactions in group dynamics is critical in scenarios, especially when the strategy of individual entities is unknown or a goal of the analysis. For example, understanding collaborative interactions between groups of humans to perform a task. Moreover, the value of an integrated analysis of the two aspects in understanding group dynamics has been demonstrated in social sciences, for instance, to understand how humans change their memberships in different social groups based on their interactions [10].

The unique characteristics of temporal data need to be effectively handled to address the challenges mentioned above. For instance, the temporal information can be modeled as a discrete point in time or as an interval with duration (if available) on a continuous scale. Moreover, focusing on the progression of time, the information can be modeled and represented as a sequence on a linear timeline, or treating it as a reoccurring cyclic phenomenon, time can be plotted on a radial layout (e.g., to show periodic patterns). Such peculiarities of the temporal data have been discussed, and several visual representations have been proposed for different application domains (e.g., see a survey [11]). Since the visual design for temporal data is heavily influenced by the goals and desired granularity level in the analysis of specific scenarios, we need to carefully embed the temporal information while designing visualizations to analyze group dynamics.

Apart from embedding time, to extract meaningful insights from the visualization, we need to integrate the context of the environment in the representation. While analyzing the temporal group memberships and interactions between entities, the spatial information may be critical to understanding and contextualizing the group dynamics. For instance, analyzing the movement of autonomous robots who need to cooperate and avoid damages (e.g., due to unplanned malfunctions). Generally, several visualizations have been proposed to study movement patterns (e.g., [12]). Taking an even broader perspective, many visual techniques have been proposed (e.g., 3D space-time cube), and challenges have been identified in embedding spatial information into temporal visualizations (e.g., [13, 14, 15]). However, directly applying these techniques to analyze complex group dynamics is not feasible or straightforward. In this thesis, we take inspiration from the existing spatio-temporal visualizations and modify them based on the specific needs of the analysis in a scenario.

Apart from movement, the context regarding the sequence of actions, reactions, and interactions of entities in a scenario must be provided to help understand the exhibited group behavior. This can be done by analyzing the accompanying environmental events that happened within a short time window of the exhibited behavior by an entity. For instance, first gaining the ability to move obstacles in a game environment, followed by placing them at key locations to restrict the movement of enemies. Also, some interactions may have a delayed effect on other entities, indicating a strategic maneuver, e.g., pushing a bomb toward an enemy, which explodes after a few timesteps and kills the enemy in a multi-agent virtual game simulation. Several timeline visualizations with embedded events have been proposed to understand and explore patterns in event sequence data (e.g., see a survey [16]). Thus, to represent the progression of group behavior, in this thesis, we encode the group dynamics as a sequence of actions on a timeline, together with other related events of the studied environment.

The thesis proposes different visualization approaches with the following three design principles to address the challenges and facilitate the visual exploration of group dynamics.

> **Design Principle 1: Static Visual Encodings to Embed the Temporal Changes**
>
> In order to understand group behavior, we need to compare the dynamic actions, interactions, and memberships of entities. The unique behavior exhibited by an entity can be understood by comparing individual entities. Likewise, since we expect a high number of temporal changes in group dynamics, the temporal shift in the exhibited behavior can be analyzed by comparing different timesteps. Hence, to avoid a high cognitive load, we plot the temporal changes through static visual encodings instead of using other approaches (e.g., animation-based techniques).

> **Design Principle 2: Sequential Analysis of Group Dynamics**
>
> An individual change may have a cascading effect on the group structure (e.g., size), follow-up entity interactions, and, in general, the group behavior. Hence, the encodings should facilitate the analysis of these sequential effects. We use a timeline to show the dynamics and enrich it by embedding the relevant information, e.g., the group structure, and differentiate between inter- vs. intra-group interactions.

> **Design Principle 3: Preserve Context in the Visual Exploration of Group Dynamics**
>
> Different environments have specific rules that the entities have to obey. This leads to various restrictions on their behavior (e.g., movement only on fixed tracks). Moreover, environmental factors also influence the behavior of entities (e.g., temporary disruption in the movement due to a random malfunction environmental event). Hence, to contextualize the behavior exhibited by entities, we integrate the relevant environment details and events in the visual representation.

We use the design principles and propose novel visualization approaches to explore group dynamics in a diverse set of scenarios. For dynamic overlapping groups, we explore the evolving research interests of authors, the performance of classification models in their training process, and developer contributions in software repositories. Regarding entity interactions, we analyze human interactions in a mixed reality environment, competition and collaboration between artificial intelligence agents in a virtual bomb laying game (Pommerman [17]), and a train scheduling simulation environment (Flatland [18]). We also analyze an example of an integrated visual representation of memberships in dynamic overlapping groups and evolving entity interactions by extending a proposed approach. Finally, reflecting on the proposed approaches, we discuss a few works in progress, the limitations of proposed approaches, and concrete ideas for future work.

1.1 RESEARCH OBJECTIVES

To address the challenges in visualizing group dynamics, the thesis focuses on the two key aspects of group behavior: membership in overlapping groups and interactions between entities.

The value in visualizing changing memberships in dynamic overlapping groups (or sets) has been highlighted before [19], and the visual complexities have been well understood and discussed [20]. However, only a few approaches have offered solutions, as indicated by a recent survey [21]. Hence, the objective is to propose novel visualizations for temporal analysis of memberships in overlapping groups. The visualization should provide a temporal overview of changes in group memberships of entities. Moreover, the design should support visual comparison between user-defined groups of entities or selected timesteps. In some scenarios, entities can belong to a group with varying significance, e.g., a researcher publishing ten papers vs. another author contributing one article in the same research field. Hence, the objective is to explore visual designs supporting the analysis of membership details (e.g., weight) of each entity in a group.

> **Research Objective – RO 1: Dynamic Overlapping Groups**
> Novel visualization approaches for memberships in dynamic overlapping groups to (1.1) provide a temporal overview, (1.2) support comparative analysis of different entity groups or timesteps, and (1.3) exploration of membership details (e.g., weight) of an entity in a group.

We aim to analyze the sequences and cascading effects of entity interactions. Hence, the next objective is to categorize the existing visualizations that facilitate analysis of multi-user or multi-agent behaviors acting in a dynamic environment. Then, we focus on embedding interactions in a visualization. The entities in a team usually interact with each other to coordinate and achieve a shared goal. In contrast, interactions between different teams are competitive. Thus, the objective is to explore novel visualization designs that facilitate a sequential analysis of such collaborative and competitive group behaviors. Finally, since entities move in an environment while acting and interacting with others, the spatial context is crucial to understand group behavior. Hence, we aim to explore visual techniques for an integrated analysis of evolving entity interactions along with their movement.

> **Research Objective – RO 2: Evolving Entity Interactions**
> (2.1) To categorize the existing visualizations and understand the encodings for entity behaviors and interactions. Explore novel visualization designs to (2.2) analyze the competitive and collaborative entity interactions, and (2.3) integrate spatial context in the analysis of group dynamics.

Apart from the two aspects addressed in the thesis, other factors may influence group dynamics, e.g., individual characteristics of entities, diversity of group members, social norms, and specific restrictions in the studied scenario, to name a few. However, in this thesis, we do not explicitly consider such factors that are subjective or applicable only in specific environments. The narrowed focus limits the scope but is still generalizable to model several complex dynamic processes. For instance, the included aspect of multiple group memberships can model a wide variety of relations, e.g., based on a categorical attribute. Similarly, interactions between entities can model a wide range of behaviors, for instance, communication and cooperation, tangible contact (e.g., lifting an object), online discourse, or signing legal agreements (e.g., between companies).

1.2 THESIS OUTLINE

The thesis has eight main chapters, structured in three parts: dynamic overlapping groups, evolving entity interactions, and conclusion. Along with them, the background chapter provides information on prior work.

PART I – DYNAMIC OVERLAPPING GROUPS

Chapter 3 (RO 1.1 and 1.2) presents Set Streams, a novel visualization approach to analyze dynamic and overlapping group memberships of entities. The partitions of overlapping groups are represented in separate rows, while columns in the horizontal axis show discrete timesteps. The streams between two timesteps represent the temporal changes in group memberships. Three application examples—expertise of researchers, software evolution, and multi-label classification—together with expert feedback showcase the effectiveness and applicability of the proposed approach.

Focusing on membership details of an entity in a group, Chapter 4 (RO 1.1 and 1.3) proposes another visualization of dynamic overlapping groups using layered set intersection graphs. Sets and their intersections are encoded as rectangular boxes in a layered layout. Entities are shown as individual circles inside rectangles, with the circle size encoding the respective membership weight of an entity in a group. The visualization design and rich interactions support the detailed exploration of an individual entity, a group, and changes between two timesteps. Insights from two application examples—evolving interests of researchers and software developer activities—show the capabilities and usefulness of the visualization.

PART II – EVOLVING ENTITY INTERACTIONS

Chapter 5 (RO 2.1) focuses on visualizing entity interactions and events from user sessions of mixed reality environments. To do so, it derives a design and application space with a set of seven categories that could be combined to build visualization approaches addressing the analysis goals. A scenario of evaluating user studies combines the visualizations from relevant categories and proposes a visualization to analyze the collaborative interactions of remote users. The setup involved two users in different locations sitting opposite to each other in a mixed reality scene, who had to collaborate to correctly position the three real and virtual pieces of a puzzle. The visualization encodes the interactions with puzzle pieces as vertical lines, along with the waveform representation of a recorded conversation between users in a timeline.

Chapter 6 (RO 2.2) presents a visualization approach for understanding the competitive and collaborative behaviors of agents trained using artificial intelligence. It uses a specific virtual game environment, Pommerman [17], where the objective of two opposing teams is to kill enemies by dropping bombs and strategic maneuvers. The Pommerman community uses the environment as a testbed for developing artificial intelligence techniques. The proposed timeline visualization encodes each agent in a row, events as colored markers, and interactions with vertical lines. We evaluate the approach in a study with community members and visual analytics experts.

In Chapter 7 (RO 2.2 and 2.3), to explore the aspect of movement in entity interactions, we focus on the Flatland [18] environment, a virtual simulation as a testbed for scheduling trains on fixed tracks. Modeling it as a multi-agent system, artificial intelligence techniques are used to explore novel and robust solutions for scheduling trains. The environment has much larger map sizes and more agents than Pommerman. Our visualization approach facilitates the analysis of collaborative interactions exhibited by trains. The approach enables exploring the spatial aspect of interactions by node-link representation of abstracted movements, as well as linked analysis of time and space. Insights from winning solutions of competition and feedback from experts provide a formal evaluation of the approach.

Part III – Conclusion: Group Dynamics

In Chapter 8 (RO 1.1, 1.2, and 2.2), we extend Set Streams and propose a visualization for an integrated analysis of the two aspects in group dynamics: memberships in overlapping groups and entity interactions. The visualization shows insights from two application examples. The first example analyzes the evolving business portfolio of companies along with the interactions between them, e.g., company acquisitions and licensing agreements. The second example shows the dynamic collaborative interactions in scientific research fields.

Finally, in Chapter 9, we discuss a few works in progress on enriched analysis of group dynamics (e.g., interactions with the environment and using multimodal channels; RO 2.3). The thesis then summarizes the proposed approaches and concludes with a discussion of their limitations and the future outlook.

2 BACKGROUND

Scenarios involving multiple entities are difficult to understand. The complexities involved in the dynamic processes are reflected in the behavior exhibited by the entities. For instance, entities change their group memberships, interact with each other, and move around while performing other actions. Various visualization techniques have been proposed to help understand individual aspects of such behaviors. However, a joint visual analysis is often complex and not usually addressed by a single approach. Hence, we will discuss the related visualizations for overlapping groups and evolving interactions.

2.1 VISUALIZING ENTITY GROUPS

Entities in the same group share some commonalities. Several visualizations have been proposed to represent the membership of entities in one or more groups. Since most research focuses on showing static group structures, we first discuss them to understand the challenges in representing temporal changes in overlapping groups. Next, focusing on time, we discuss visualizations for dynamic communities in graphs and timeline-based representations of entity groups. Finally, we highlight the few proposed approaches for visualizing dynamic overlapping sets.

2.1.1 *Visualizing Static Group Structures*

Connections between entities are often modeled as edges in a graph, which is usually visualized through a node-link diagram, an adjacency matrix, or a hybrid representation of both (e.g., NodeTrix [22]). Since entities may belong to multiple groups simultaneously, the groups overlap. Vehlow et al. [23] surveyed visualization techniques for group structures in static graphs. With a focus on explicitly encoding the group membership of nodes, they structured the literature into four types of representations: visual node attributes (coloring the nodes), juxtaposed (separate or attached representation of groups with the graph), superimposed (e.g., line and contour overlay), and embedded (in the graph layout: node-link or hybrid).

Figure 2.1a shows an example where the group information is embedded as colored pies in the individual nodes of a graph. Other techniques use lines and contour overlays in a node-link diagram (e.g., [25, 26]). However, it becomes difficult to get an overview and compare the overlaps among groups (RO 1.1 and

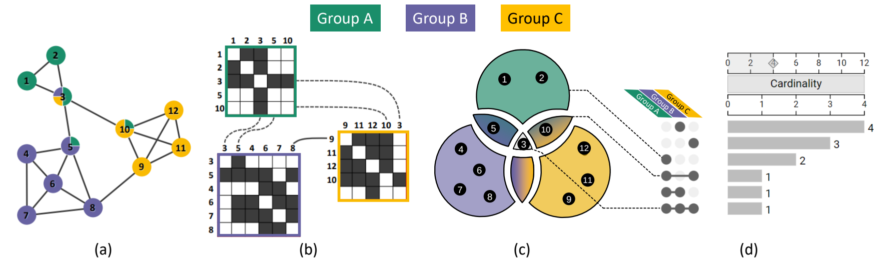


Figure 2.1: Visualizing group membership in graphs through (a) colored pies in the nodes of a node-link diagram, or (b) with duplicate nodes in hybrid graph representation, as surveyed by Vehlow et al. [23]. Modeling the groups as sets (c) Venn diagram shows contained elements in different intersections, while (d) UpSet [24] partitions the set intersections as rows.

1.2). For instance, the size or number of nodes in specific group overlaps has to be visually computed by counting the number of nodes with a particular combination of colors. In hybrid graph representations, as shown in Figure 2.1b, the graph communities, based on group membership, could be shown as adjacency matrices. However, the encoding duplicates the elements with multiple group memberships in different matrices and marks them with dotted lines (e.g., node 3). The encoding partly addresses the challenge of estimating the size of overlap among two groups by judging the number of dotted lines between two matrices. However, analyzing overlaps involving more than two groups has to be inferred by tracking individual nodes and counting them.

On the contrary, set visualizations explicitly model the overlaps as intersections and encode them in the design. Several static set visualizations have been proposed, which have been organized in a survey by Alsallakh et al. [19]. For instance, a Venn diagram in Figure 2.1c shows different set intersections (separated with some gap between them for clarity) and the contained elements. Regarding scalability, similar visualizations can show a high number of elements, e.g., through aggregation in area-proportional Euler diagrams. However, explicitly showing the set intersections as an overlapping region between different shapes does not scale well beyond 4–5 sets. Hence, for better scalability, the representation of the group structure needs to be redesigned. Using partitions to flatten the group structure, UpSet visualization [24] in Figure 2.1d, shows each exclusive set intersection (each region in the Venn diagram; connecting lines show the mapping), as a row. The cardinality of each intersection is then encoded as a horizontal bar chart in the respective row. Although analyzing a group as a single unit (all exclusive set intersections involving a set) becomes challenging, the interactions partly help by, e.g., merging the corresponding rows on demand. However, since the designs already result in a dense representation with some visual clutter or do not show time, ex-

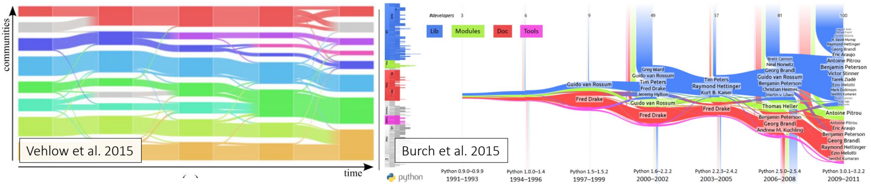


Figure 2.2: Timeline approaches to show: evolving graph communities as colored ribbons [27] (left), or the temporal changes as colored flows (packages in Python code) of a river [28] (right).

tending them to embed dynamic group memberships is not straightforward and remains challenging (RO 1).

2.1.2 *Visualizing Dynamic Communities in Graphs*

Conventionally, group dynamics is analyzed by modeling the evolving relationships of entities as a dynamic graph (edges represent relationships between entities modeled as nodes [29]). This already results in complex network structures, which are then studied using statistical analysis on the identified clusters or communities (e.g., [30, 31, 32, 33]). However, such analysis methods are limited as they do not support exploratory analysis of the group dynamics. Since visualizations offer the flexibility of exploratory analysis, e.g., through rich interactions and linked views, the existing dynamic graph visualizations could be used to study group dynamics.

Highlighting the representation of time as a major distinguishing feature, Beck et al. [34] surveyed dynamic graph visualizations and focused on graph representations on a timeline. For instance, Vehlow et al. [27] use a matrix layout, as shown in Figure 2.2 (left), visualizing the graph communities on a vertical axis and mark the temporal changes as ribbons between adjacent columns. Animation-based techniques have also been useful for temporal analysis. For instance, GraphDiaries [35] proposed staged transitions in an animation of a node-link diagram, where node color identifies its membership in a group. While animated approaches tend to suit better for the analysis of adjacent timesteps [36], timeline-based approaches might offer better support for tasks involving more than two timesteps, e.g., understanding changes across all or a small range of timesteps (RO 1.1 and 1.2).

Several graph layouts and embedding techniques have been explored. For instance, a recent technique visualizes a timeline of the changing graph network by focusing on the structure and graph communities via diachronic node embeddings [37]. Including spatial attributes in the analysis, Landesberger et al. [38] propose an approach that visualizes dynamic categorical data along with the geographic context in linked views. However, in the examples presented above and techniques included in the survey, a graph node may belong to only one group at a time. Generally, it is still difficult to embed the overlapping group membership information through a visual attribute, as they are already used in the dense design of the graphs to show its structure (connections) and temporal changes. Abstraction partly helps to free up a visual channel but usually comes at a cost (e.g., aggregation hiding the graph structure). Hence, the examples have limitations in visualizing evolving overlapping groups in dynamic graphs (RO 1).

Finally, since dynamic graphs encode changes in only one behavioral aspect, i.e., a specific relation among nodes modeled as edges, the dynamic graph data model is not sufficient to capture the other aspects of group behavior, e.g., entity interactions. Although multi-layer graph visualizations could represent multiple aspects simultaneously as different layers of edges between nodes, as seen by diverse approaches in a survey [39], embedding temporal changes in the already dense visual design remains challenging.

2.1.3 *Timeline-based Visualizations of Dynamic Groups*

Timelines are an effective way to read and understand temporal data. Usually, timelines use the horizontal or vertical axis to map time from left to right or top to bottom in a linear layout. In the timeline, the changes in the groups are represented through diverse encodings, e.g., glyphs for discrete timesteps and flows for a continuous timescale. In the context of representing dynamic groups, several timeline-based visualizations have been proposed.

Using a river metaphor, ThemeRiver [40] proposed a technique to represent the temporal changes as the flow of a river from left to right. Themes, which can be considered as groups in our context, are represented as river currents, whose vertical width denotes the number of contained entities at a specific time. The technique inspired several other approaches, explicitly representing groups of entities. For instance, as shown in Figure 2.2 (right), Developer Rivers [28] represents changes in the contributions to modules (groups) of Python code by developers (entities), as colored flows. Similarly, to show the evolution of software, a storyline-like visualization was proposed, which clusters the individual lines representing developers contributing to a module (group) of a code repository [41].

Other examples of flow-based techniques include CiteRivers [42], which represents the timeline of citing papers by grouping them (e.g., by conference tracks: VAST, SciVis, and InfoVis) in colored currents of the river. The City on the River (CotR) visualization [43] shows an individual contributor (group) as a colored stream, while the collection of products on which the contributions were made (entities) are encoded as the width of the streams on a horizontal timeline. Byron and Wattenberg [44] proposed Streamgraphs, a design inspired by ThemeRiver, but focusing on both legibility and aesthetics.

The approaches discussed above enable analyzing the evolution of groups over time. Often the entities contained in a group are aggregated and not shown individually, but the details can be fetched when needed. However, these timeline-

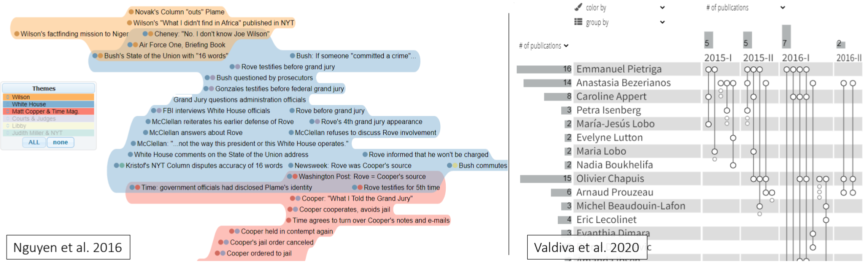

Novak's Column "outs" Plame
Wilson's "What I didn't find in Africa" published in NYT
Wilson's factfinding mission to Niger
Cheney: "No. I don't know Joe Wilson"
Air Force One, Briefing Book
Bush's State of the Union with "16 words"
Bush: If someone "committed a crime"...
Rove testifies before grand jury
Bush questioned by prosecutors
Gonzales testifies before federal grand jury
Grand Jury questions administration officials
FBI interviews White House officials
Rove before grand jury
McClellan reiterates his earlier defense of Rove
Rove's 4th grand jury appearance
McClellan answers about Rove
McClellan refuses to discuss Rove involvement
McClellan: "...not the way this president or this White House operates."
White House comments on the State of the Union address
Rove informed that he won't be charged
Kristof's NYT Column disputes accuracy of 16 words
Newsweek: Rove was Cooper's source
Bush commutes
Washington Post: Rove = Cooper's source
Time: government officials had disclosed Plame's identity
Rove testifies for 5th time
Cooper: "What I Told the Grand Jury"
Cooper cooperates, avoids jail
Time agrees to turn over Cooper's notes and e-mails
Cooper held in contempt again
Cooper's jail order canceled
Cooper ordered to jail
Themes
Wilson
White House
Matt Copper & Time Mag.
Courts & Judges
Libby
Judith Miller & NYT
ALL
none
Nguyen et al. 2016
color by
group by
# of publications
# of publications
5 5 7 2
2015-I 2015-II 2016-I 2016-II
16 Emmanuel Pietriga
14 Anastasia Bezerianos
8 Caroline Appert
3 Petra Isenberg
2 María-Jesús Lobo
2 Evelyne Lutton
2 Maria Lobo
2 Nadia Boukhelifa
15 Olivier Chapuis
6 Arnaud Prouzeau
3 Michel Beaudouin-Lafon
4 Eric Lecolinet
3 Evanthia Dimara
Valdiva et al. 2020

Figure 2.3: Timeline-based dynamic set visualizations encoding membership in a set through background color [45] (left), and a set as a hyperedge connecting multiple entities in rows [46] (right)

.

based designs assume a single group membership of entities and do not show the overlap among groups. As a result, a temporal overview of entities with multiple group memberships is not reflected in the timeline (RO 1.1).

2.1.4 *Dynamic Set Visualizations*

Since sets inherently focus on the overlap (set intersections), we look toward set visualization approaches. While several static set visualizations have been proposed, as surveyed by Alsallakh et al. [19], only a few approaches exist for representing dynamic sets. Bubble Sets [47] places the elements in a timeline, while colored overlays represent their membership in a set. But an element belongs only to one set at a time. Extending the idea further, TimeSets [45] (Figure 2.3 left), shows color-coded sets as an overlay on the horizontal timeline and vertically positions the elements (events) in set intersections at the boundaries of the respective sets. However, it becomes difficult to provide a clear overview, especially of intersections involving three or more sets (RO 1.1).

Since a hyperedge can model the relation between two or more hypernodes, it can be considered equivalent to a set. Hence, visualizations to represent dynamic hypergraphs become relevant for the discussion. A short survey on hypergraph visualizations compares a few relevant techniques [21]. For instance, Valdivia et al. [46] proposed PAOHVis, a technique to visualize such dynamic hypergraphs. As shown in Figure 2.3 (right), people modeled as elements are placed in individual rows, while columns represent timesteps. A hyperedge between elements, representing co-authorship in a scientific article, encoded as a vertical black line, can be considered as a set. Another example is HyperStorylines [48], which combines PAOHVis and storyline visualizations to represent temporal hypergraphs. The horizontal axis shows a storyline, while a hyperedge is encoded as a vertical line connecting two or more entities. Although these visualizations are useful for tracking the memberships of individual entities across time, gaining a

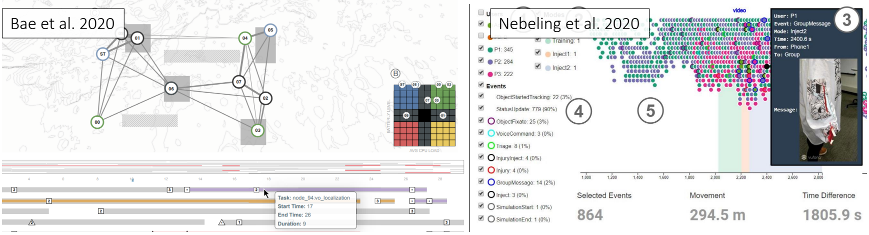


Figure 2.4: MOSAIC Viewer by Bae et al. [52] shows communication interactions between multiple robots in science expeditions (left). MRAT visualization by Nebeling et al. [53] encoding events from multiple participants involved in an augmented reality crisis simulation exercise (right).

temporal overview of the specific overlaps and comparing different entity groups remains challenging (RO 1.1 and 1.2).

2.2 Analysis of Entity Interactions, Event Sequences, and Movement

Apart from the group memberships of entities, the dynamic behavior is also reflected through the interactions among entities. The sequence of important events leading to or following the interactions provides context. Additionally, the movement of entities is crucial in certain scenarios where the spatial context clarifies the strategy behind entity interactions. Hence, we include the visualization of these aspects in the discussion.

2.2.1 *Analyzing Entity Interactions*

Jordan and Henderson [49] lays the foundation for research discussions on analyzing interactions between people and their surroundings. They describe interaction analysis as *"an interdisciplinary method for the empirical investigation of the interaction of human beings with each other and with objects in their environment."* Follow-up works have tried to understand group behavior through the interactions between involved entities. For instance, Jakobsen and Hornbaek [50] analyzed the collaboration behavior between two people using a large wall-size display and explored insights on proxemics and multimodal interactions. Similarly, Tang [51] has also recorded the collaborative sessions and did the video analysis to get insights on the usage of hand gestures, hand drawings, and spatial orientation in the drawing space and how they affect the group collaboration. Although these studies highlight the importance of analyzing interactions for understanding group behavior, they relied on watching the recorded session videos. They did not leverage visualizations to analyze the entity interactions.

An interaction among entities could take place in several forms, depending on the scenario. For instance, Blascheck et al. analyzed the interactions of a user with an interface together with the data from a think-aloud protocol [54]. Treating the interactions and thinking-aloud actions as events, their visualization showed the temporal sequence of the events along with the respective regions of the interface for every participant. This helped them to compare the behavior of different participants. Similar visualizations show the interactions of software developers with the integrated development environment (IDE) [55, 56, 57]. These visualizations show time on a horizontal axis, whereas the vertical axis represents different source code files and dialog boxes of the IDE. However, these approaches visualize interactions of an individual entity.

Focusing on scenarios with multiple entities, VISTACO [58] shows collaborative interactions on tabletop displays. It plots the participant interactions with the tabletop display and shows the drag operation (on the tabletop) as paths in the proposed visualization. Exploring the different collaboration styles, Isenberg et al. [59] analyzed group collaboration around a tabletop display with a visual analytics system to explore document collections. They propose eight different collaboration styles, ranging from close to loose collaboration among entities. Although they do not propose any visualizations for the analysis, the collaboration styles are generalizable to other scenarios.

In some scenarios, communication between entities is a prominent indicator of group dynamics, e.g., business meetings. Hence, communication can also be considered as an interaction among entities. A timeline-based approach—the horizontal axis shows time, while entities are positioned on the vertical axis—has been proposed to analyze group dynamics of human interactions in meetings [60]. Similarly, the MOSAIC Viewer [52] (Figure 2.4 left) technique shows a summary of communication between autonomous robots as links in an aggregated node-link diagram. However, we still lack research on exploring visual channels to encode interactions that facilitate analysis of the group behaviors along with other contextual attributes (e.g., movement; RO2).

2.2.2 *Analyzing Group Dynamics through Event Sequences*

Events could provide context in the immediate vicinity of the entity interactions. Also, interactions among entities could be modeled as event sequences. A recent survey by Guo et al. [16] organizes the visualizations for event sequence data along four proposed dimensions: data scales, analysis techniques, visual representations, and user interactions. Although all the dimensions are relevant to our discussion, we cover most related works in the context of understanding group dynamics, especially entity interactions.

Event sequences are often represented in a timeline. For instance, MOSAIC Viewer [52] (Figure 2.4 left) shows a timeline view at the bottom, with each row encoding the important events and their timespan for individual autonomous robots.

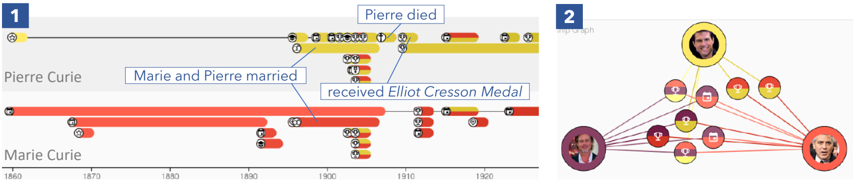


Figure 2.5: A timeline by Latif et al. [5] shows the sequence of shared events (both individual and shared) in the connected lives of prominent personalities. A graph summarizes the connections between them.

Together with an overview of communication interaction among the robots, the visualization helps the robot operators investigate the status of robots in their synchronized worldview. Another example of a different scenario shows an event-centric visualization of a crisis simulation exercise in augmented reality involving multiple participants. The technique MRAT [53] (Figure 2.4 right) encodes events associated with each participant and a 2D projection of their location on the floor plan. Similarly, ReLive by Hubenschmid et al. [61], bridging in-situ and ex-situ visual analysis of mixed reality user sessions, shows a timeline of events. However, the examples mentioned above do not explicitly encode the group dynamics and interactions among participants.

In games, players need to interact, strategize, and perform actions in a sequence. Since game environments usually involve complex dynamic behaviors, we look at game user research and find that event sequence visualizations have been used to analyze the behavioral player data. A comprehensive overview of gameplay visualizations can be found in a survey by Wallner and Kriglstein [62].

We look into specific examples of visualizations for multi-player games, especially because the players need to compete, collaborate, or both in these scenarios. Understanding these behaviors requires a sequential analysis of their actions in the game environment and related events. To analyze the competitive gameplay behavior in racket-based sports involving two players, various systems (e.g., TacticFlow [63], RASIPAM [64], and RallyComparator [65]) have been proposed. They model player actions in a rally as a multivariate event sequence and use pattern mining to discover the tactics in the sequences. Then, the discovered tactics are visualized by either encoding the aligned aggregated multivariate event sequences in a flow visualization ([63]) or by aggregating the tactics and showing them with intuitive glyphs ([64, 65]). However, such techniques rely on pattern mining and provide an overview of gameplay behavior by usually focusing on the frequently occurring patterns, which may not necessarily be the most interesting ones as certain patterns may naturally occur more frequently than others [66].

Beyond games, visualization of event sequences has also attracted attention in Human-Computer Interaction and other domains. For example, in visualizing event sequences of a student's learning path (e.g., [67]), patient's electronic health records, and personal histories (e.g., [68, 69]). These approaches, however, do not explicitly show interactions with objects and link event sequences to each other to help better understand temporal action-reaction relations among multiple entities. However, some techniques model interactions as an event and represent them as a sequence. For example, works concerned with visualizing dynamics between multiple entities, such as in conversations (e.g., [70]) or interactions and mobility in interior spaces (e.g., [71]).

Focusing on the analysis of collaboration, VICPAM [72] proposes a timeline visualization with important events showing how people work in multiple display environments. It contains a timeline and spatial view along with the raw videos. Although, the interactions were not explicitly encoded and had to be inferred from the events. Although encoding interactions in a 2D timeline visualization is challenging, few approaches have tried addressing them. For instance, Latif et al. [5] propose a horizontal timeline with individual rows reserved for each entity (Figure 2.5 left). Each entity is assigned a color. Shared events are encoded in the rows as horizontal lines with the respective color of an entity. They summarized the relationships between entities using an aggregated node-link diagram with additional nodes for events, as shown in Figure 2.5 (right). Some of these approaches could be adapted to provide a temporal summary of interactions between entities. However, we lack a systematic understanding of the mapping between visual encodings and relevant aspects of group behaviors to help design appropriate visualizations (RO 2.1 and 2.2).

2.2.3 *Spatio-temporal Analysis of Group Behaviors*

The movement of an entity is an important aspect to analyze while understanding its behavior. A survey [73] classifies the visualizations for movement into four categories: (a) looking at individual trajectories (e.g., [74, 75, 76]), (b) segments of the trajectories to explore local movement patterns (e.g., [77, 78, 79]), (c) aggregation of multiple movement trajectories (e.g., [80, 81, 82]), and (d) investigating movement in context (e.g., [83, 84]). We focus on spatio-temporal analysis in dynamic scenarios involving multiple entities, where an entity's movement is affected by the behavior of others.

It has been found that the type of medium offering extended workspace (e.g., physical wall-sized displays or virtual reality) affects the spatial organization and behavior of users [85]. Moreover, the scenario setup also affects the spatial behavior. In a cooperative game of Miners on a large wall display [86], the behavior of a human group of participants was visualized through linked views, including a timeline of events, heatmaps for gaze points, touch interactions, and the usage of physical space, along with a raw video of the recorded session. The visualizations helped reveal useful insights, such as that participants often moved together, indicating coordination among them. In contrast, in other scenarios, territoriality was observed when the collaborative behavior of participants was visualized, indicat-

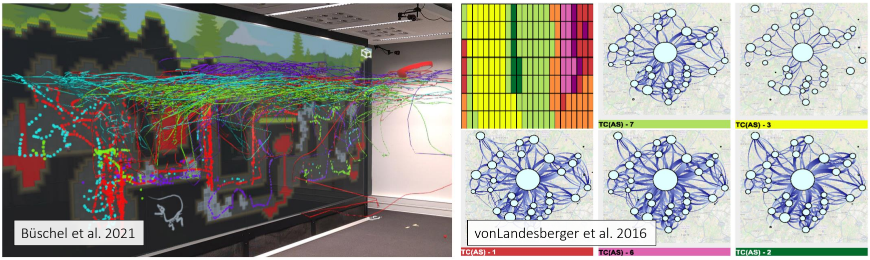


Figure 2.6: MIRIA [89] visualizes the recorded spatial interactions with a large display using augmented reality for in-situ analysis (left). MobilityGraphs [90], a scalable technique to show aggregated movement as juxtaposed graphs (right).

ing the strategy to divide the area to achieve the common goal [58]. Exploring the difference in scenario setup, a multi-player game Pac-Many [87] was proposed to be played on a large wall display. In two scenarios, collaboration and competition between participants were observed by manually analyzing recorded session videos. In the collaborative scenario, teammates focused on small regions and not on the entire map of the game, indicating territoriality [88]. However, while competing, players tried to look at all regions to be aware of the opponent and sometimes physically blocked them.

Regarding mixed reality scenarios, Billinghurst et al. [91] discusses the different collaboration setups (co-located and distant users) and the need to understand group interaction behaviors. Kloiber et al. [92] visualized the motion data in virtual reality while preserving the context during analysis. They represent the trajectory data together with a timeline visualization and enrich it by adding the key events or actions of the users. Extending the idea further, Büschel et al. [89] propose MIRIA, a mixed reality toolkit (for multimodal inputs) to support the in-situ analysis of spatial interactions in multi-display and augmented reality environments, as shown in Figure 2.6 (left). The toolkit implements 3D trajectory plots, 3D trails, 2D heatmaps, 2D scatterplots, 2D point plots, media views, and a 2D event timeline for visual analysis.

Visualizing the mass movement of entities has been helpful in understanding their mobility behaviors. For instance, Guo [93] visualized the migration flow on a map, preserving the spatial context with directed or undirected edges. Since movement data may contain other related attributes, they may be plotted on parallel coordinates for an integrated analysis. Extending the idea further with reduced visual clutter, the MobilityGraphs [90] technique was proposed, as shown in Figure 2.6 (right). It aggregates the movement in nearby locations and represents a region as a node, while movement between the regions is modeled as a weighted directed link. The node-link graph is overlaid on a map to preserve the context. The technique effectively shows patterns in longer time spans. Other recent approaches also focused on scalability by sampling and abstracting the multivariate movement data (e.g., [94]). We take inspiration from these examples to show entity movement in visualizations for group dynamics.

Apart from encoding the movement, spatial queries have been proposed to understand the proxemics and interactions of entities in indoor spaces. For instance, EXCITE [95] supports specifying spatial queries as events (e.g., the distance between person 1 and 2 is less than 2 meters) and shows the instances of its occurrences in a row of a horizontal timeline. The query also supports specifying certain events (e.g., holding a tablet facing towards a large display) and adding a row in the timeline. While some interactions can be considered as events, it does not encode interactions explicitly. The idea was extended by enabling the visual specification of compound queries to analyze group interactions, location, and proximity in the EagleView technique [96]. These spatial query techniques can be used to analyze specific situations in the behavior of entities acting in a group while using a linked event timeline to integrate space and time (RO 2.3).

Part I

DYNAMIC OVERLAPPING GROUPS

3 DYNAMIC ENTITY MEMBERSHIPS IN OVERLAPPING SETS AS STREAMS

A group of entities can be mathematically modeled as a set. In doing so, a group overlap is represented by a set intersection. Hence, to understand the group behavior, the challenge of visualizing dynamic overlapping groups can be formulated as a dynamic set visualization problem. Such a visualization would be applicable in several scenarios—e.g., developers (entities) contributing to multiple modules (sets) of a code repository— revealing insights about the behavior of entities through their changing memberships (RO 1). For instance, specialist developers who initially contributed to specific code modules later became generalists; or developers who consistently contributed to only a few modules.

In this chapter, we propose a novel dynamic set visualization approach, Set Streams, that encodes the temporal changes of element-set memberships as streams on a timeline going from left to right. We first explain the design considerations that guided the development of the approach in Section 3.1. Next, we elaborate on the design and visual encodings in the Set Streams approach (Section 3.2). To test the applicability and effectiveness, we use the approach to investigate data from three scenarios and analyze feedback from visualization experts in Section 3.3. Finally, we end with a short discussion on the advantages and limitations of the approach in analyzing group behavior (Section 3.4).

3.1 DESIGN CONSIDERATIONS

To guide the design of the approach, we derive three considerations that are specific to the dynamic set-typed data but relevant across different scenarios. They are derived based on the existing literature (e.g., the visual analysis tasks for static set-typed data [19]), generic visualization needs for temporal data, and our experience in designing approaches for discrete data.

3.1.1 *DC1: Get a Temporal Overview*

Animations and timelines are the two broadly used approaches to visualize temporal data. Generally, animation-based approaches are easy to understand individual temporal changes. However, it has non-trivial limitations, e.g., as highlighted by Tversky et al. [97]: *"Animations are often too complex or too fast to be accurately perceived. Moreover, many continuous events are conceived of as sequences of discrete steps."* Partly addressing the limitations, stepwise animation techniques for dynamic sets have been proposed, which aim to reduce the user's gaze shift by optimizing the grouping and sequence of the temporal changes [98]. However, the animation still demands a high cognitive load to understand the overall changes across timesteps.

Timeline-based approaches have the potential to address the limitations, which in the context of dynamic graph visualization is evident in the recent research shift from animation to exploring timeline-based designs [34]. Specific to the dynamic set-typed data, the design of the visualization should clearly represent the changes in set structure, e.g., the evolving overlap among sets (RO 1.1). To represent temporal changes, existing techniques have explored different encodings, e.g., color, shape, and opacity. In our approach, we choose to provide a static overview of changes on a linear timeline, where discrete timesteps are represented as columns.

3.1.2 *DC2: Follow Elements across Time*

At its core, the source of the temporal changes in the set structure (e.g., growing or shrinking overlaps) is the dynamic memberships of individual elements in sets. Hence, for an in-depth analysis, the design should support users in analyzing the changing set memberships of an element or a group of elements over time. We use the design from alluvial diagrams, which encodes the temporal changes in groups of items as ribbons or bands. The diagrams are intuitive and flexible. Hence, they have been adapted to visualize temporal information in diverse contexts, e.g., CiteRivers [42], ThemeRiver [40], Developer Rivers [28], and AOI Rivers [99], among others. However, we first need an efficient encoding to visually represent the set structure, or the overlaps, while avoiding visual clutter on the timeline. Hence, taking inspiration from the UpSet design [24], we flatten the set overlap and map each non-empty overlapping region, or in other words, every non-empty exclusive set intersection to a row in the timeline visualization. As a result, we can apply flow-based encoding and represent the temporal changes as branching and merging streams from left to right on a timeline, enabling users to follow elements across time (RO 1.1).

3.1.3 *DC3: Compare Groups of Elements*

Supporting analysis of temporal changes in a group of elements sharing some commonality is important. A clearer understanding of the relative similarities and differences in the temporal trend emerges when multiple groups (elements of interest) are explicitly compared during the visual analysis. Since such comparison is valuable to infer insights based on the changing set memberships, we design the approach with an aim to enable the comparison of at least two groups of elements. However, since streams do not explicitly show individual elements, we need an alternate encoding to highlight the temporal changes in elements of a group. Superposing [100] colored streams on top with the height proportional to the number of elements in a selected group, we can compare the two selected groups of elements. Hence, we use two colors to identify elements exclusively in the two groups, one for elements in both groups and another for the non-selected elements. Additionally, the design needs an integrated, simple, and intuitive query mechanism to specify the two groups of elements for comparison (RO 1.2).

3.2 SET STREAMS VISUALIZATION APPROACH

The design considerations already provide a rough sketch of visualization for dynamic overlapping sets. In this section, we describe the developed approach by first formalizing the data model, elaborating the encodings used in the timeline visualization, integrated query-based selection interactions, and other linked views. The full interface of the developed prototype is shown in Figure 3.1 and Figure 3.3.

3.2.1 *Data Model*

To get an intuitive understanding, if set $A = \{x, y\}$ and $B = \{y, z\}$, then there are three non-empty **exclusive set intersections** with elements: [only in A] $= \{x\}$, [only in B] $= \{z\}$, and [only in $A \cap B$] $= \{y\}$. Formalizing the data model, let $F = \{S_1, S_2, \ldots, S_n\}$ be a family of n **base sets**, where each set $S_i \subset E$ contains elements from a **universe** E. The power set of this **family of base sets** $\mathcal{P}(F)$ describes every possible combination of these sets with $|\mathcal{P}(F)| = 2^n$. For each family of sets $F_j \in \mathcal{P}(F)$, we can compute a **set intersection** (or overlap) of the contained sets $I(F_j) = \bigcap_{S \in F_j} S$. An element $e \in E$ might belong in multiple intersections, for instance, $e \in S_1, S_2, S_3 \Rightarrow e \in I(\{S_1, S_2\}), I(\{S_2, S_3\})$. As we want to avoid repeated encoding in different intersections for showing the flow of elements (**DC2**), we further define an **exclusive set intersection**

$$\bar{I}_F(F_j) = \{e \in E | (\forall S \in F_j : e \in S) \land (\forall S' \in F \setminus F_j : e \notin S')\}$$

(i.e., each element included is not included in another base set that is not considered in the intersection). In the above example ($e \in S_1, S_2, S_3$), $e \notin \bar{I}_F(\{S_1, S_2\})$, $\bar{I}(\{S_2, S_3\})$, but $e \in \bar{I}_F(\{S_1, S_2, S_3\})$ if also $e \notin S_4, S_5, \ldots, S_n$. Note that if an element $e \in E$ is not contained in any $S_i \in F$, then $e \in \bar{I}_F(\emptyset)$. Hence, the set of all exclusive intersections over F forms a partition of E. To model temporal changes with m timesteps, we introduce a **sequence of families of base sets** $\mathcal{F} = (F^1, F^2, \ldots, F^m)$ over the same universe E, where each $F^k = \{S_1^k, S_2^k, \ldots, S_n^k\}$ consists of the same number of n base sets. Through this, we can follow a set, which represents a categorical attribute of the elements, across time.

3.2.2 *Timeline Visualization*

To provide a static overview of the temporal changes in set memberships (**DC1**), we use a grid-like structure for the timeline view. The discrete timesteps are shown as columns, while each row represents a non-empty exclusive set intersection. Moreover, to identify the sets involved in an exclusive intersection, $\bar{I}_F(F_j)$, the involved base sets ($S_i \in F$), are encoded with a darker color and connected with a horizontal line (■━■ □ □), while others are shown via light-colored rectangles. As shown in Figure 3.1, by default the rows are ordered based on the number of sets contained in the respective exclusive intersection, and the group of rows is labeled appropriately, e.g., *Exclusive 1-set intersections*, followed by *Exclusive 2-set intersections*, etc. A rectangular box (▬) represents a cell of the grid structure, encoding the number of elements in a non-empty exclusive set intersection (row) as the height of the gray bar, for a specific timestep (column). While hovering a cell, the corresponding row, participating base sets in the exclusive set intersection, and the timestep label are highlighted (Figure 3.2). Also, the number of contained elements is shown as a tooltip.

To represent the temporal changes in set memberships of an element, we draw streams connecting the respective nodes in the adjacent timestep columns. Formally, two nodes representing the exclusive intersections $\bar{I}_{F^k}(F_j)$ and $\bar{I}_{F^{k+1}}(F_{j'})$ are connected if $w := |\bar{I}_{F^k}(F_j) \cap \bar{I}_{F^{k+1}}(F_{j'})| > 0$. The number of elements w undergoing the same transition between the two timesteps is encoded as the width of the stream. The elements that were added in a timestep or appeared for the first time are shown by the stream originating from the top edge on the left of the respective column. Likewise, elements that were deleted or do not belong to any base set in the next timesteps are shown by downward streams ending on the right of the respective column. Since elements may also skip belonging to any set for some timesteps, they are encoded as upward-going streams until above the first row and rejoining in the respective later timestep column. To avoid clutter while drawing the streams entering or exiting a cell, we sort them based on the vertical position of their destination. Therefore, streams going up are placed first, followed by the horizontal streams connecting the cell in the same row, and finally, those that are going down.

To convey the number of elements in set overlaps at each timestep, a cardinality distribution is shown by a histogram above the respective column. Every cardinality c of the family of sets involved in the respective exclusive intersections is mapped to a bar in the histogram. The bar height shows the number of elements in the respective exclusive intersections at timestep k that have cardinality c:

$$|\bigcup_{F_j \in \mathcal{P}(F^k):|F_j|=c} \bar{I}_{F^k}(F_j)|$$

To abstract the rows of individual exclusive set intersections having the same cardinality c, they can be interactively aggregated in the grid as a single row. The approach also integrates sorting rows based on different parameters. For instance, by the decreasing order of contained elements in a selected timestep k ($|\bar{I}_{F^k}(F_j)|$) or summed across all timesteps ($\sum_{k=1}^{m} |\bar{I}_{F^k}(F_j)|$). Additionally, to prioritize the stability—consistency of memberships in each exclusive intersection—the rows can be sorted based on the consistency of the contained elements (Figure 3.3). It is computed as the ratio of elements that do not change their membership to all elements contained in the intersection, and then the ratios are summed across all timesteps. Moreover, to reduce clutter, rows with most membership transitions (or streams) between them can be positioned closer together. To do this, we compute the number of elements switching their membership between two exclusive intersections across all timesteps and use a greedy approach: place the exclusive intersection with the highest number of incoming elements first and then always place the most similar one next. Finally, the rows can also be sorted by assigning priority to a base set, which first lists all exclusive intersections in which the set is involved and sorts them with increasing cardinality, followed by the remaining set intersections in default order. Figure 3.2 demonstrates the aggregation and sorting features where all rows with cardinality three are aggregated and then sorted based on their stability.

3.2.3 *Query-based Selection*

To enable visual comparison of elements changing their memberships in different groups (**DC3**), we first propose a simple query mechanism to select and specify the element groups. Next, to avoid using new encodings to show the selected element groups, we overlay the results on the timeline visualization as streams. We experimented with different solutions and decided to use an intuitive approach by embedding the query into one short sentence with different selectable parameters. Hence, the query can be read as a natural language text, which makes it self-explanatory while still providing enough flexibility to specify several queries. The query includes the following parameters:

- *Set Operation:* Three types of set operations are available in the drop-down selection field. It includes two types of set intersections: the non-exclusive intersection $I(F_j)$ and the exclusive intersection $\overline{I}_F(F_j)$. Moreover, a set union operation is also available, which is defined as $U(F_j) = \bigcup_{S \in F_j} S$.

- *Base Sets:* To mark a base set and apply it to the query, the respective checkbox can be selected. Any combination of the base sets can be activated to specify F_j.

- *Timestep:* Finally, users choose the timestep k relevant to the query from a drop-down selection field.

To restrict the visual clutter, we limit the comparison between two element groups, allowing the specification of only two queries. The resulting groups of elements—*Group A* and *Group B*—are encoded with a unique color (A: orange; B: green). Some elements may belong to both groups. Hence, to discern these shared elements, we use a third color (A+B: black). Next, the respective streams in the timeline visualization are highlighted using the respective colors.

Apart from using the parameterized sentence for query specification, the two groups of elements can be selected by interacting with the visual elements. For instance, clicking a cell in the grid structure automatically creates a query with the respective exclusive intersection (row), $\bar{I}_{F^k}(F_j^k)$, involved base sets (F_j^k), and the timestep k (column). The query is applied to specify Group A by default, or Group B by toggling the respective radio button on the left of the query sentence. Similarly, clicking on a stream selects the elements undergoing the transition between two timesteps. As an edge selection goes beyond what can be represented in the query form described above, we switch to an alternative sentence describing the selection.

3.2.4 *Linked Views*

Since the streams in the timeline visualization aggregate the elements, we integrate a list of all elements as an additional linked view on the right. A search bar at the top can be used to quickly search for a specific element, marking the search results with a blue dot in the list (Figure 3.3). On the left of each element's name, colored bars indicate if the element belongs to any of the selected groups. The list is ordered to first show the search results, followed by the elements in both selected groups (Group A+B), then the ones only in Group A, only in Group B, and finally listing all the remaining elements in the dataset (universe E). Within each category, alphabetic order is the secondary sorting criterion.

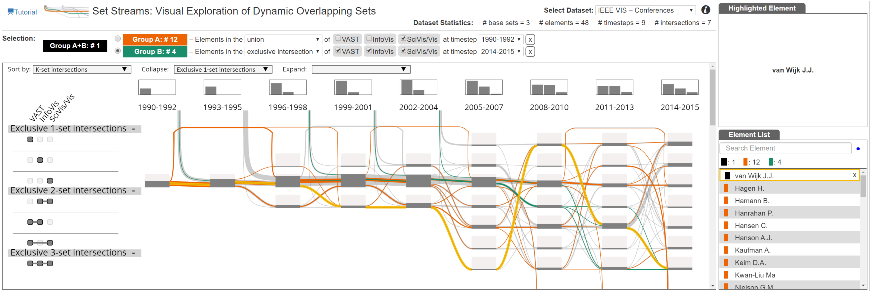
3.3 APPLICATION EXAMPLES AND EXPERT FEEDBACK

To assess the usefulness of the proposed visualization approach, we analyze datasets from three diverse scenarios. We derive insights about the temporal trends in the expertise of researchers, software development, and changes during the training of a machine learning classifier. To complement the assessment, we include feedback from expert users, who evaluated the approach using the discussed examples. The implemented prototype with application examples is available in the supplemental material [101] and hosted online[1].

3.3.1 *Expertise of Researchers*

The meta-data of scientific publications includes, e.g., a list of authors, keywords describing the contributions, publication venue, and year of publication. We model the authors or researchers as elements, while the research themes (indicated by keywords or publication venue) as sets. A publication by a researcher on particular theme(s) determines its membership in the respective set(s). Doing so, using our approach, we can visually analyze the temporal trends, for instance, growing

[1] (Accessed May 2023) https://s-agarwl.github.io/setstreams



Set Streams: Visual Exploration of Dynamic Overlapping Sets

Select Dataset: IEEE VIS – Conferences

Dataset Statistics: # base sets = 3 # elements = 48 # timesteps = 9 # intersections = 7

Selection: Group A+B: # 1

○ Group A: # 12 – Elements in the union of ☐VAST ☐InfoVis ☑SciVis/Vis at timestep 1990-1992

● Group B: # 4 – Elements in the exclusive intersection of ☑VAST ☑InfoVis ☑SciVis/Vis at timestep 2014-2015

Sort by: K-set intersections

Collapse: Exclusive 1-set intersections

Expand:

Highlighted Element: van Wijk J.J.

Element List: ■: 1 ■: 12 ■: 4

- van Wijk J.J.
- Hagen H.
- Hamann B.
- Hanrahan P.
- Hansen C.
- Hanson A.J.
- Kaufman A.
- Keim D.A.
- Kwan-Liu Ma
- Nielson G.M.

Figure 3.1: A screenshot of the Set Streams interface shows a grid structure for the main visualization: exclusive set intersections are encoded rows, while timesteps are represented as columns. The changes in set membership are visualized by streams from left to right. The *IEEE VIS* dataset shows the most frequent authors (elements) contributing to the three conference tracks (sets). Two groups have been selected for comparison: orange-colored streams mark the group of *SciVis/Vis* contributors in 1990–1992, while green shows the authors who have contributed to all three tracks in 2014–2015. The common elements of these two groups of authors are shown in black; author *van Wijk* is additionally highlighted in yellow on user selection.

interest in new topics or shifting research interest of the experts. Modeling subcommunities in a research field as sets, we can understand how they evolved over time. We discuss one example of the *IEEE VIS* conference series and its tracks, but the tool includes two other examples, a keyword-based dataset of publications in the visualization research and a broader computer science community.

Dataset. We use the *Visualization Publication Data Collection* [102], which contains data of *IEEE VIS* between 1990–2015 and a total of 2752 publications. The different tracks of the conference series represent the sets: *SciVis/Vis* (the original *Vis* conference is considered here as a predecessor of *SciVis*), *InfoVis*, and *VAST*. Authors of publications become the elements. Focusing on established researchers, we filter the authors with a minimum of 15 publications, resulting in 48 authors. Instead of yearly trends, we aggregate publications over periods of three years each to one timestep.

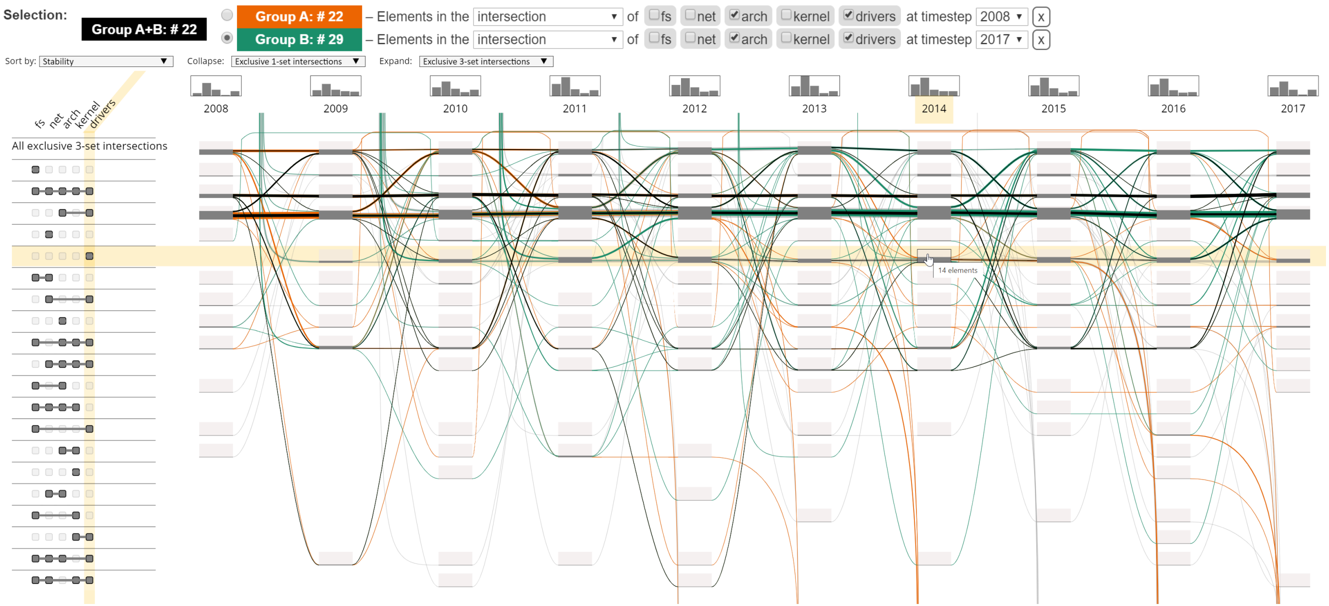
Findings. Analyzing the historical development of the *IEEE VIS* conference series, as shown in Figure 3.1, we observe that it originated as the *SciVis/Vis* track, later branching into the *InfoVis* track (1996–1998) and *VAST* track (2005–2007). Since only a few streams from the top join in the exclusive intersections involving *VAST* in the timestep 2005–2007, we infer that the established researchers included in the dataset started publishing in the track from the beginning. But, they did not start exclusively publishing in the *VAST* track. Several diagonal connections in the later timesteps (from 2005) generally indicate the evolving interests of researchers between the tracks. Also, the presence of all combinations of tracks indicates that authors published exclusively in all combinations, which shows that the community is not segregated due to these tracks. However, the publications by the established researchers are not balanced in the combinations. Next, we compare specific groups of researchers. As marked in Figure 3.1, we select the group of early contributors in Group A (marked in orange) and recent generalists in Group B (marked in green). Among the included researchers in the dataset, the two groups share only one researcher (*van Wijk*, in black, but also highlighted in yellow). Observing the few orange edges ending at the bottom, we infer that many early researchers are still active in the community. Looking at the green incoming edges to the exclusive intersection of *SciVis/Vis* in early timesteps, we infer that the recent generalists joined the community by contributing exclusively to *SciVis/Vis*.

3.3.2 *Software Evolution*

Applications of analyzing the evolution of a software project include keeping involved people up-to-date with the development, easily identifying the specialists or experts, or understanding the historical changes in code structure while boarding a new team. Such scenarios have been researched by others with stream-based visualizations, for instance, focusing on code structure [103] or also discussing developer contributions [28].

Dataset. We collect the data of commits by software developers into the five main modules of the *Linux* code repository from 2008 to 2017. The code modules are modeled as sets, and the developers who contributed to the module (committed changes in the module files) are the elements of the respective set. Ignoring those who rarely committed to the repository, we filtered those developers who made at least 100 commits, resulting in 111 contributors. To focus our analysis, as shown in Figure 3.2, we aggregate all exclusive 3-set intersections and sort the rows based on consistent contributions or stability of the exclusive intersections.

Findings. In spite of analyzing a long timespan, from Figure 3.2 we observe that many developers have consistently contributed to the code repository across ten years (2008–2017). Specifically, there is a group of generalists who contributed to all modules (the third row) throughout the whole period (*Jiri Kosina*, *Greg Kroah-Hartman*, *David S. Miller*, *Linus Torvalds*, and *Al Viro*). The high number of elements in the exclusive intersection of *arch* and *drivers* (the fourth row) indicates a significant overlap of developers contributing to both modules simultaneously and consistently. For comparison, we select the early contributors in the (non-exclusive) intersection of the modules as Group A (marked in orange for 2008), while the late contributors as Group B (marked in green for 2017). We can infer that 22 committers were common in the two groups. Most of these common committers (black) are either generalists or contributed only in *drivers* and *arch* (thick black lines in respective rows). The histogram above timestep columns consistently shows a spike in the number of elements in 2-set intersections, which supports the gained insight.



Selection: Group A+B: # 22

○ Group A: # 22 – Elements in the intersection ▾ of □fs □net ☑arch □kernel ☑drivers at timestep 2008 ▾ x

● Group B: # 29 – Elements in the intersection ▾ of □fs □net ☑arch □kernel ☑drivers at timestep 2017 ▾ x

Sort by: Stability ▼ Collapse: Exclusive 1-set intersections ▼ Expand: Exclusive 3-set intersections ▼

Figure 3.2: Software evolution: Contributors of the Linux project form the elements, assigned to different parts of the system (*fs*, *net*, *arch*, *kernel*, and *drivers*) according to their code contributions within a year (2008–2017). Exclusive 3-set intersections are aggregated. Rows are sorted based on stability. Intersections of *arch* and *drivers* are selected for the years 2008 (orange) as Group A and 2017 (green) as Group B for comparison between the two contributor groups.

Among specialists who focus on fewer code modules, we see that most of them contributed to *drivers* and *fs* modules.

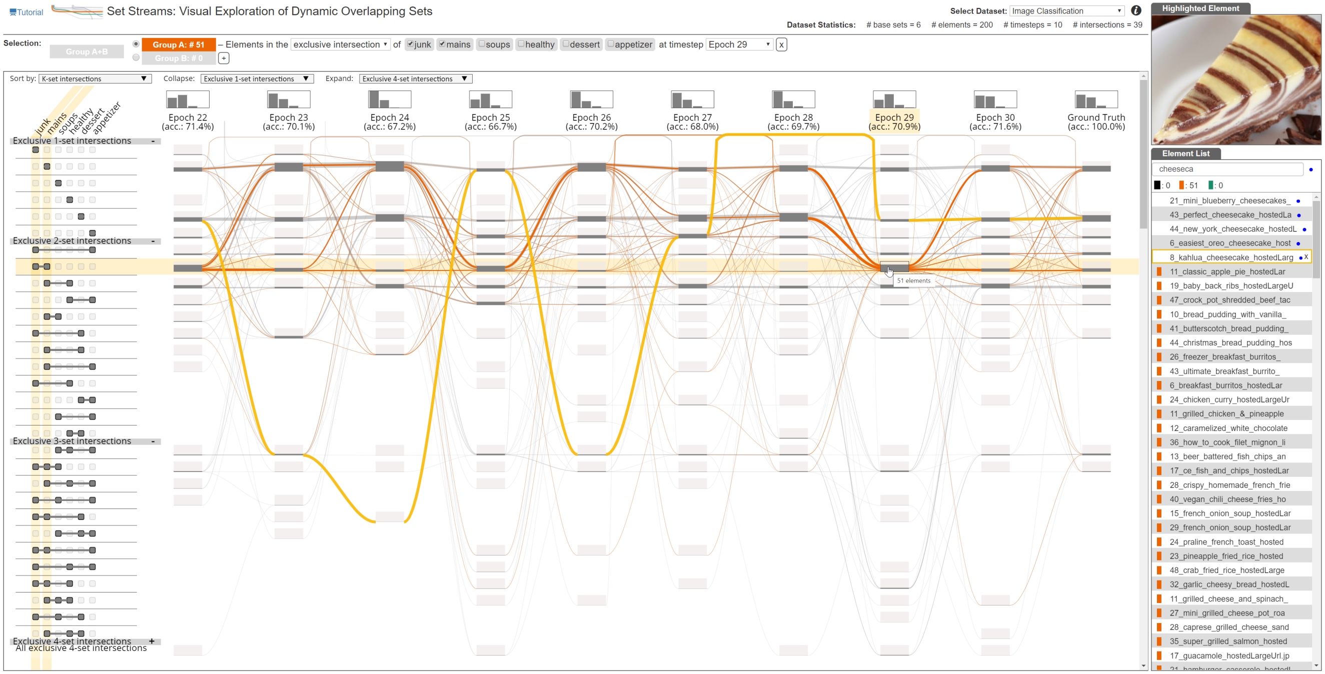
3.3.3 *Multi-label Classification*

In machine learning, the task of assigning labels, e.g., to images, is usually done by training models using *supervised classification* methods. In scenarios where multiple labels, modeled as sets, can be assigned to a data item (element), to show the labeling results, the overlap among sets needs to be visualized. In addition, when the machine learning developer wants to analyze the training process, a temporal component needs to be included in the visualization. A visual approach addressing these challenges can be used to explore valuable insights, e.g., data items that are hard to classify correctly by the model during training and the classes among which the classifier is most confused.

Dataset. We include a dataset from an image-classification scenario where images of food dishes need to be labeled in overlapping categories[2], e.g., a dish can be both *junk* food and a *main* dish. We model the labels as sets, while images are the elements. The dataset contains 200 images and 6 labels along with the labels predicted by a convolutional neural network at different training epochs. We focus on the last epochs, after which the training was stopped. To provide a basis for judging the correctness of predicted labels during each epoch, we add the computed accuracy in each timestep at the top and include the ground truth as the last column.

Findings. From Figure 3.3 we see that the accuracy of the model does not improve consistently with more training, but fluctuates around 70%. We can also observe the model's confusion in identifying dishes as only *mains* or both *mains* and *junk* across training epochs, which does not align with the change in accuracy since they appear in different epochs. It shows that even when accuracy indicates a quite stable transition, bigger changes might actually happen *behind the scenes*. To explore further, we select a group of images in the exclusive intersection of *junk* and *mains* in Epoch 29 (orange group in Figure 3.3). We observe: (i) the similarity of the labeling results with respect to this intersection in Epoch 22 and 25, even though they have different accuracy rates, (ii) only about half of the marked elements are correctly classified while the other images spread with respect to the ground truth, (iii) before, in Epoch 28, many of the marked elements were (mostly wrongly) classified as *dessert*. To highlight the predicted labels for an image across training epochs, we can left-click the image names in the element list. By doing so, we can find stable examples that are correctly classified in most epochs or other cases of more interest to the developers, e.g., the unstable outliers that exhibit the model's confusion prominently. For instance, the highlighted picture of a cheesecake in Figure 3.3 (yellow) should be classified as *dessert*, but jumped around

[2] (Accessed May 2023) https://github.com/thatbrguy/Multilabel-Classification



Set Streams: Visual Exploration of Dynamic Overlapping Sets

Select Dataset: Image Classification

Dataset Statistics: # base sets = 6 # elements = 200 # timesteps = 10 # intersections = 39

Selection: Group A+B ● Group A: # 51 – Elements in the exclusive intersection of ☑junk ☑mains □soups □healthy □dessert □appetizer at timestep Epoch 29 ○ Group B: # 0

Sort by: K-set intersections Collapse: Exclusive 1-set intersections Expand: Exclusive 4-set intersections

Tutorial
junk
mains
soups
healthy
dessert
appetizer
Exclusive 1-set intersections
Exclusive 2-set intersections
Exclusive 3-set intersections
Exclusive 4-set intersections
All exclusive 4-set intersections
Epoch 22 (acc.: 71.4%)
Epoch 23 (acc.: 70.1%)
Epoch 24 (acc.: 67.2%)
Epoch 25 (acc.: 66.7%)
Epoch 26 (acc.: 70.2%)
Epoch 27 (acc.: 68.0%)
Epoch 28 (acc.: 69.7%)
Epoch 29 (acc.: 70.9%)
Epoch 30 (acc.: 71.6%)
Ground Truth (acc.: 100.0%)
51 elements
Highlighted Element
Element List
cheeseca
: 0 : 51 : 0
21_mini_blueberry_cheesecakes_
43_perfect_cheesecake_hostedLa
44_new_york_cheesecake_hostedL
6_easiest_oreo_cheesecake_host
8_kahlua_cheesecake_hostedLarg
11_classic_apple_pie_hostedLar
19_baby_back_ribs_hostedLargeU
47_crock_pot_shredded_beef_tac
10_bread_pudding_with_vanilla_
41_butterscotch_bread_pudding_
44_christmas_bread_pudding_hos
26_freezer_breakfast_burritos_
43_ultimate_breakfast_burrito_
6_breakfast_burritos_hostedLar
24_chicken_curry_hostedLargeUr
11_grilled_chicken_&_pineapple
12_caramelized_white_chocolate
36_how_to_cook_filet_mignon_li
13_beer_battered_fish_chips_an
17_ce_fish_and_chips_hostedLar
28_crispy_homemade_french_frie
40_vegan_chili_cheese_fries_ho
15_french_onion_soup_hostedLar
29_french_onion_soup_hostedLar
24_praline_french_toast_hosted
23_pineapple_fried_rice_hosted
48_crab_fried_rice_hostedLarge
32_garlic_cheesy_bread_hostedL
11_grilled_cheese_and_spinach_
27_mini_grilled_cheese_pot_roa
28_caprese_grilled_cheese_sand
35_super_grilled_salmon_hosted
17_guacamole_hostedLargeUrl.jp

Figure 3.3: Training of a multi-label classifier for image classification: The timeline shows the predicted labels for each image across various epochs (training stages) of the classifier. The last column represents the ground-truth assignment of labels. The exclusive intersection of labels *junk* and *mains* is selected in Epoch 29 (orange).

different exclusive intersections until, in epochs 29 and 30, finally being classified correctly.

3.3.4 *Expert Feedback*

To confirm the validity and usefulness of the findings described above, as well as to receive general feedback, we invited different expert users to test the approach as part of an online study. We used our professional network to recruit at least one expert for each application example. In total, 5 expert users (E1–E5) participated, E1 and E2 having significant experience in bibliographic analysis, E2 and E3 in software evolution research, and E5 in training classifiers. We provided them with the tool (including a tutorial) and a preliminary version of the text (including everything in the chapter except this section on expert feedback). As part of an online questionnaire, we initially asked them to go through the tutorial and explore the tool before starting the questionnaire, which all participants confirmed. The first task was to reproduce the observations described in the result section of the respective application example (each participant was assigned the application example fitting his/her expertise). Second, the experts were asked to extend the analysis of the application example and report the insights found. Then, the experts were invited to comment on (a) the sorting and aggregation and (b) the query-based selection capabilities of Set Streams. The study concluded with options to provide overall feedback on the most and least useful features, missing information or features in the tool, additional analysis tasks that could be performed, and additional remarks. The study was designed to take about 60 minutes. The questionnaire and all responses are available in the supplementary material [101].

Reproduced and Extended Findings. All experts commented that they were able to reproduce most of our findings, while some of them had problems due to clutter (E1) or relating the findings with the figure (E4). The experts were able to extend the analysis and discovered: the most common exclusive set intersections (*InfoVis* and *VAST* – E1; *arch*, *kernel*, and *module* – E3), stability of element memberships in at least one set across all timesteps (E2), uncommon exclusive set intersections (*kernel* and *drivers* – E4; *net* and *arch* – E4), and unusual behavior of some elements not belonging to any set in a few timesteps (E5). Although the experts found additional insights, one expert (E3) commented that the analysis becomes difficult without being closely associated with the dataset and without having a specific question in mind.

Functionalities. All experts liked the functionality of sorting rows: based on a timestep (E4), based on the ground truth to get an overview of the dataset (E5), and based on the priority of specific sets (E1). The experts also liked the aggregation functionality and commented that it helped to reduce the clutter (E1). However, E1 and E4 also mentioned that using dropdown lists for these features is not intuitive and E3 suggested that it could be improved by providing more information on the sorting criteria via tooltip. All five experts liked the query-based selection feature: it is self-explanatory and flexible (E1), formulating queries by selection is easy (E5), and it is useful to compare two groups of elements (E1 and E4). However, the experts also commented that differentiating gray and colored edges becomes difficult (E3 and E5) and too many choices could confuse the users (E1 and E2).

Overall Feedback. Three experts (E1, E4, and E5) mentioned that they found the query-based selection to be most useful. The experts suggested using natural language to describe the selection (E4), tooltips to convey extra information (E1 and E3), and integrating domain-specific information such as the classification accuracy of an element at a specific epoch (E5). Expert E2 commented that Set Streams already supports too many tasks and could be limited to reduce visual complexity. Experts E3 and E4 also mentioned that it is difficult to perform free-exploration tasks in Set Streams. Expert E5 suggested stabilizing the ordering of names in the element list and providing a search feature. We have already incorporated the latter two suggestions for the final version.

3.4 DISCUSSION

The approach visualizes one aspect of the complex group behavior, namely, evolving memberships of elements in one or more sets. In this context, as evident from the derived insights and expert feedback, the proposed approach helps to understand the behavior due to the membership of entities in overlapping groups. The insights demonstrate that the timeline design is effective in providing an overview of the changing membership in sets (RO 1.1), e.g., the journey of researchers from publishing in one theme to later contributing in multiple tracks or analyzing the instability in groups of images which have been most confusing for a machine learning classifier. Moreover, the design integrates a comparison between two selected groups of entities revealing useful insights (RO 1.2), e.g., comparing groups of software developers who were early and recent contributors in specific modules (*arch* and *drivers*) of a code repository. Such comparative analysis helped identify the consistent contributors (developers common in both groups) and understand their code contribution behavior. In this section we reflect on the important characteristics of the proposed generic design of the dynamic set visualization approach, discussing the strengths, limitations, and ideas for future work.

3.4.1 *Data Ordering and Aggregation*

The implemented default order and grouping of the grid rows in our approach is based on the cardinality of sets involved in an exclusive intersection. This provides a contextual basis while interpreting the branching and merging streams (**DC2**). For instance, the horizontal streams indicate unchanged memberships of elements, while those taking additional set memberships are shown as streams connecting the rows below in the next timestep. Although crossing streams produce visual clutter, the proposed sorting criteria reduced the number of crossings. Moreover, other suggested reordering methods could also be implemented, e.g., for other alluvial diagrams [27]. Along with ordering, abstracting the details by aggregating intersections could help focus on the specific aspects, as demonstrated by aggregating *Exclusive* k*—set intersections* in the examples. Since the partition of elements in rows is mutually exclusive, any combination of selected rows for aggregation would be theoretically possible. However, designing an intuitive and easy-to-use interface for such a versatile aggregation mechanism would be challenging.

3.4.2 *Scalability and Generalizability*

Although the approach can manage to show a few hundred elements, reading and selecting the thin streams representing a small number of elements becomes difficult. The scalability with respect to time is similar to other Sankey-based diagrams on a timeline (**DC1**), being able to show about a dozen timesteps. The scalability gets affected with a higher number of base sets, due to the vertical scrolling to accommodate the exponentially high number of combinations, especially when they are non-empty. This is a typical challenge for set visualizations. Partly addressing the challenge, interactive filtering techniques can be explored to select and focus on the important set intersections only. Regarding the generalizability of the approach, except for the details panel, the proposed encodings are applicable in diverse scenarios, as reflected in the application examples. However, the full potential of the approach can be realized by tailoring it to the scenario, e.g., by integrating application-specific statistics.

3.4.3 *Temporal Trends and Details of Entities*

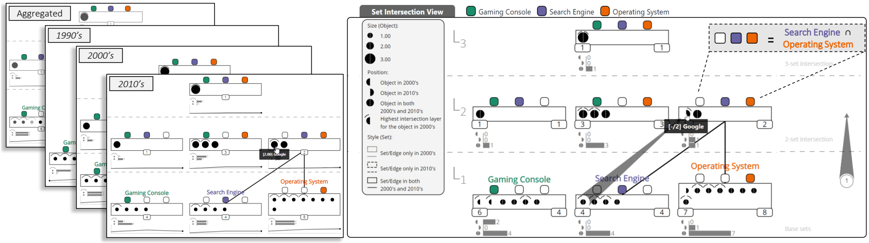
The evolving memberships of elements revealed insights about the sets and their intersections, e.g., the growing or shrinking popularity trend of particular sets or their combinations based on the number of contained elements. Apart from this, to explicitly compare an arbitrary group of elements (**DC3**), a query and selection mechanism was included. However, it is limited to comparing only two groups of elements simultaneously. Additionally, the approach does not model the membership weight of elements in a set. Consequently, the insights based on the differences between elements in the same sets cannot be perceived in the visualization, e.g., researchers publishing prominently only in *VAST* vs. other contributors in the same set. Hence, although the approach enables the exploration of patterns in dynamic group memberships, handling membership strength or set membership weight remains a challenge, which is addressed in the next chapter.

4 LAYERED SET INTERSECTION GRAPHS FOR ELEMENT-SET MEMBERSHIPS

Dynamic overlapping sets can be used to model many real-world scenarios. For instance, evolving business portfolios of companies or changing expertise of experts (e.g., researchers or software developers as elements) in diverse communities or code repository modules (sets). Since the changing memberships reflect an important aspect of the elements' behavior, dynamic set visualizations can be used to analyze the group dynamics (RO 1).

Focusing on scalability, most of the existing set visualizations aggregate the details of individual elements. For instance, as described in the previous chapter, Set Streams [104] is a dynamic set visualization technique to provide a temporal overview of element-set memberships as streams. The design aggregates the details of elements present in a set intersection at a given timestep by showing only the cardinality. However, aggregation hides the element details and differences among the elements do not get visualized. Analyzing such details and differences is relevant in the scenarios where they characterize and give detailed information about an element's membership in one or multiple sets. For instance, the number of contributions of an expert, modeled as the weight of its membership, indicate the level of expertise and prominence of his/her role in a community (e.g., senior researcher, repository moderator). However, we still lack techniques that (a) represent temporal changes in element memberships (RO 1.1) and (b) support a detailed comparison of changes between two timesteps (RO 1.2) while (c) encoding the details of element memberships in a set (e.g., element-set membership weight; RO 1.3).

This chapter describes a dynamic set visualization approach that visualizes the membership weight of an element in a set. The design is based on the layered set intersection graph, where each node represents a base set or an intersection of base sets, and edges represent direct subset relationships. The data model and the construction of a layered graph are described in Section 4.1, followed by the explanation of visual encodings for a static set, aggregated set, and diff representations in Section 4.2. The two application examples use the proposed technique to show the changes in researchers' fields of interest and the evolution of developer activities in a software project in Section 4.3. The chapter ends with a discussion on the scalability and generalizability of the technique (Section 4.4).



(a) Aggregated and static set intersection graphs

(b) *Diff* view between timesteps *2000's* and *2010's*

Figure 4.1: The toy dataset being visualized is about evolving business portfolios of companies (elements) across three types of products (sets). Each set is assigned a unique color; the intersections are shown as rectangular nodes, while contained elements are encoded as black circles in the node. The representation shows (a) aggregated or individual static set intersection graphs and (b) differences in set intersection graphs between two timesteps.

4.1 A TOY DATASET, DATA MODEL, AND SET INTERSECTION GRAPHS

The proposed visualization is based on a layered set intersection graph, where each node represents a base set or an intersection of base sets, while edges represent direct subset relationships. To understand it better, we first look at a toy dataset, the data model, and the construction of layered set intersection graphs.

4.1.1 *Visualizing a Toy Dataset*

Let us take an example of a real-world scenario: understanding the business strategy of companies by analyzing the temporal change of their product lines. The toy dataset contains information of 13 companies that make products across three categories over three decades (each decade is a timestep). Figure 4.1 visualizes the dataset where product categories are modeled as sets, while companies are elements. If a company makes a particular type of product, then it belongs to the corresponding set. Sets are shown as rectangles with their elements (circles) inside them. Nodes in layer n represent the intersections of n base sets, e.g., *Gaming Console* ∩ *Search Engine* ∩ *Operating System* is in layer L3 in Figure 4.1.

Elements in different sets are duplicated and represented in multiple nodes. Clarifying the used terminology, the number of base sets to which an element belongs in a timestep is called its **degree**. An element is **exclusive** to a base set if it does not belong to any other set. Formally, an element is exclusive to an intersection if its degree is the same as the number of base sets in an intersection, i.e., if the element does not belong to any set besides those in the intersection. We mark exclusive elements with a hat (⌒). These elements do not appear in any layer above.

Set intersection graphs are computed for each timestep. Aiming to provide different perspectives, the graphs are visualized individually or summarized by an

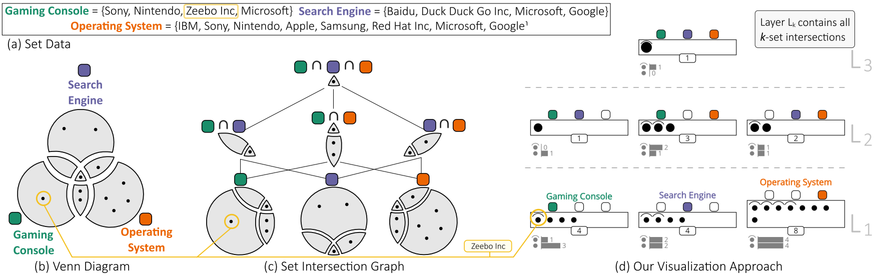

Gaming Console = {Sony, Nintendo, Zeebo Inc, Microsoft} Search Engine = {Baidu, Duck Duck Go Inc, Microsoft, Google}
Operating System = {IBM, Sony, Nintendo, Apple, Samsung, Red Hat Inc, Microsoft, Google}
(a) Set Data
Search Engine
Gaming Console
Operating System
(b) Venn Diagram
Zeebo Inc
(c) Set Intersection Graph
Layer L_k contains all k-set intersections
L_3
L_2
L_1
Gaming Console
Search Engine
Operating System
(d) Our Visualization Approach

Figure 4.2: The toy dataset for one timestep (*2010s*) with 3 types of products as sets (encoded with colors) and companies as elements, based on the type of products they manufacture. (a) The raw data. (b) Corresponding Venn diagram. (c) The constructed set intersection graph. (d) The proposed visualization approach represents a layered set intersection graph. Element *Zeebo Inc* is highlighted in all representations.

aggregated representation across all timesteps (Figure 4.1a). The exact changes between any two timesteps are shown by a *diff* view. For instance, in Figure 4.1b, *Google* is highlighted, and a tapered directed edge shows that the strategy of the company shifted in the *2010s* from *2000s* to producing both ***Search Engine*** and ***Operating System***.

Providing an intuitive understanding of our approach, Figure 4.2 shows its equivalence with a Venn diagram. The figure shows one set intersection graph for a single timestep (*2010s*) of the sample dataset. To explicitly represent all set overlaps, a set intersection graph (Figure 4.2c) is constructed for each timestep. As shown in Figure 4.2d, each set and intersection (graph node) is visualized by a rectangle, and elements of that set are black circles inside the rectangle. The figure highlights an element *Zeebo Inc* in different representations of the dataset.

The proposed set intersection graphs are inspired by concept lattices used in FCA (e.g., [105, 106]). The top concept of the lattice (topmost node in the graph shown in Figure 4.2c) has those elements that are present in all sets; the bottom concept has those elements that do not belong to any set. For most realistic datasets, the bottom concept is empty and thus not visualized in Figure 4.2c.

4.1.2 *Data Model*

The input for the proposed visualization approach is a non-empty set of m elements $E = \{e_1, e_2, \dots, e_m\}$ and a family of n base sets $F = \{S_1, S_2, \dots, S_n\}$ such that $S_i \subseteq E$. Each element can belong to one or more base sets, which undergo discrete temporal changes. The time dimension is represented as an ordered sequence of p timesteps $T = \langle t_1, t_2, \dots, t_p \rangle$ ($\forall k < k' : t_k < t_{k'}$). Depending on the application, the timesteps can be interpreted as snapshots or time ranges.

An $m \times n$ matrix W^k contains the data for timestep t_k where rows represent elements ($|E| = m$) and columns represent base sets ($|F| = n$). If cell $w_{ij}^k > 0$, then element e_i is in the base set S_j. The value of the cell determines the element's weight of membership in that set. If there is no meaningful weight definition available for a certain application, a binary value is sufficient (e.g., $w_{ij}^k \in \{0, 1\}$).

4.1.3 *Set Intersection Graphs*

A graph is constructed where every node represents a subset of F and contains as elements the intersection of elements of those base sets, e.g., the node for $X = \{S_1, S_3, S_4\}$ contains the elements in $S_1 \cap S_3 \cap S_4$. All the nodes in the graph are included that result in non-empty intersections (including a node for each base set but excluding a node for $\emptyset \subset F$). An edge is added between nodes X_i and X_j if the former is a direct subset of the latter, i.e., $X_i \cup \{S_k\} = X_j$ with $S_k \in F$.

The membership weight of an element e_i in node $X = \{S_j\}$ at timestep t_k is given by w_{ij}^k. For a set intersection, a weight has to be computed. We chose to sum up the element's weights of memberships across all the base sets in X. Formally, the weight of an element e_i in a vertex X at timestep t_k is computed as:

$$W'(e_i, X, t_k) = \sum w_{ij}^k, \quad \forall S_j \in X \qquad \text{(Equation 1)}$$

For instance, if an element is in base sets S_2 and S_4 with weights 1 and 2, then the element will be in $S_2 \cap S_4$ with weight 3. The process is repeated to get *set intersection graphs* $(G^1, G^2, ..., G^p)$, one per timestep.

4.1.4 *Aggregated Set Intersection Graph*

To compute the fixed layout across all timesteps, a *super-graph* [34] is built by merging the set intersection graphs of all timesteps. The super-graph nodes are formed by merging equivalent nodes across all set intersection graphs. Nodes from the set intersection graphs for timesteps t_k and $t_{k'}$ are equivalent if they represent the same subset of F, i.e., $X_i^k = X_{i'}^{k'}$. Edges that connect equivalent nodes at different timesteps are merged accordingly:

$$(X_i^k, X_j^k) \equiv (X_{i'}^{k'}, X_{j'}^{k'}) \Leftrightarrow X_i^k \equiv X_{i'}^{k'} \wedge X_j^k \equiv X_{j'}^{k'}$$

Hence, the resulting *super-graph* contains all nodes or edges that are present in at least one timestep, with equivalent nodes and edges contained only once (no replication).

4.2 THE LAYERED DYNAMIC SET VISUALIZATION

In the approach, first, a layered layout of set intersection graphs is computed (Section 4.2.1). The element memberships in individual timesteps are visualized through static set representation (Section 4.2.2). An aggregated representation provides an overview of the temporal changes (Section 4.2.3). Changes across any two selected timesteps are visualized by a *diff* view (Section 4.2.4). Interactive filtering and linked views (Section 4.2.5) enable an in-depth exploration of the set data. Figure 4.3 shows the prototype's interface.

4.2.1 *Layered Layout of Set Intersection Graphs*

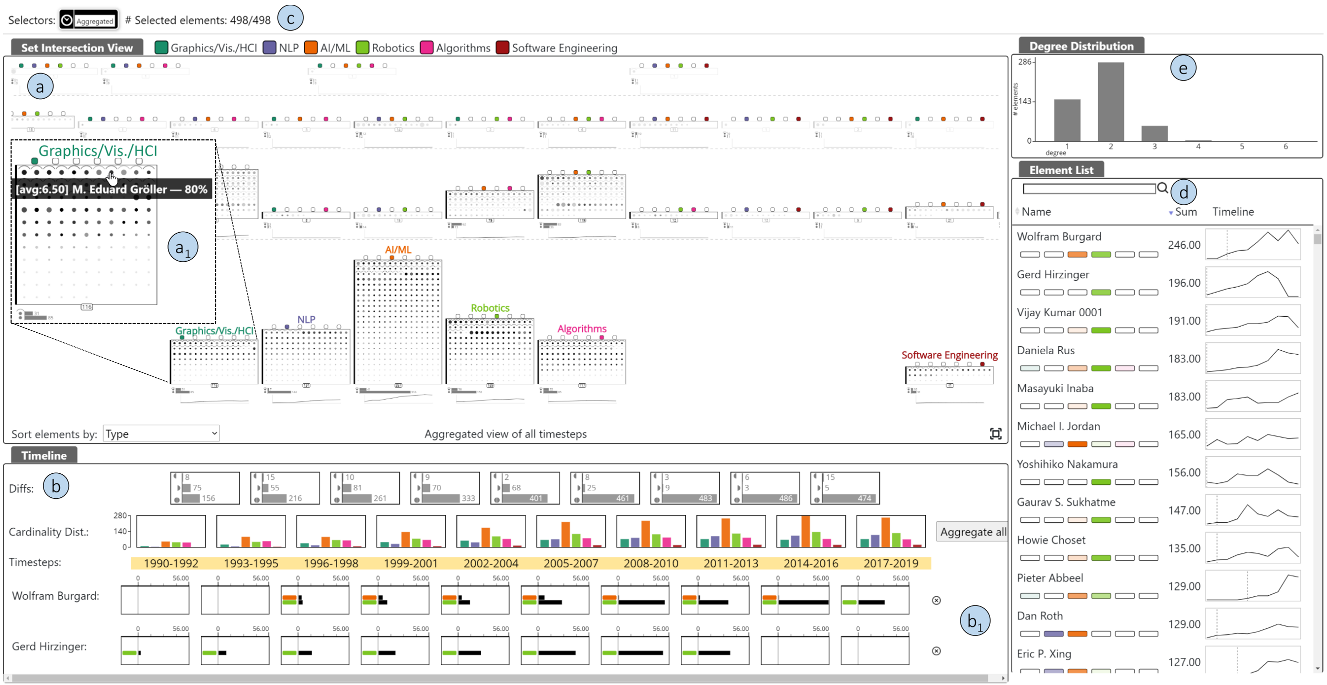
Having a stable layout in the visualization of dynamic graphs helps users preserve their *mental map* of the graph while flipping through its different versions [34]. Adopting this approach, the aggregated set intersection graph is used for a global layout, as it contains all nodes and edges that appear in at least one timestep. Hence, a fixed spatial position for each node in the graph is computed. This results in a stable and global layout of the set intersection graphs for each timestep.

The position of nodes and links in the aggregated graph is computed using a layered graph layout based on the Sugiyama algorithm [107]. Since a subset relation, as represented by edges, cannot form cycles, the cycle removal step of the Sugiyama algorithm is skipped. Furthermore, the topology-based layer computation is replaced by placing the nodes (intersections) in layers based on the number of participating sets in an intersection. Doing so gives semantics to the layers: a layer L_k contains only k-set intersections. For instance, a node representing [***Gaming Console*** ∩ ***Search Engine*** ∩ ***Operating System***] is assigned to L_3 (Figure 4.2d top). All nodes are ordered within layers according to the barycentric heuristic to minimize edge crossings.

4.2.2 *Static Set Representation*

Studies have shown that it is easy to perceive a group of related objects when they are drawn within a closed contour [108]. We use the design guideline to show elements belonging to a set inside a closed curve. Hence, in the approach, nodes of the set intersection graph are visualized as rectangles while contained elements as circles inside the rectangle. A circular shape was chosen because it can be divided into two distinguishable regions (semi-circles) required for the *diff* representation (Section 4.2.4). The area of a circle encodes the corresponding element's weight of membership (Equation 1). Small colored boxes identify the participating sets in an intersection (□🟪🟧 = ***Search Engine*** ∩ ***Operating System***).

Exclusive elements are marked with a hat (⌒) on top of the corresponding circles, indicating the highest layer in which an element is present (as the element does not have additional memberships). All circles inside a rectangle are ordered by their type (exclusive and non-exclusive). Within each type, elements are sorted by their decreasing membership weight. Such a representation enables us to see the distribution of elements within a set. Other criteria provided include sorting elements by their weight or name (Figure 4.3a bottom). Since for every circle, a



Selectors: Aggregated # Selected elements: 498/498

Sort elements by: Type

Aggregated view of all timesteps

Name	Sum
Wolfram Burgard	246.00
Gerd Hirzinger	196.00
Vijay Kumar 0001	191.00
Daniela Rus	183.00
Masayuki Inaba	183.00
Michael I. Jordan	165.00
Yoshihiko Nakamura	156.00
Gaurav S. Sukhatme	147.00
Howie Choset	135.00
Pieter Abbeel	129.00
Dan Roth	129.00
Eric P. Xing	127.00

Figure 4.3: The proposed dynamic set visualization technique with (a) a set intersection view in the middle (here, showing an aggregated representation), (b) a timeline view with cardinality distribution of base sets in each timestep and relevant statistics for diff representations, (c) applied filters, (d) a list of elements, and (e) degree distribution. The dataset shows research areas in Computer Science as sets (encoded in colors), researchers as set elements, and the number of publications in the respective area as element-set weight. The evolution chart of two selected researchers are shown in b_1 for later discussion (Section 4.3.1).

fixed amount of space (maximum area) is allocated and the width of all rectangular boxes is the same, the number of circles in a line inside every rectangle is fixed. Hence, the height of a node (rectangular box) indicates the cardinality of the corresponding intersection. The exact cardinality as a number is shown at the bottom of the node. The number of exclusive and non-exclusive elements is visualized by horizontal bars below the node (Figure 4.3a_1).

4.2.3 *Aggregated Set Representation*

To show an overview of temporal changes in set memberships across all timesteps, the approach integrates a time-aggregated set representation. The weight of each individual element is averaged across all timesteps in individual sets and intersections, which is then encoded via the area of the corresponding circle (Figure 4.3a_1). The visual representation is kept similar to static sets to minimize the need for memorizing additional visual encodings. The aggregated set representation (Figure 4.3a_1) uses opacity to encode the percentage of timesteps a certain visual element (i.e., a set, an element) is present; low opacity (gray) encodes a low percentage while high opacity (black) indicates that the visual element is present at all timesteps. For elements, filled circles in the aggregated set have varying opacity. Similarly, for a set, the left edge of the corresponding rectangular box is thickened and filled with the computed opacity level. In Figure 4.3a_1, the rectangle representing ***Graphics/Vis./HCI*** has a thick black left edge, which means the set was present in every timestep (had at least one element in every timestep).

The percentage of timesteps and average weights can be retrieved interactively on demand (on hovering), as shown in Figure 4.3a_1. Elements can gain or lose membership in sets over time. An element's maximum degree over time is marked with a hat marker in the rectangle(s) representing the corresponding set or their intersection(s). For instance, in Figure 4.3a_1, the highlighted circle shows researcher *M. Eduard Gröller* published an average of 6.5 articles per timestep within the field of ***Graphics/Vis./HCI*** in 80% of the timesteps. The hat marker shows that the researcher never published in any other field, together with ***Graphics/Vis./HCI*** in the same timestep.

4.2.4 *Diff Representation*

Explicitly pointing out differences between two timesteps of a dynamic graph helps in analyzing changes [109, 110, 111, 112]. Likewise, a *diff view* is integrated into the approach to represent exact changes in sets between any two timesteps. The visual elements are vertically divided into two parts. The left half shows data from the earlier timestep (t_k), while the right half shows data from the later timestep ($t_{k'}$ where $k < k'$). A circle (element) is split into two halves (left and right semi-circles) to indicate its presence in two timesteps, as shown in Figure 4.4a. Similarly, hat markers are also vertically separated into two arcs, showing the max-

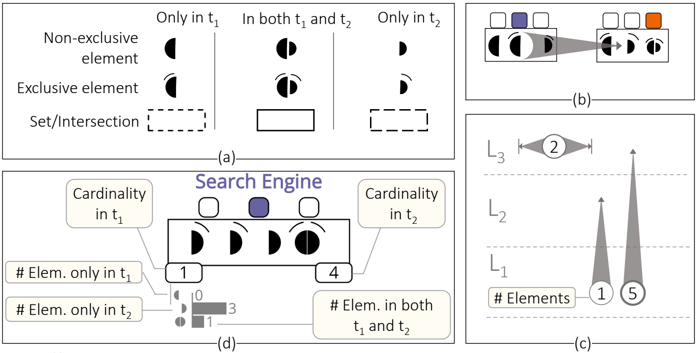


Figure 4.4: *Diff* between t_1 and t_2 showing: (a) visual encodings for elements (circles) and sets (rectangles), (b) a change in set membership by a tapered edge, (c) a group of elements undergoing similar changes by summary edges, and (d) *diff* view of the set *Search Engine* with annotations.

imum degree of an element in the two timesteps. Likewise, the set cardinalities at two timesteps are shown near the bottom left and bottom right corners of the corresponding rectangular box (Figure 4.4d). After experimenting with several designs, we finally chose to display rectangular boxes in the *diff view* with different stroke patterns to represent their existence only in t_k with a dotted border, only in $t_{k'}$ with a dashed border, and in both timesteps with a continuous border (Figure 4.4a).

The tapered edges explicitly encode an element's membership changes, which are available on-demand to reduce clutter. As shown in Figure 4.4b, a tapered edge highlights the shift in the element's membership from *Search Engine* to *Operating System*. The tapered edge shown in the figure is horizontal. If the element's degree (number of base sets it belongs to) changes across two timesteps, the tapered edge is drawn between two layers. As shown in Figure 4.1b, the inter-layer tapered edge shows that *Google* gained membership of the *Operating System* set at a later timestep. Summary edges abstract tapered edges with the same source and destination layers and show the number of elements inside a circular base at the origin. For instance, from Figure 4.4c we can infer that five elements with degree 1 (source is L_1) in t_1 gained membership in two additional sets in t_2 because their degree became 3 (destination is L_3).

Three cases arise based on the presence of elements and existence of sets in two timesteps: (i) present only in the earlier timestep (t_k), (ii) only in the later timestep ($t_{k'}$), and (iii) present in both timesteps. These cases are distinguishable through our chosen encodings, as shown in Figure 4.4a. For each set, the number of elements in the three cases is visualized by horizontal bars beneath the corresponding rectangular boxes (Figure 4.4d). Inside a rectangular box, the elements are primarily ordered by the three cases, while secondary ordering is on their weight of membership. The ordering helps to see the distribution of change in the membership of elements inside each set and intersection.

4.2.5 *Linked Views, Filters, and Interactions*

Besides the *set intersection view* given by the layered graph, we integrate other views to visualize details of set data and provide filters with interactions, supporting in-depth visual analysis. The video in supplemental material [113] shows the working of linked views, filters, and interactions in action.

A **Timeline** shows all the labeled timesteps ordered chronologically and drawn in a horizontal layout (Figure 4.3b). Small rectangles above each timestep label contain colored vertical bars to indicate the cardinality of base sets. The ability to select the timesteps allows easy navigation between different timesteps. Keyboard navigation with arrow keys is also supported. Temporal aggregation can be done by clicking the *'Aggregate all'* button. The selection of two timesteps for *diff view* requires clicking any two timesteps while holding the *Ctrl* key. Additionally, the *diff* view between adjacent timesteps can be retrieved by selecting the rectangles in the *'Diffs'* row above. Each rectangle in the row contains three horizontal bars showing the number of elements present: only in the left timestep, only in the right timestep, and in both timesteps.

An **Evolution Chart** of an element is a series of rectangles in a row placed below the timestep labels (Figure 4.3b_1). Each rectangle shows an element's weight of membership in sets for one timestep, encoded as horizontal bars. With this representation, the evolution chart enables a comparison between elements. It is drawn on demand when an element is selected (Figure 4.8).

An **Element List** shows a list of elements as rows (after applying current filters) on the right side of the interface (Figure 4.3d). Each row in the list represents one element with additional details: the name of the element, the sum of its membership weights in all sets among selected timestep(s), and a timeline showing a temporal variation of its cumulative membership weights ( ). The vertical dashed line in the timeline marks the timestep when the element first appeared. In the *static view*, colored boxes in each row indicate the membership of an element in the corresponding sets (▭ ▬ ▭). In the *diff view*, each box is subdivided vertically into two halves (▭ ◧ ◨) and filled according to the presence of the element in the corresponding sets across two timesteps. Whereas in the *aggregated view*, the opacity of color indicates the percentage of timesteps the element is present in the corresponding base set (▭ ▬ ▬). Clicking on a row selects the element, highlights it in the *set intersection view*, and draws its *evolution chart*.

The **Degree Distribution Chart** shows a distribution of filtered elements in terms of the degree of individual elements (Figure 4.3e). Existing visualizations such as RadialSets [114] have shown the usefulness of degree distributions in the analysis of set data. We extend the idea by showing degree distribution for any group of elements that fulfill the selection criteria. In the *diff view*, each degree bar is vertically split into two bars, one for each timestep.

Filters: The ability to query and filter a group of elements based on visual selection is a powerful way to analyze sets. We integrate a mechanism where a user can simultaneously select: (i) timesteps, (ii) sets and intersections, (iii) *summary edges*, and (iv) degree of elements. Each selection acts as a filter that returns a group of elements fulfilling the selected criteria. The applied filters are represented as tags (e.g., ⏲ 2010's , 🏷 Search Engine ✕ , ⬆ L 1 to L2 ✕ , and Ⓓ degree=1 ✕) above the *set intersection view* (Figure 4.3c). The resulting group of elements is linked to the element list (Figure 4.3d) and degree distribution (Figure 4.3e) components of the interface and updates them accordingly.

Other Interactions: The *set intersection view* (Figure 4.3a) supports panning and zooming. Hovering over any element shows connecting edges between the related sets and intersections. The element is selected when its circle or semi-circle is clicked. The selection stays persistent when selecting another timestep or view (*static, aggregate, and diff*). With this feature, one can trace an element across different timesteps when flipping through *diff* views. Left and right arrow keys enable switching to previous or next timesteps respectively.

4.3 APPLICATION EXAMPLES

To demonstrate the applicability and effectiveness of our approach, we study two realistic application examples. The prototype containing these examples is in the supplementary material [113] and hosted online[1].

4.3.1 *Researchers' Field of Interest*

Publication venues (conferences, journals, etc.) can be mapped to fields of science, fields can be modeled as sets, and researchers are elements. Publication by a researcher in a field of science determines the set membership. The publication year adds the temporal component. We collected publication data from conferences of 6 research fields (sets). We filtered those researchers who published at least 30 articles over all fields and timesteps, obtaining 498 researchers. Since researchers can publish in multiple fields, they can appear in multiple sets. Hence, the six sets overlap, with 32 different set intersections. The dataset covers publications from 1990 to 2019, divided into ten timesteps of three years each.

Overview of temporal changes. The aggregated view in the middle (Figure 4.3a) shows that every set is present in all the timesteps (black left edge of rectangles in the bottom layer). Vertical colored bars in the timeline (Figure 4.3b) and the line chart below the nodes in L1 (Figure 4.3a) show that the number of researchers in ***AI/ML*** has grown rapidly compared to other fields. While panning the aggregated

[1] (Accessed May 2023) `https://s-agarwl.github.io/dynamicsets`

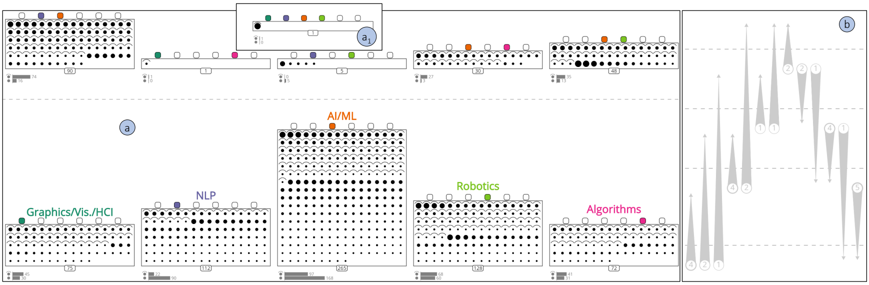


Figure 4.5: Cutouts from the prototype interface showing (a) set intersection view of the timestep 2017-2019 from the computer science research dataset (Section 4.3.1) and (b) summary edges in the diff view between 2016 and 2017 timesteps from the Linux GitHub repository dataset (Section 4.3.2).

view, we observe that there are only four layers in the dataset. It indicates that no researcher has published in more than four research fields at any timestep.

Most active researchers. To find the researchers with the highest number of publications and their field of research, we sort the element list by decreasing order of the '*Sum*' column (left-click on the column header), as shown in Figure 4.3d. To know the researcher's field of study, we look at the colored boxes in each row. The colored boxes (orange and green) for the first researcher *Wolfram Burgard* () show that he has published only in ***AI/ML*** and ***Robotics***. A pattern can be seen from the first five rows in the list. In each row, the green-colored box is very prominent (less transparency), which indicates that the top five researchers have consistently published in ***Robotics***. The second row shows researcher *Gerd Hirzinger* () with only a green colored box, indicating his specialization in ***Robotics***. These observations are confirmed by their evolution charts (Figure 4.3b_1). As seen from the timelines of the first five rows, the researchers are active except *Gerd Hirzinger*, who stopped publishing in the last two timesteps. Additionally, the number of articles they publish per year has been declining except for *Mayasuki Inaba* (). For further exploration, the publication details of a researcher are available by right-clicking the corresponding element.

Varying contributions in research fields. Investigating the timestep 2017–2019, we observe that there is a high overlap among sets (Figure 4.5a). The set ***AI/ML*** is the largest (height of the rectangle), while other intersections contain fewer elements. Comparing two different set intersections: $X_A = \{$ ***Graphics/Vis./HCI***, ***Algorithms*** $\}$ vs. $X_B = \{$***Graphics/Vis./HCI***, ***NLP***, ***AI/ML***, ***Robotics***$\}$ (Figure 4.5a_1), we find that both contain only one element. Through different sizes of circles, the approach can help spot the differences in the membership weights of elements. For instance, X_A has one element *David R. Karger* (small circle), who published three papers, while X_B has a researcher *Sergey Levine* (big circle), who published 48 papers.

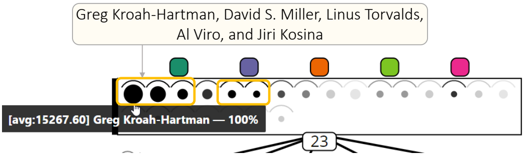

Greg Kroah-Hartman, David S. Miller, Linus Torvalds, Al Viro, and Jiri Kosina
[avg:15267.60] Greg Kroah-Hartman — 100%
23

Figure 4.6: Consistent contributors in five modules (identified by colors) of the Linux GitHub repository across all timesteps.

Higher individual publications in *Robotics*. Analyzing the details of specific elements, we investigate the researchers who have published in both *NLP* and *Robotics* during 2017–2019. In Figure 4.5a, the rectangular node representing the intersection of the two sets contains five circles. None has a hat marker, indicating that no researcher published in just these two fields. For the base sets *NLP* and *Robotics*, in Figure 4.5a, we can see that the heights of corresponding rectangles are similar, suggesting that their cardinalities are almost the same (112 and 128, respectively). Our approach enables further analysis based on the circle size and their distribution. Through the default sorting by the membership weight, we see that the size of circles in *Robotics* is larger than in *NLP* (both exclusive and non-exclusive elements). Hence, we can say that the number of publications by most individual researchers in *Robotics* is higher than researchers in *NLP*. On hovering, we find that the maximum number of publications in *Robotics* is 34 by *Masayuki Inaba*, whereas in *NLP* it is 25 by *Graham Neubig*.

4.3.2 *Evolution of Developer Activities in Software Projects*

In this example, we analyze changes in software development activities. Analyses like these can show staff churn, productivity differences, and modules requiring more work, thus helping manage a software project [115]. We study 5 Linux modules from its GitHub repository. The modules (*fs*, *drivers*, *arch*, *net*, and *kernel*) are the sets, and the elements are the committers (developers). The membership weight is the number of commits done by a developer to a module. We divide the repository evolution from 2008 to 2017 into ten yearly timesteps and filter the developers who made at least 100 commits to these modules, obtaining 111 committers.

Consistent commits in all modules. We select the aggregate view by clicking the '*Aggregate all*' button. Since presence across all timesteps is encoded via opacity, to find the most consistent developers, we look for black circles. Hovering over them reveals the developer names, average commits in every timestep, and percentage of presence in all timesteps, as shown in Figure 4.6. These developers contributed to all modules (the rectangle is in layer L_5) in every timestep. The circle sizes

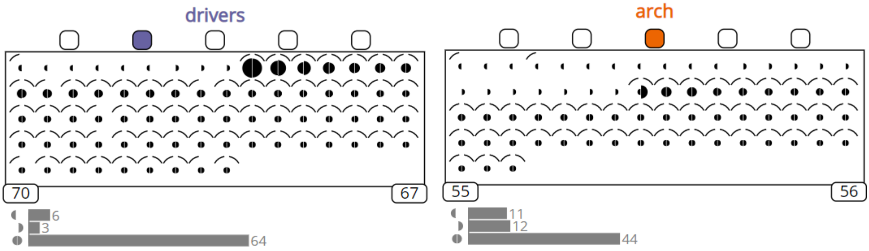


Figure 4.7: Comparing stability of developer contributions among modules *drivers* and *arch* in the *diff* view between 2016 and 2017.

show a large difference in the average number of commits, from 15267.60 for *Greg Kroah-Hartman* (Figure 4.6) to 834.20 for *Jiri Kosina*.

Module stability. Investigating changes in developer activities in the last two timesteps (2016 and 2017), we select the *diff* view between 2016 and 2017. In the bottom layer, we focus on the two biggest rectangles representing the *drivers* and *arch* modules (Figure 4.7). The *arch* module has the least change in terms of cardinality (55 to 56). A change in cardinality alone is not a good indicator of stability. On a closer look, we see that the module has many semi-circles. The horizontal bars beneath the rectangle (*arch*) show that 11 previous committers did not contribute in the later timestep. The cardinality remained stable because 12 new developers contributed to the module. In contrast, we see that the *drivers* module has the most significant number of developers (64) who contributed in both timesteps. Hence, across the two timesteps, the *drivers* module was the most stable in terms of developer contributions.

Developers shifting their focus among modules. Still focused on analyzing changes in developer contributions in the last two timesteps, we look at the corresponding *diff* view. Zooming on the summary edges in the *diff* view, we see many inter-layer tapered edges going up and down (Figure 4.5b). Upward edges indicate that developers contributed to more modules than before, and vice-versa for downward edges. The summary edge from L_2 to L_1 indicates that five developers narrowed their focus to only one module. Selecting the edge populates the element list with their names.

Different patterns of contributions. To highlight the different patterns of contributions among developers, we use evolution charts (Figure 4.8). We observe stable and consistent contribution patterns to two modules by *Felipe* and to all five modules by *Greg*. We also see an inconsistent contribution across timesteps (*Paul*). Additionally, there is a developer who did not contribute to any module for some years (*Bartlomiej*).

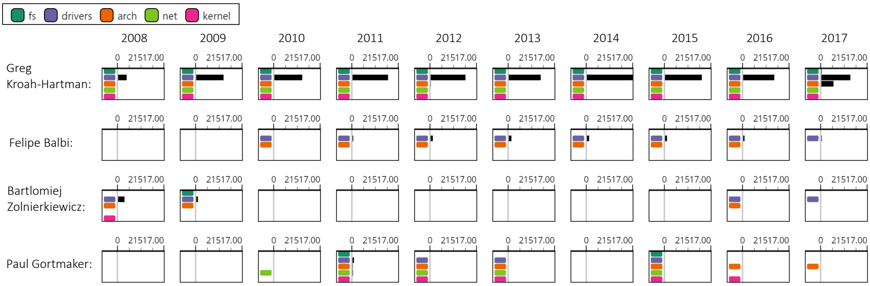


Figure 4.8: Evolution charts of four committers showing different contribution patterns across five Linux modules (encoded in color).

4.4 DISCUSSION AND FUTURE WORK

Unlike existing set visualization techniques, our approach can model and visualize the membership weight of individual elements for each set they belong to, together with the dynamic associations between elements and overlapping sets over time (RO 1). The proposed technique allows in-depth visual analysis of how sets and their overlaps grow and shrink and how elements 'migrate' through sets. In the context of understanding group behavior, the approach provides a temporal overview of individual entities in groups (RO 1.1), e.g., stable and consistent contributions of experts in research fields or modules of a code repository. The design enables analyzing exact changes in group memberships between two timesteps (RO 1.2) while preserving the details of membership for each entity (RO 1.3). The insights from two real datasets highlight the feasibility of such an analysis. In this section, we reflect on the used visual encodings for dynamic set memberships, discuss the possibility of extending existing static set visualization techniques and scalability of the proposed approach.

4.4.1 *Encodings for Dynamic Set Memberships*

Depending on the scenario, some set intersections are more important than others. Usually, intersections involving many sets are the most important but contain few elements. We used a layered layout to provide context and aid the analysis by specifically differentiating between the set intersections based on the number of involved sets. Although the lattice structure in the layered layout takes some time for an analyst to understand, it has been found useful by users, as demonstrated in a user study [116]. However, the lattice diagrams in the user study were static with minimal interactions. Hence, the usability of the proposed system cannot be based on it. The future work could include performing further studies on the understandability of lattice diagrams in the context of analyzing overlapping set data.

The proposed approach embeds the set membership details of each element as the size of circles inside rectangular nodes. The ordering of elements within a node and using opacity to encode the temporal information helps to see an overview of the weighted set membership across all timesteps. The chosen shapes (for elements and sets) worked well with tapered edges to help analyze the exact changes between two timesteps. However, since the temporal changes are at different granularity levels (e.g., membership weight of each element, overall changes in a set intersection, number of gained or lost memberships in sets), remembering the used visual encodings becomes difficult. The proposed filters and interactions help to ease the analysis, but the system still involves a high cognitive load. Since the used encodings are complex and not easy to remember, there might be scope for simpler visual encodings in alternate designs, which can be explored by future research in this direction.

4.4.2 *Extending Existing Set Visualizations*

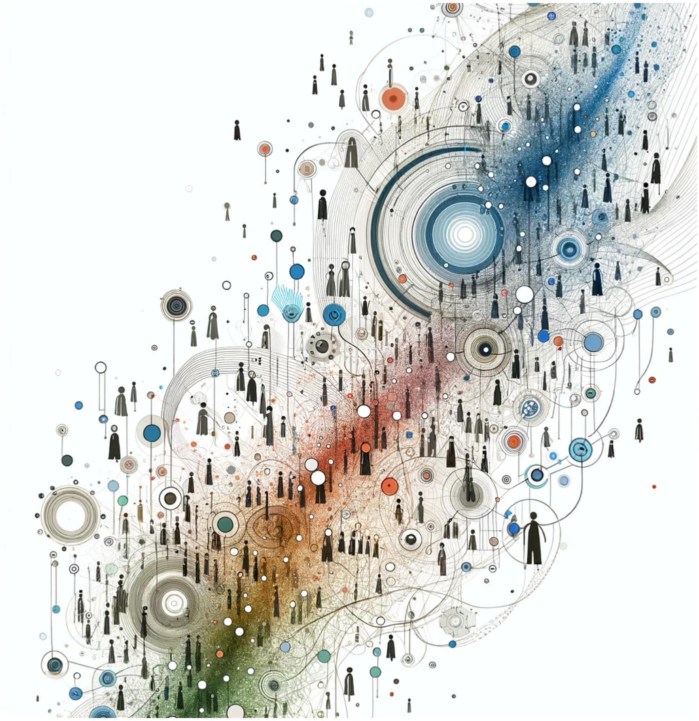
Most of the research in set visualization has focused on analyzing a single timestep (see a survey [19]). Theoretically, any static set visualization can be extended to show temporal changes in the set membership of elements. The extension can include linked views or usage of animation. However, since many techniques do not show individual elements (e.g., [24, 114, 117, 118]), they will be unable to show the temporal change in the element-set membership weights. Euler and Venn diagram-based variants have limited scalability, whereas in matrix [119] and node-link-based techniques [120, 121], the set intersections are not represented explicitly. Aggregation-based techniques represent each set intersection and are highly scalable, but they do not show individual set elements [24, 114, 117, 118]. As a result, these techniques are unable to show the details of set elements, such as element-set membership weight. Interactions and additional linked views can overcome some drawbacks in these techniques, but their design and integration are not straightforward.

4.4.3 *Scalability*

Scalability for dynamic set visualization includes the number of sets and relevant set intersections, the number of elements, the level of detail shown for set membership, and the number of timesteps that are represented. Like any approach that explicitly models set overlaps, the number of relevant intersections can explode. In the worst case, the set intersection graph has 2^n nodes. But in practice, often, a large majority is omitted as these correspond to empty set intersections. We observed in the tested data that up to six significantly overlapping sets could be represented without the visualization becoming too dense. Region-based or line-based overlay techniques [19, Section 4.2] does not scale any better if sets significantly overlap with each other. Other approaches that show individual elements (e.g., Bubble Sets [47], OnSet [122]) scale similarly to our technique with respect to the number of elements. As demonstrated, our approach is scalable up to ten timesteps. Since the changes in element memberships might not be interesting for all timesteps or for all set intersections, future work would include using data analysis techniques to highlight the most relevant timesteps and intersections.

Part II

EVOLVING ENTITY INTERACTIONS



5 ANALYZING USER BEHAVIORS FROM MIXED REALITY SESSIONS

Analyzing the actions and interactions of entities in a dynamic environment helps to understand their behaviors. This includes understanding their individual decisions at a fine-grained level or their overall strategy of interacting with others to accomplish a task. To gain such types of insights, visual exploratory analysis is crucial. To build the relevant visualizations, we first systematically derive a design space, mapping the relevant user behavior attributes to the appropriate visual encodings (RO 2.1). To achieve this goal, we focus on a specific type of dynamic scenario: virtual and mixed reality sessions.

We are witnessing an increasing trend of including multiple users and physical objects in virtual and mixed reality environments. Some examples of such scenarios include multi-player games, virtual spaces for customers and business meetings, and collaborative training for emergency situations [53]. In such scenarios, there are multiple users and objects, where both can be either virtual or real. To blend reality and virtuality in a 3D virtual environment, avatars are usually used to represent users, while virtual objects are mapped to real objects. Understanding the behavior of users in these dynamic scenarios is challenging because it requires analysis of both spatial and temporal attributes from the data of recorded sessions. To understand the behavior on a fine-grained level, it becomes necessary to answer questions, such as, *what actions were performed by the entities? which users interacted with other entities? when?* and *how did the behavior of a user affect the others?* On the basis of this analysis, developers of virtual and mixed reality applications and researchers in human-computer interaction can draw conclusions on, e.g., user behavior or performance. The need to understand multiple attributes and streams of actions in these scenarios, together with the blending of virtual and real spaces, makes the challenges unique to the virtual and mixed reality environments.

Although not explicitly targeted at virtual or mixed reality environments, a few visualizations (e.g., [123, 124]) have been proposed to gain insights into the navigation behavior in virtual game worlds. Similarly, the visualizations can be helpful in understanding the behavior of users and their interactions from the recorded data of virtual and mixed reality sessions. However, a holistic analysis approach to investigate multiple aspects of the data from such sessions is still missing. This chapter aims to present a systematically derived design space for ex-situ visualizations analyzing user behaviors and interactions from the recorded session data of virtual and mixed reality sessions. To derive the design space, the existing visualization approaches analyzing traditional user interactions, eye movements, physical motion, and stories are sampled in Section 5.1. The derived categories in the design space are explained in Section 5.2, along with their combinations for usage in two possible scenarios. Using the design space, insights from the analysis of a concrete remote collaboration mixed reality scenario are discussed in Section 5.3. Finally, future research challenges based on the design space and insights from the application example are discussed in Section 5.4.

5.1 RELATED AREAS AND VISUALIZATIONS

Recently, a few approaches have been proposed to visualize the user sessions of virtual and mixed reality environments. For instance, Nebeling et al. [53] proposed a toolkit to visualize the recorded data of mixed reality sessions. However, they do not discuss alternate ways of encoding specific aspects of the data. Gruenefeld et al. [125] propose a rapid prototyping approach that includes a state-based replay of the recorded user sessions. Kloiber et al. [126] show in-situ trajectories as visualizations to understand the movement of objects, helping to reflect on the user performance in collaborative tasks inside a virtual reality environment. Since these works are recent and rare in visualizing mixed reality session data (to the best of our knowledge), we do not discuss it in a separate category and instead reference it across the paper, where appropriate.

To derive the design space, we first look at the visualization examples from the related fields: analyzing interactions, eye tracking, physical motion, and stories. These four fields analyze the same aspects of the data as we intend to visualize recorded sessions of virtual and mixed reality scenarios or are relevant for the purpose (e.g., visual summary of the sessions). Although there is a vast number of approaches for visual analysis of spatio-temporal data (e.g., gesture recognition, movement analysis, event classification, and event sequence mining), the focus is on including techniques relevant to the design space capable of showing extracted events, actions, and interactions from the user session data. Hence, instead of following a qualitative sampling approach, a diverse set of examples are selected with the goal of covering a broad range of approaches.

5.1.1 *Interactions*

There are a few visualizations for analyzing user interactions by extracting the relevant information from the recorded session data. Blascheck et al. combine the data generated in a user study through recordings of the user's interactions with a visual analytics interface and a think-aloud protocol [54]. Interactions and thinking-aloud actions are treated as events. A timeline-like visualization showed the event sequences along with the respective regions of the interface for every participant. The design helped them to compare the behavior of different participants. Regarding the interactions of software developers with the integrated development environment (IDE), timeline visualizations have been proposed, marking relevant interactions as events [55, 56, 57]. The vertical axis encodes different source code files and dialog boxes of the IDE, while time is depicted on the horizontal axis. A similar visual design has been used to show interactions between people from recorded videos of business meetings in a physical room [60].

5.1.2 *Eye Tracking*

Analyzing the gaze location of a user conveys context (e.g., during a collaborative activity) and helps in understanding the interaction with other users or the environment. Modeling the gaze as an event, eye tracking studies record eye movements of a user watching or interacting with a static stimulus (picture) or dynamic stimulus (video or interactive interface). The semantically meaningful regions in the stimulus are called areas of interest (AOI). While investigating the individual gaze events in detail is important to understand the user behavior, it is also necessary to get a sequential overview of how users shift their gaze from one area of interest to another. Space-time cubes are often used to visualize the shifting gaze behavior by showing time in the z-axis and position in the x- and y-axes [127, 128]. Other techniques use 3D scanpaths, attention maps, and linked view visualizations, with the stimulus being an immersive video [129] or a virtual 3D scene [130]. Visual analytics approaches support the comparison of different users by representing their gaze behavior while abstracting the real stimulus [54]. Blascheck et al. survey further visualization approaches [131]. For virtual and mixed reality user sessions, eye movement data can also be recorded [129], but the topic is relevant only if it is applied to combinations of an interactive stimulus and human body movement.

5.1.3 *Physical Motion*

Apart from eye movements, there exist other visualizations of physical motion, such as (a) individual trajectories [74, 75, 76], (b) segments of the trajectories to explore local movement patterns [77, 78, 79], (c) aggregations of multiple movement trajectories [80, 81, 82], and (d) the environment along with the movement to preserve its context [83, 84]. These visualizations use different techniques, such as static and animated maps, interactive space-time cubes, time lenses for trajectories in small segments, or color for density fields. A survey report describes them in detail [73]. Visualizations focusing particularly on motion capture data are used to show clusters of human poses, encode them with a gradient color scale, and then spatially position them in the order of their occurrence [132]. Some visualizations already show the movement of entities from sessions captured in virtual reality while abstracting details of the environment [123, 82, 124].

5.1.4 *Stories*

Storyline visualization is an intuitive approach to provide a visual summary of an event sequence involving different entities. An early implementation used a storyline to summarize plots of movies [133] where each character of the movie is shown as a separate horizontal line and the x-axis represents time. The lines can bend and be grouped when the respective characters are in the same film location (i.e., when they interact in a movie frame). Various layout algorithms have been developed to enhance their effectiveness by reducing overlap, improving aesthetics, and supporting on-the-fly layout computation for streaming data [134, 135, 136]. They enriched adaptations of the approach by encoding the related events through icons and background colors. The idea has been adopted in different fields; one such example is visualizing software evolution through storylines [41]. A time curves visualization shows the similarity of events through spatial proximity while preserving their temporal sequence [137]. These visualizations can be used to show data from virtual and mixed reality environments to convey a coherent and comprehensive overview of the user sessions.

5.2 DESIGN AND APPLICATION SPACE

Card and Mackinlay [138] state that the purpose of a *visualization design space* is *"to understand the differences among designs and to suggest new possibilities."* In the visualization literature, a variety of general visualization design spaces and taxonomies have been discussed [139, 138, 140]. In these theoretical frameworks, data models, visualization categories, and tasks often form the key elements. To study visualization options on a fine-grained level, some works tailor such design spaces to specific types of data and visualization (e.g., dynamic graphs [34, 141], composite visualizations [142], word-sized graphics [143, 144]) or applications (e.g., eye tracking visualization [131], software visualization [145], or games visualization [146]). But despite the variety of such existing frameworks, we are not aware of any work targeting such a tailored visualization design space for user sessions in virtual and mixed reality environments.

To structure the design space, we first introduce the data that is recorded and analyzed, then provide a categorization of visualizations. Since we also discuss application scenarios, we call the suggested framework *design and application space*. Unlike most other related frameworks, which structure the visualizations based on examples from within the respective domain, we have to work with the examples from the related domains discussed above because there is not yet sufficient coverage within the domain (i.e., the visualization of virtual and mixed reality user sessions).

5.2.1 *Data*

To experience virtual and mixed-reality environments, humans usually wear a head-mounted display. The positions of the user's head and controllers are tracked by the sensors to synchronize the actions, movement, and orientation in the real world with the entities in a virtual scene. Moreover, mixed reality environments may also increase the complexity by adding entities in the real world (for tangible feedback) mapped to a virtual entity.

Entities in mixed reality sessions can be active, such as users, virtual avatars, and physical drones. Additionally, the entities can also be passive objects with which active entities interact, e.g., controllers, virtual and real objects. We model these entities as vertices $v \in V$. Tracking an entity includes recording its position in three-dimensional space and orientation defined by three angles. Hence, at any given time, each entity can be described with a function $p : V \times T \to \mathbb{R}^3$ for positions (or including orientation: $p : V \times T \to \mathbb{R}^6$). A characteristic feature of mixed reality sessions is that entity positions are described in two worlds. Hence, they can be modeled as $p_{\mathcal{R}}(v, t)$ for the real world and $p_{\mathcal{V}}(v, t)$ for the virtual world. Physical entities in the real world are traced by mounting optical trackers on the body (e.g., OptiTrack) or using image-based sensors (e.g., Kinect). Depending on the scenario, tracking can be fine-grained, e.g., finger positions or palm orientation, which are tracked and stored through sensors such as Leap Motion.

Entities may interact with each other and trigger events actively or passively. We model them as events $e \in E$. Since events occur at a certain time and last for a certain duration, we model time as a function $t : E \to T$ (or $t : E \to T^2$ for time spans). Events and interactions involving multiple entities can be mapped as $V : E \to 2^V$. Thus, from the entity–event relationships, we can derive the set of events $E(v, t)$ that involve an entity $v \in V$ at point $t \in T$, as well as the involved entities $V(v, t)$ in an event $e \in E$ at point $t \in T$. The events $e \in E$ can further be discerned by whether they are (global events) or have a location of occurrence in reality or virtuality (local events). Local events, like objects, carry positions $p : E \times T \to \mathbb{R}^3$ in reality ($p_{\mathcal{R}}$) or virtuality ($p_{\mathcal{V}}$). Events in virtuality, such as actions of active virtual avatars or collisions of passive objects, can be easily recorded as log files. For mixed reality objects, different data streams must be merged to detect the respective events. Events can be triggered in the real world through input and sensing devices, such as controllers. Additionally, more sophisticated types of motion (e.g., gestures) can also be extracted from the recorded position data of users.

The data recorded for virtual and mixed reality sessions may also involve a holistic recording of the scene. A scene $s \in S$ can either be a two-dimensional image as recorded by a camera or a three-dimensional capture of the scene, which can be interactively explored. It is possible to map each timestep to an image of the real or virtual scene $s : T \to S$.

Table 5.1: Classification of existing visualizations into seven categories for analyzing user behavior from mixed reality sessions.

	Interactions	Eye tracking	Physical motion	Stories
(A) Entity identifiers	[55, 56]	[54, 128, 127, 129]	[132, 60, 74, 75, 78, 80, 84]	[136, 134, 135, 147]
(B) Event identifiers	[55, 56, 124]	[54, 128, 127, 129, 130]	[73, 60, 77, 81, 84]	[136, 135, 137, 147]
(C) Entity timeline	[55, 56]	[54, 129]	[60]	[136, 134, 135, 147]
(D) Event timeline	[55, 56]	[127, 128]	[132, 76, 81]	[137]
(E) Event density fields	[135, 124]	[127, 128, 129, 130]	[132, 75, 78, 81, 80, 124]	[137, 135]
(F) Trajectory view	[54, 57]	[148, 127, 128]	[123, 74, 76, 75, 77, 78, 81, 82, 124]	
(G) Scene view	[133, 135]	[127, 128, 129, 130, 54]	[132, 60, 74, 76]	[133, 147]

5.2.2 *Visualization Categories*

To classify the related visualizations, we systematically explored them and assigned keywords. The keywords reflected concepts (e.g., *time*, *event icon*, *summary*) that are useful for visual analysis of data recorded from mixed reality user sessions. A keyword was assigned to a visualization if it represented the same reflected concept. Based on the similarity of data property, we grouped the keywords. As a result of the grouping, we generated seven categories. Table 5.1 shows these categories along with references to the related publications. We describe each category in the following paragraphs, applying them to the study of user behavior in virtual and mixed reality.

(A) Entity Identifiers. An entity $v \in V$ can be either a user or an object in a virtual or mixed reality environment. Entity identifiers are used to identify each user/object present in the environment uniquely. Different visual encodings such as text, icons, colors [132, 129, 75, 78, 80, 84, 53], and position [54, 136, 134, 135, 147] can be used to represent them. Moreover, the similarity between entities can be shown by a dendrogram [132, 54, 128]. These identifiers are often used in combination with visualizations from other categories, such as *entity timeline*, *trajectory view*, and *scene view*.

(B) Event Identifiers. These identifiers are used to uniquely identify each event $e \in E$ that occurred in the session (or type of event, respectively). Different visual encodings such as text [147], icons [129], shapes [135, 84], colors [54, 137, 55, 56, 130, 60, 77, 81, 124, 53], and position [127, 77] are used to represent events. Their usage is most often in combination with visualizations from other categories, such as *entity timelines* [54, 55], *event timelines* [132], and *trajectory views* [77, 84].

(C) Entity Timeline. An entity represented by $v \in V$ has features that change over time, for instance, associated events $E(v, t)$, interactions with other entities $V(v, t)$, or other attributes. The visualizations of entities in this category show a temporal sequence of these features. Besides the timeline, the dominating visual structure of the visualization is a set of entities $V' \subset V$, for instance, encoded in lines or as rows of the linear timeline. It is common to represent time on the horizontal axis [55, 56, 54, 129, 60, 134, 135, 136, 147]. Entity timelines can be drawn for individual [127, 128, 76] or multiple entities [135, 54, 55, 56, 147].

(D) Event Timeline. Although they also show a timeline, visualizations in this category focus on representing a set of events $E' \subset E$ and their temporal sequence of occurrence as primary visual glyphs. They are often discerned by their event type, which provides a structure for the timeline. An event timeline can be represented in linear [128, 127, 55, 56, 130, 53] and non-linear [132, 137, 81] layouts. It is common to encode the time span of events by the size/area of the glyph [135, 127, 128, 130, 55, 56], and also by the relative distance between event identifiers [137].

(E) Event Density Fields. Groups of local events $E' \subset E$ are associated with positions $p(e, t)$ $(e \in E')$ and other attributes such as involved entities $V(e, t)$ $(e \in E')$. An event density field shows information of event sets E' aggregated across time $t \in T$ through histograms [54, 78, 81], heatmaps in a 2D spatial context [128, 127, 130, 129, 124], size/area of glyphs [135, 80], or 3D surfaces [75]. These visualizations can be augmented with context to highlight additional attributes, for instance, representing event density on the map juxtaposed with another view showing linked static entities [124, Sect. 4.1]. A cluster of closely placed event glyphs also represents the density of events in a timeline [137, 132, 53]. Different patterns of clusters [137] can be used to compare event timelines of multiple mixed reality user sessions.

(F) Trajectory View. This category includes visualizations that show movement $p(v, t)$ of the entities $v \in V$ across time $t \in T$. The movement is usually shown by projecting position on two dimensions [53] and representing time through either a gradient color scale [123, 73, 75, 81] or a third dimension [127, 128, 148, 76]. The direction of movement is also shown by glyphs [74, 77, 78, 53]. Details of the position can be abstracted by projecting it on the y-axis while showing time on the x-axis [54, 57]. The trajectory can be enriched by visualizing additional attributes of entities, for instance, showing trajectories of entities with different colors to represent different types of objects carried by a player in a virtual game [124, Sect. 3].

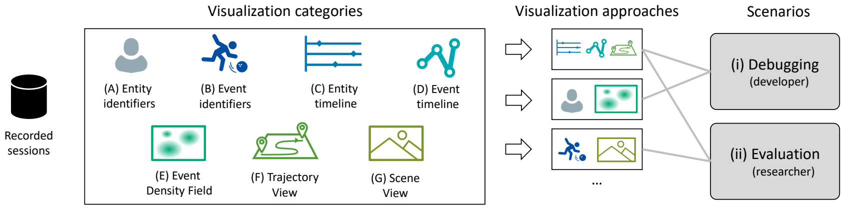


Figure 5.1: Design and application space for visualizations of recorded virtual or mixed reality sessions; the seven categories of visual encodings (A–G) provide the building blocks of specific visualization approaches, which can be used in two scenarios: (i) debugging the environment and (ii) analyzing data from user studies in a research context.

(G) Scene View. This category includes visualizations that show scenes $s(t)$ along time $t \in T$. The environment details can be abstracted in these visualizations [123, 132, 133, 135, 147] where the level of abstraction depends on the data analysis task. Techniques used in these visualizations include using multiple images (as keyframes like in a comic strip) [127, 128] and video/animation [127, 128, 129, 132, 76]. Scenes from both real [60, 54, 76, 53] and virtual worlds [129, 130] can be included to provide a complete overview of the user session.

The above categories of visualizations can occur independently. However, they are often mixed with one another (see references that occur multiple times in Table 5.1). Multiple views that are synchronized by brushing-and-linking interactions provide a simple solution for this. But it is also possible to combine several of these categories within an integrated representation. For instance, a three-dimensional *space-time cube* of entities combines an *entity timeline* (; one axis) with a *trajectory view* (; two remaining axes) [76, 127, 128]. Figure 5.1 illustrates this combination as a selection of visualization categories that are connected to a specific application approach.

5.2.3 *Application Scenarios*

Once we have the building blocks from the design space of visualizations, we can apply them in different combinations for different scenarios. Instead of discussing visual analysis tasks for a specific virtual or mixed reality application, we focus on two general scenarios. However, the categories in the design space can be combined and used for a custom scenario based on specific requirements.

(i) Debugging. Developers of virtual/mixed reality applications often face the challenge of identifying errors and judging the effectiveness of the solution patches. A common way to approach this is by executing the program, replicating events in the environment, and then looking at a real-time rendering of the scene. For example, it is cumbersome to fine-tune the coordinate systems of different sensors

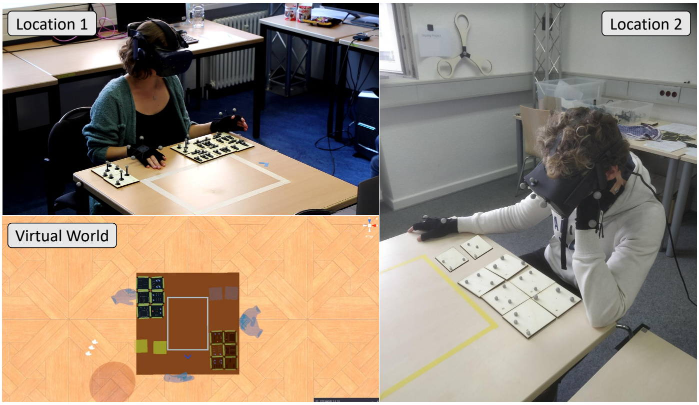


Figure 5.2: Remote collaboration application example – Two participants in different locations collaborate in virtual reality; together they figure out a certain arrangement of the components in the virtual world to solve a puzzle.

in the scene and synchronize them. Recent works by Ashtari et al. [149] and Speicher et al. [150] systematically derived and reported on such challenges in the development of such applications. They include (a) facing too many unknowns in development, testing, and debugging [149] and (b) the need to work with multiple types of devices [150, Section 5]. The challenges are usually addressed by analyzing the log files, system messages, and source code to identify errors. Visualizing the session data, which consists of multiple streams from different devices, can be helpful in addressing these challenges. Visualizations have been found to be useful for supporting developers in debugging and designing several aspects of virtual reality environments [147, Sect. 4]. However, the visualizations do not include spatial information, and they are limited to specific environments. Hence, the visualizations that show multiple aspects of a mixed reality environment are important for supporting developers in debugging and designing virtual and mixed reality applications.

(ii) Evaluation. A challenge HCI researchers face while evaluating user studies is understanding the complex movement and behavior patterns of multiple users interacting in mixed reality. The user data has multiple degrees of freedom, and it is challenging to map the data of multiple users or other entities in such a way that patterns (e.g., two entities being in the same position at the same time) become visible. Without any alternate representation of the recorded data, it becomes difficult to verify and evaluate the data itself. Ashtari et al. [149] also highlighted evaluation challenges in understanding details of specific situations (e.g., a stimulus that distracted the user). Additionally, researchers need to analyze data from multiple sessions to evaluate the design of the proposed novel features for the environment. Visualizations can help in addressing these challenges. Hence, they are important for supporting an initial analysis of user study data and for conducting qualitative studies.

5.3 Application Example: Remote Collaboration

In the application example, two participants collaborate in one shared virtual environment. They interact with each other and various objects, although they are not co-located in the same physical space. In the recorded scenario, the participants sit in different rooms (cf., Figure 5.2, Location 1 & Location 2). However, in the virtual environment, they appear to be sitting at one table facing each other (cf., Figure 5.2, Virtual World), allowing worldwide immersive collaboration. Voice is recorded and streamed to the respective other location so that the participants can hear each other. Both participants have tiles on their desks. The positions and orientations of the tiles are tracked optically and synchronized with the virtual environment. Hence, every user can see a virtual representation of the other user's tiles. The objective of the collaboration scenario is to arrange all the tiles according to a plan each collaborator has only a part of. The participants must collaborate to complete the puzzle. With this application, we target a research scenario where a visualization should support the qualitative *evaluation* of user sessions. The tool[1] is shown in Figure 5.3 and is available in the supplemental material [151].

5.3.1 *Visualization Design*

Virtual reality designers and researchers needed to compare users based on their actions, how they communicated with each other, and their interactions with environmental objects. To fulfill these requirements, we incorporate an *entity timeline* (☰) as shown in Figure 5.3a. We use colored glyphs to identify different types of events (*event identifier*, 🏃) shown as a legend in Figure 5.3b. Since several entities can be involved in an event, the event timeline should represent connections between involved entities as well. We took inspiration from PAOHvis [152], and extended the design to show events together with entities. We chose a matrix layout for the entity timeline where the horizontal axis represents time (from left to right), and the vertical axis lists entities in individual rows (Figure 5.3c). The length of each scene corresponds to the width of the visualization and is annotated below. To visually represent the density of events, we integrate a histogram that shows *event density fields* (▦), where the size of each bin is set by default to six seconds (Figure 5.3d). Verbal communication can also be considered as an event; we show the density of their conversation by a waveform visualization (Figure 5.3e). We integrate a *scene view* (🖼) component that plays the recording of a selected virtual scene (Figure 5.3f). A red vertical line across all plots represents the current position of the playback.

[1] Hosted at (Accessed May 2023): https://s-agarwl.github.io/mrsessions

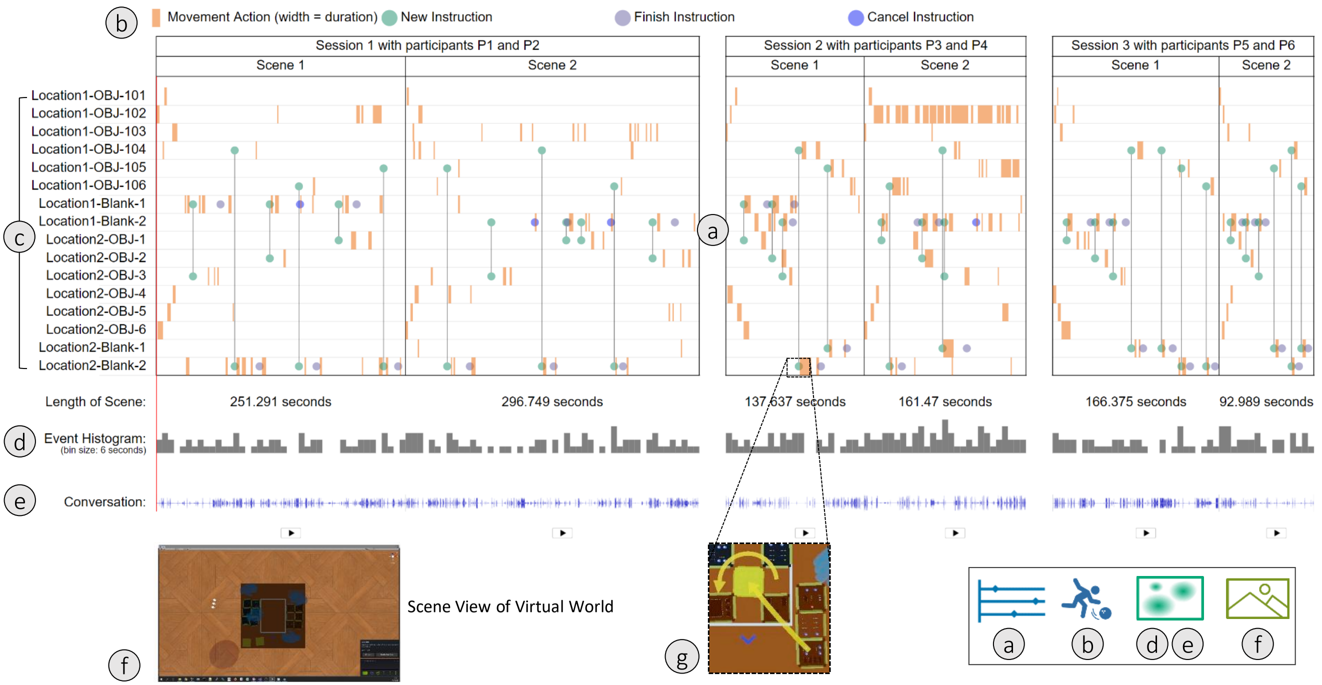


Figure 5.3: Remote collaboration application example – (a) A timeline visualization showing sessions of a mixed reality environment. (b) Colored glyphs are used to identify events, while (c) entities (users and objects) are shown in separate rows. A vertical line between two rows denotes interaction (touch) between the corresponding entities. The density of events and recorded conversations between participants are visualized through (d) histogram and (e) waveform, respectively. (f) Playback of videos recorded from the virtual environment for each scene.

5.3.2 *Insights*

To evaluate the collaborative game, the visualization helps in analyzing the strategies of the players and hints at the obstacles they face. We illustrate this with the six scenes of three pairs of players shown in Figure 5.3.

Comparing Interaction Strategies. In Session 3 – Scene 2, the width of the column (participants P5 and P6) has the shortest duration compared to all other scenes. Also, fewer *Movement Actions* related to objects indicate the high effectiveness of the players. Further, a strategic pattern can be observed: First, the collaborator in Location 1 starts to interact three times with three different objects in Location 2 (three consecutive green dots linked to the object *Location1-Blank-2*). Afterwards, the collaborator in Location 2 does the same but with *Location2-Blank-1* and *Location2-Blank-2*. Both players communicate little (cf., Figure 5.3e), indicating that verbal interaction was not as needed as in other scenes. In contrast, in Session 1 – Scene 2, which is the longest, we can derive a less efficient pattern. The first interaction starts after some verbal exchange. Then, the collaborators begin slowly using the objects to interact with the objects of the other location. In the middle, we can observe a pair of consecutive dots in the same two rows, indicating that there was a mistake in the interaction.

Exploring Collaboration Details. To further investigate this specific part of the scene, we can listen to the audio and watch playback using the scene view to gain insight into what went wrong. The last interaction between *Location1-Blank-2* and *Location2-OBJ-2* is interesting, as its finish event is not very close to its start and further, *Location2-OBJ-2* is moved again. At the same time, verbal exchange increases, indicating that there was a discussion. In Session 2 – Scene 2, three objects at both locations are moved at the beginning of each scene (the three consecutive *Movement actions*). This indicates that each player can fulfill certain actions without collaborating. This seems to be a common pattern: The inter-location interaction in all scenes starts after these three actions.

5.4 DISCUSSION AND FUTURE CHALLENGES

The application example shows that users adopt different collaborative strategies to solve a task. Using the visualization, we were able to discover some of the strategies and identify similar behavioral patterns. We considered entity interactions and temporal aspects of the data. For other realistic applications, the spatial aspect could play an important role as well and can be visualized using a *trajectory view*.

The proposed design space to analyze user sessions of virtual and mixed reality environments is only a first step towards understanding user behavior and their interactions (RO 2.1). With the ease of recording the data from these sessions, advances in virtual and mixed reality hardware, and the rise of dedicated applications, the need and value of such analysis will become even more apparent. To further facilitate such analysis in the future, further research is required to address the following challenges in effectively visualizing such user sessions.

5.4.1 *Dual Representations*

Studies have shown the value of representing virtual objects in reality by using drones as haptic devices (e.g., [153, 154]). This dual representation of some entities is a unique feature of mixed reality applications. To observe the user behavior in such scenarios, especially in user studies by researchers, it might suffice to fuse the two representations in the visual analysis. However, there are other use cases where studying the divergences and occasional misalignment of the two representations is necessary to ensure a smooth, immersive experience. For instance, developers would need to calibrate the hardware for synchronized representations of the same entity. Although there exist techniques and guidelines to help visually compare two entities in general (e.g., [100, 155]), they do not assume nor exploit the characteristic feature of dual representations.

5.4.2 *Diverse Data and Dynamics*

User sessions in virtual and mixed reality applications generate a rich and complex dataset. It involves diverse data streams such as trajectories, events, video, and audio. To facilitate an in-depth visual analysis, even the less–complex interactive scenarios like using a desktop interface require advanced solutions that integrate and provide a consistent view of several data streams [54, 55]. Adding to the complexity, the inclusion of multiple users, their interactions, and movement in a 3D environment is beyond the current capabilities of the visualizations. Further research is required on visual encodings for such complex scenarios, understanding differences between in-situ and ex-situ analysis, and computational methods for interactively processing the recorded data streams from virtual and mixed reality sessions.

5.4.3 *Comparison and Abstraction*

Analyzing individual user sessions provides some insights but is limited in understanding the group dynamics. Only after considering several user sessions could it reveal typical usage strategies, common obstacles, and relevant misalignment. Visual comparison and aggregation of user sessions need to be supported. To compare different interactive (i.e., individual) sessions, temporal alignment and detection of similar behavior and actions become important. However, the recorded data reflects the users' actions on a low-level granularity. For comparatively evaluating the strategies employed by different users, we need to develop meaningful abstractions that can be reliably detected in an automatic or semi-automatic process. This process should be embedded in the visualization interface because analysts might need to adapt the definition of certain high-level strategies during the analysis.

5.4.4 *Beyond Mixed Reality Sessions*

We did not include any fine-grained constraints (e.g., by special hardware) that are applicable only in the case of mixed reality sessions to derive the design and application space. Hence, the proposed visualization categories can also be applicable to scenarios beyond such environments (RO 2.1). The categories could be helpful while designing visualizations in other scenarios, as demonstrated in the next two chapters. For instance, to visualize the cooperative and competitive interactions in a multi-player game (Chapter 6) or to understand the collaborative scheduling strategies for planning movement in a constrained environment (Chapter 7).

6 COLLABORATIVE AND COMPETITIVE MULTI-AGENT INTERACTIONS

In a dynamic environment, entity behaviors are interdependent. To achieve their goals, entities often need to make decisions based on their interactions with others and actions performed by others. For instance, to collaborate on a task, entities need to interact with partners and perform actions in a specific sequence. In contrast, competing entities usually counter the strategy of enemies through well-timed reactive action sequences. In both cases, the necessity of interactions among entities and dependency on the actions of others presents a unique challenge while analyzing the exhibited behaviors (RO 2.2).

Conventionally, the interaction behavior is analyzed from a sequential perspective. For instance, Bakeman et al. [156] state that a *"defining characteristic of interaction is that it unfolds in time."* In games, the sequential analysis has been helpful in understanding the behavior of players through the chronology of their actions (e.g., [157, 158, 66]). Modeling interactions as events, visualizations for temporal event data have been well studied (e.g. [159]). Recently, Guo et al. [16] presented a comprehensive survey of such visualization techniques. However, these approaches assume the event sequences to be independent. Hence, they are not feasible to extend and encode the interactions between two or more entities. As a result, the context of understanding their actions in interdependent dynamic scenarios is lost during the visual analysis.

In general, using computer games for Artificial Intelligence (AI) research is common and accounts for about 50% of all published work in the field [160]. Competitions with simulated game-based environments (e.g., [161, 162, 163]) have become a useful and harmless medium to test, train, and benchmark new AI algorithms [164]. The goal of multi-agent game-based environments is to help advance the research on training AI agents that can compete, cooperate, or do both to complete their objectives. To improve the performance of such agents, developers need to analyze the interdependent behaviors and strategies (i.e., the sequences of actions) learned by the AI agents. To this end, we propose a novel timeline-based visualization for the exploration of strategies executed by the agents in a specific game-based dynamic environment *Pommerman* [17]. The game environment was specifically designed to assess competition and collaboration among agents and features an active research community, thus serving as an ideal application to demonstrate our visualization.

This chapter is organized into six sections. First, the Pommerman environment is briefly described, along with the reliance on using playback for group dynamics analysis (Section 6.1). Based on the derived design goals in Section 6.2, the

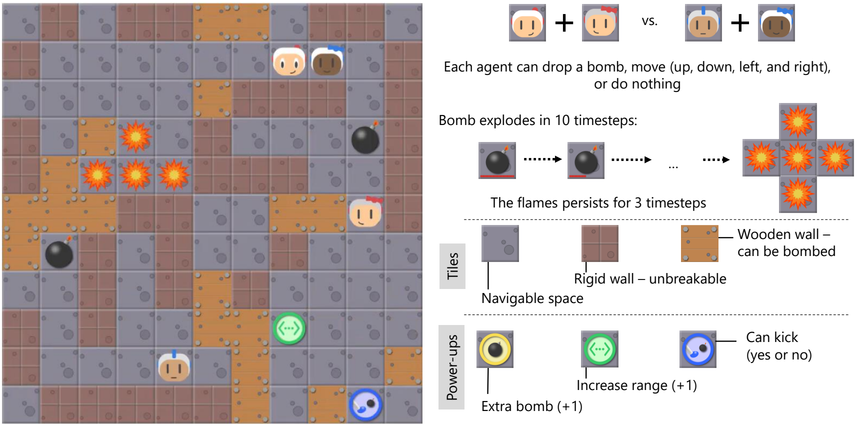

vs.
Each agent can drop a bomb, move (up, down, left, and right), or do nothing
Bomb explodes in 10 timesteps:
...
The flames persists for 3 timesteps
Tiles
Navigable space
Rigid wall – unbreakable
Wooden wall – can be bombed
Power-ups
Extra bomb (+1)
Increase range (+1)
Can kick (yes or no)

Figure 6.1: A snapshot of a game playback in Pommerman (left). The grid environment and the game mechanics are illustrated on the right.

design of our approach is explained in Section 6.3. The visualization explicitly encodes agent interactions with the in-game entities and marks important events affecting the behavior of agents in the environment on a timeline. To assess the usefulness of the approach, we analyzed the behavior of winning AI teams in a competition (Section 6.4). In addition, we also collected feedback from members of the Pommerman community and researchers from visualization and game analytics, describing the analysis results in Section 6.5. We end the chapter reflecting on the limitations of the approach and listing concrete ideas for future work in this research space (Section 6.6). An interactive web-based tool[1] implements the proposed approach. The supplementary material [165] includes the questionnaire and responses of participants from the user study along with the tool.

6.1 THE POMMERMAN GAME

Pommerman [17] is a variant of the classic multiplayer game *Bomberman* [166]. A game in Pommerman can have a maximum of four players. There are two modes: (a) all players compete against each other or (b) two teams, consisting of two players each, compete against each other. As shown in Figure 6.1, the map of the game is a board with 11 × 11 tiles where each tile can be a free navigable space, a rigid block, or a wooden wall that collapses when a nearby bomb explodes. The layout of the map is generated randomly for each game, but the starting positions of the players remain the same. Each player can lay a bomb, which explodes after a fixed duration (ten game steps). Flames from the bomb explosion persist for three game steps. Each player has to wait for the previously laid bomb to explode before

[1] Hosted at https://s-agarwl.github.io/bombalytics

laying another bomb. There also exist three types of power-ups limited in number and hidden beneath wooden walls, which offer: (i) an increase in the number of bombs a player can place simultaneously, (ii) an increase in the range of the bombs laid by a player, and (iii) the ability to kick bombs. To win a game, players (or teams in team mode) have to eliminate their opponents. In this work, we focus on the team mode, where players compete and collaborate to win a game.

The Pommerman game was built to train agents to compete and collaborate in a multi-agent environment [17]. A constraint on real-time decision-making (an agent has only 100 milliseconds to decide) makes it even more challenging to develop agents. But, the Pommerman community has already trained several agents via different techniques and tested them against each other [167, 168, 169, 170, 163, 171, 172]. Pommerman competitions are organized to promote research in this field, such as at the *NeurIPS* 2018 and 2019 conferences. Knowledge gained from these competitions has led to a better understanding of the underlying techniques. However, most commonly, performance analysis is done only on the number of games won by the agent, which hides the qualitative aspects of the behavior. This limits the ability of developers to investigate the learned strategies and further improve the performance of the agents. Developers can only watch individual games for a qualitative assessment, which includes checking for competition and collaboration strategies. This was confirmed by a developer of a top performing agent of the *Pommerman NeurIPS* 2018 competition, stating that: *"We find these [learned strategies] by running several battles and recognition by a human."*

6.2 DESIGN GOALS

For investigating agent behavior (collaborative and competitive) and comparing the performance of two teams, we first considered the goals that we deemed central for designing the visualization (RO 2.2). These design goals are based on informal communication with Pommerman community members, our experience in visualizing event sequences, and insights from related approaches. Beyond these specific goals, we tried minimizing visual complexity, using expressive labels, and building an intuitive visualization.

6.2.1 *G1: Overview of Event Sequences in a Game*

Currently, the developers of Pommerman agents use playback to analyze recorded games. While playback is useful in general, developers need to watch an entire animation to get an implicit overview of the event sequences in a game. However, to reduce the time required for analysis, it becomes important to obtain an explicit overview of the events that occurred in the game. The overview should display the distribution of events across the entire game, which could also point out different phases.

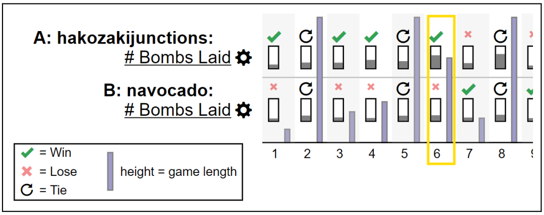


Figure 6.2: The summary component shows multiple games in a competition between two teams. For each game, it shows the game results through colored icons, the game duration through the height of thin purple bars, and a selected game metric for each team through dark gray bars.

6.2.2 *G2: Local Patterns and Repetitions*

Collaboration and competition strategies between agents are exhibited by interactions between the agents and specific items, for instance, kicking a teammate's bomb. The design of the visualization should support finding such local patterns. Since the same strategy might be executed several times in a game, the visualization should also show these repetitions. The developers currently rely either on summary game statistics or on playback to infer behavior patterns. However, aggregated statistics only provide an incomplete picture as they neglect the intermediate processes. On the other hand, identifying multiple occurrences of the same pattern of actions and movements in playback is tiring.

6.2.3 *G3: Overview of Multiple Games*

To compare two teams in a competition, usually 30–50 games are held. Hence, the visualization should also support statistical comparisons between two teams based on several metrics and provide a basis for selecting the most interesting matches for closer analysis.

6.3 VISUALIZATION APPROACH

We propose Bombalytics, a novel visualization approach, and implement it in a tool called *PomVis*. Figure 6.3 shows a screenshot of its interface, which consists of four components. Next, we discuss the data required for the visualization, followed by a description of each component in the interface.

6.3.1 *Data*

The Pommerman environment provides a command line option to record the state of a game at each step. Developers of autonomous agents for Pommerman use this

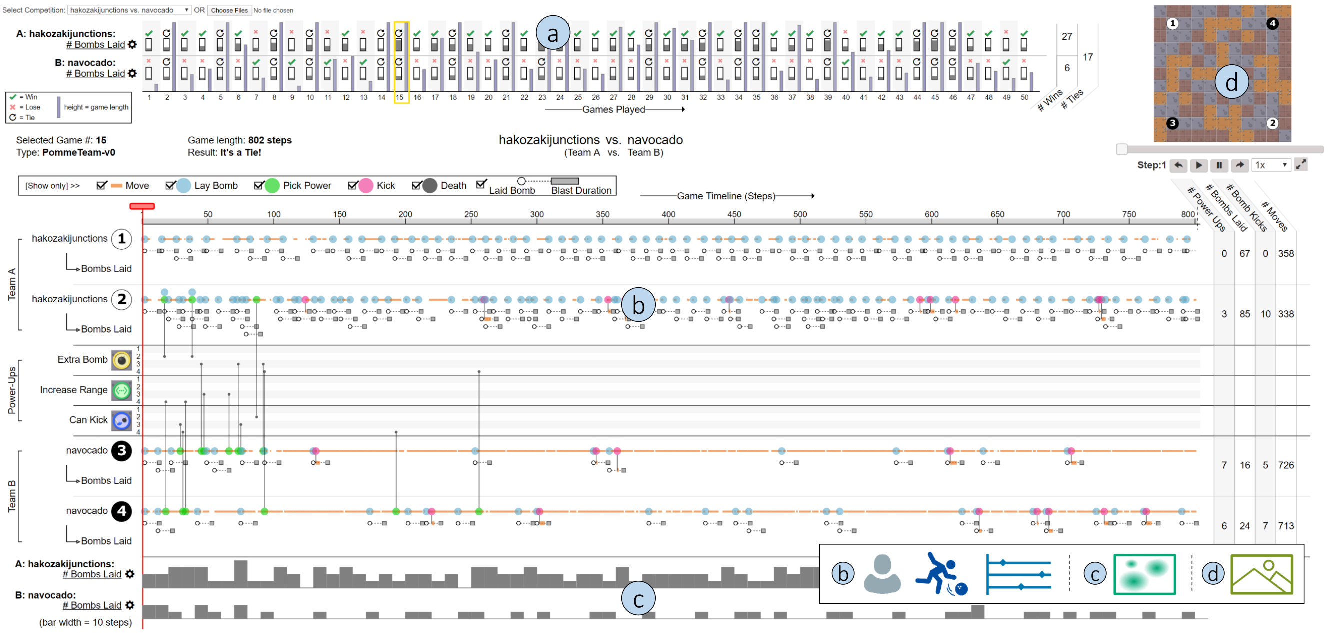


Figure 6.3: The *PomVis* interface consists of four components: (a) a summary of all the games in a competition, (b) a detailed timeline visualization of the selected game, (c) histograms to contrast the action densities of two teams, and (d) playback of the selected game.

option to analyze, e.g., the number of wins, losses, and ties. To enable easy and widespread use of our tool among the developers, we rely on this recorded data without further instrumentation of the game. The game states recorded in the data are used to generate a playback and a summary. We extract the actions performed by the agents and identify bomb explosions.

We analyze sample data consisting of six competitions, which were held between three agents of the 2018 competition (in the top 10 final rankings): *hakozakijunctions*, *navocado*, *skynet955*, and the *simpleAgent*, which is the default learning agent provided in the Pommerman environment. The executable container images of the agents were fetched from Docker Hub[2]. A team in our sample data consists of two instances of the same agent. Each competition consists of 50 games between the two respective teams.

6.3.2 *The Summary Component*

The summary component at the top of the interface (Figure 6.3a) provides a high-level overview of all the games in a competition (**G3**). Individual games are represented along the horizontal axis in columns and are numbered, as visible from the enlarged image in Figure 6.2. The two teams are shown as separate rows. The result of a particular game is represented as icons: *Win* (✓), *Lose* (✗), or *Tie* (↻). We compute seven different game metrics for each team in every game, specifically the number of

1. moves ('#Moves'),
2. bombs laid ('#Bombs Laid'),
3. kicks to bombs ('#Bomb Kicks'),
4. pick-ups for any power ('#Power-ups: Any Power'),
5. pick-ups for 'extra bomb' power ('#Power-up: Extra Bomb'),
6. pick-ups for 'increase range' power ('#Power-up: Increase Range of Bomb'), and
7. pick-ups for 'can kick' power ('#Power-up: Kick').

The values of one selected metric for individual games are visualized through dark gray bars placed in the respective rows of each team. The game metric can be changed by clicking the underlined label of the metric or the gear icon. The length of a game in a competition is encoded by the height of a thin light purple bar. The total number of wins and ties for each team are shown at the end of the rows (Figure 6.3a). Clicking a particular game column draws the detailed visualization of the corresponding game in the components below, as discussed next.

[2] (Accessed May 2023): https://hub.docker.com/u/multiagentlearning

6.3.3 *The Timeline Visualization of a Pommerman Game*

The static timeline visualization component (**G1**) is placed in the middle of the interface, as shown in Figure 6.3b. The horizontal axis represents the temporal progression of the game (timeline) and shows each step of the game in sequence from left to right. Each entity (a player or a power-up) is shown as a separate row in the visualization (*entity identifiers* 👤 and *entity timeline* ☰). Rows representing players are split into two parts: the upper part shows actions performed by the player, while the lower part shows bombs laid by the player. Separating the players from the bombs reduces the clutter on the timeline and allows identifying the lifespan of bombs, kicks, and blast duration more clearly, as explained later.

For a clear visual distinction between the two teams, rows of players belonging to Team A are placed at the top, while those belonging to players of Team B are placed at the bottom, as shown in Figure 6.3b. The rows of power-ups are added in the middle, as they denote common resources that can be utilized by any player. The separation between the rows of the two teams helps differentiate between inter- vs. intra-team player interactions (**G1** and **G2**).

Players may perform different actions in a game which we represent via the color and shape of different glyphs (*event identifiers* 🏃 ; **G1** and **G2**). A player can move (▬), lay a bomb (●), kick a bomb (●), and pick up a power-up (●). Bomb explosions are important in the game as they might trigger other events, such as the death of a player (●), the destruction of a wooden wall, etc. We represent each bomb by a shape (○┈┈┈▬) that has an unfilled circle at the head—indicating that the bomb was laid—followed by a rectangular tail—denoting the explosion of the bomb and its duration (i.e., three game steps). The head and tail of the bomb glyph are connected by a dashed line. Since the lifespans of bombs laid by a player can overlap (if a player has an 'extra bomb' power), we place them at different vertical positions in the lower part of the row of the corresponding player if necessary. Visually representing the lifespan of every bomb makes it easy to identify actions and events related to each bomb individually (**G2**). Selecting a checkbox of the legend items (placed above the timeline visualization) highlights the corresponding actions, events, and game objects (bombs) in the visualization.

Each row of a power-up is divided into four sub-rows of equal height, each corresponding to a player, as shown in Figure 6.4. Although this introduces some redundancy, doing so helps in quickly identifying the player associated with the corresponding power-up (**G2**). Also, it becomes easy to follow a sub-row and count the number of dots to infer how many instances of the power-up were picked by the corresponding player (**G1**).

Some events can be associated with multiple entities (players and power-ups) and game objects (bombs). To visualize this association, a vertical line is drawn between rows of the corresponding entities and/or game objects (**G1** and **G2**). Figure 6.4 shows interactions between Player 3 and different power-ups as well as

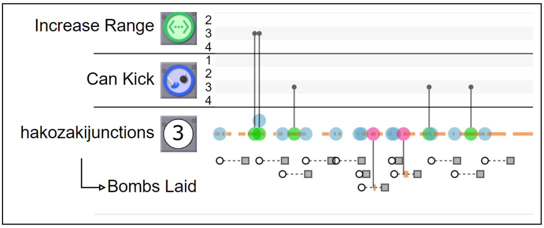


Figure 6.4: Vertical lines between rows show associations of a player (here, Player 3) with power-up rows when the player picks the powers and bombs when the player kicks them. In the example, first, the player picks two 'increase range' power-ups, followed by a 'can kick' power-up. Then, the player kicks two bombs and later picks two more 'can kick' power-ups.

bombs kicked. The movement of bombs being kicked is shown using an orange color in the timeline of the bombs (**G2**).

On the right side of the timeline visualization (Figure 6.3b), a few game metrics are shown in the columns for each player summed over the entire duration of the game (**G1**). The summed game metric values help in formulating hypotheses about the behaviors of teams and individual players. However, the behavior of players might not remain the same for the entire game. For instance, players pick almost all power-ups at the beginning of the game. To visualize the temporal distribution of the game metrics along the progression of a game (**G1**), we draw histograms (two rows, one for each team) as shown in Figure 6.3c (*event density field*). The game metric can be changed through selection. The bin size (bar width) in the histograms is ten game steps by default.

6.3.4 *Playback Component*

The components discussed before help identify the behavior of players and formulate hypotheses about the strategies they execute. To verify the formulated hypotheses, it is still essential to watch the actual playback of the game at a specific step of the game. To support this, we integrate a playback component on the top right corner of the interface, as shown in Figure 6.3d (*scene view* ; **G1**). The component includes standard playback controls. Navigation to a specific game step can be done via dragging either the slider placed above the playback controls or the red vertical status line in the timeline visualization (Figure 6.3b). The playback speed can also be modified.

6.4 APPLICATION EXAMPLE

Demonstrating the usage of the approach, we present a few strategies and unusual agent behavior identified through visual analysis of the winning AI teams in Pom-

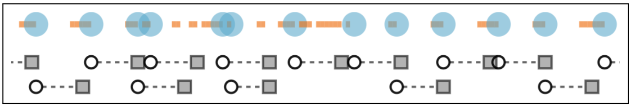


Figure 6.5: An excerpt from Game #8 of a competition between *hakozakijunctions* and *navocado* shows bold and suicidal moves by *hakozakijunctions*. The agent repeatedly lays a bomb, waits, and then moves when the bomb is about to explode.

merman competitions. The implemented prototype with the data of winning AI teams is hosted online[3] and available in the supplementary material [165].

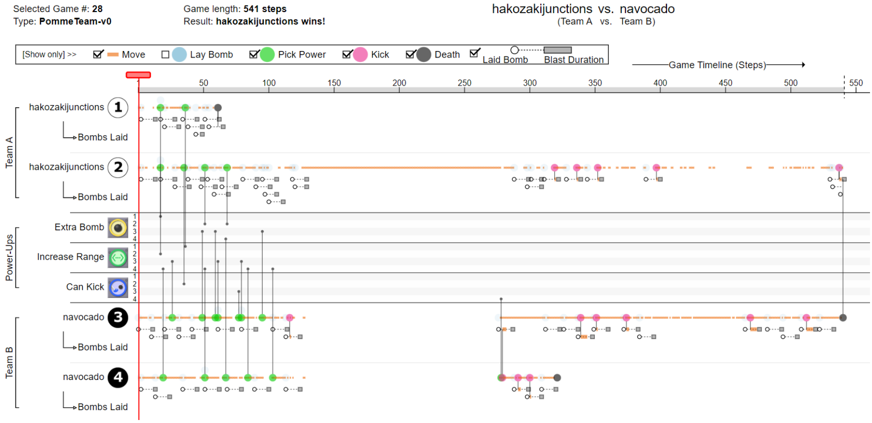
We analyze a competition between *hakozakijunctions* and *navocado* consisting of a total of 50 games. The summary component (Figure 6.3a) reveals that *hakozakijunctions* outperformed the other by winning 27 games and losing only 6, while 17 games resulted in a tie. Looking further into the summary component, we select the '# Bombs Laid' game metric and see that *hakozakijunctions* laid significantly more bombs in most of the games (dark gray bars, **G3**). However, on selecting the '# Power Ups' game metric, we find that *navocado* picked more power-ups in almost all the games. We further look into game #15, which resulted in a tie (↻), to explore details. Figure 6.3b reveals that both teams picked power-ups early in the game, inferred from the green dots and vertical lines (**G2**). However, one agent of the *hakozakijunctions* team did not pick any power-ups (Agent 1 in the first row), while Agent 4 of the *navocado* team continued picking power-ups in the later phase of the game, too (**G2**). The *hakozakijunctions* team moved less (few orange lines) and laid bombs more frequently (**G1**), inferred from the histograms below (Figure 6.3c) or from the last columns in the timeline visualization (Figure 6.3b). The *navocado* agents picked a lot of extra bomb power-ups but laid fewer bombs (columns at the end). The *navocado* agents moved a lot and seemed to explore the board (orange lines), which was confirmed via playback (Figure 6.3d) (**G1**). Agents 2, 3, and 4 laid and kicked their own bombs (pink circles and vertical lines), trying to kill the opponents (**G2**), but with no success. Eventually, the game timed out and resulted in a tie (**G1**).

Next, we list the discovered strategies and unusual behavior. Some of these strategies were also found by the participants of the user study (cf. Section 6.5.2).

Bold and suicidal move. The *hakozakijunctions* agents lay a bomb and stay on top of it. The agents only move when the bomb is just about to explode (**G2**). Figure 6.5 shows that this behavior is repeated throughout the game. The agents manage to eliminate opponents with this strategy, but in many games, they get killed by their own bombs.

Learn to kick bombs. It seems that the power of kicking a bomb makes a difference. In the six games in which the *hakozakijunctions* team was defeated, it was not able to collect 'can kick' power-ups, while *navocado* collected the power-up in these

[3] (Accessed May 2023) https://s-agarwl.github.io/bombalytics



Selected Game #: **28**
Type: **PommeTeam-v0**

Game length: **541 steps**
Result: **hakozakijunctions wins!**

Figure 6.6: An agent in *hakozakijunctions* was stuck in a loop by constantly moving between two tiles in game #28 against *navocado*, as indicated by the orange line in the middle of the timeline. During this, the *navocado* agents stopped and did not do anything (white gap in bottom two rows).

games (**G3**). In general, *hakozakijunctions* and *navocado* often kick bombs ('#Bomb Kicks' game metric). In many games, they also kick bombs laid by the other team (pink circles with lines to the other team rows) (**G2**). This behavior was especially exhibited in competitions with the *simpleAgent*, which does not kick bombs, even after collecting the 'can kick' power-up (**G1**).

Collecting redundant power-ups. The 'can kick' power-up is a binary property that, once picked, persists throughout the game. The *skynet955* agent has learned to avoid the redundant collection of 'can kick' power-ups. This can be seen from the summary component in competitions of *skynet955* vs. other teams and selecting the '# Power-up: Kick' metric (**G3**). However, as shown in Figure 6.4, *hakozakijunctions* collect the power-up more than once; it could be a strategy to prevent opponents from picking it up (**G2**).

Stuck in a loop. Sometimes, agents get stuck in a loop, repeatedly moving between two tiles. This is visible from long continuous orange lines in the timeline visualization (**G2**). For instance, as shown in Figure 6.6, in Game #28, the *hakozakijunctions* agent was stuck in a loop by repeatedly moving between the same two tiles, while the *navocado* did not do anything in the same duration (white space in the bottom two rows). The same behavior was observed in Game #14, where the *navocado* agent was stuck in a loop while its opponent waited idly. It shows that the agents have not learned to (a) avoid getting stuck in a loop and (b) exploit such vulnerabilities in opponents.

6.5 EXPERT USER STUDY

To evaluate the proposed Bombalytics approach, we administered an online questionnaire to AI, visualization, and game analytics experts. The feedback from AI experts verifies the capabilities and usefulness of the proposed technique. However, AI experts in the Pommerman community do not typically use visualizations (such as ours) while training the agents. As such, responses of other experts, in particular visualization and game data analysts being more experienced with such interfaces and analysis of player activity in general, verify the visualization design. The questionnaire and responses are provided as part of the supplementary material [165].

6.5.1 *Study Design*

The study consisted of an online questionnaire and an online version of the tool. Participants were asked to explore the tool and to go through the help page optionally before starting the questionnaire. The participants confirmed this preparation at the start of the questionnaire. Participants were allowed and reminded to switch back to the tool while filling out the questionnaire. The study was designed to take about 25 minutes, was conducted online, and ran for a period of 10 days. Participation was anonymous, and no identifying information was recorded.

Questionnaire. The online questionnaire consisted of seven parts. After explaining the purpose of the study and acquiring consent from participants (Part I), Part II asked participants to provide some background on their domain expertise on a 5-point scale labeled with *no knowledge*, *beginner*, *intermediate*, *advanced*, and *expert*. We also asked about their experience with Pommerman, playing *Bomberman* games, and whether they participated in Pommerman competitions by submitting autonomous agents. Parts III and IV asked about the summary component and detailed timeline visualization, respectively. Participants were presented with statements in these parts expressing the usefulness of the tool and were asked to rate them on a 5-point Likert-type scale anchored by *strongly disagree* to *strongly agree*. Optionally, the participants could provide detailed comments regarding what they liked and disliked about the above-mentioned aspects of the interface. Part V asked participants to textually mention the competition and collaboration strategies they were able to discover using the tool. It also asked to mention observed differences in the gameplay behavior of teams. In Part VI, we assessed the usability of the interface regarding four characteristics: efficiency, effectiveness, satisfaction, and overall [173]. We presented four statements for each category, which participants answered by selecting *Strongly disagree*, *Disagree*, *Neutral*, *Agree*, or *Strongly agree*. The participants could provide further comments on the usability of the tool. Part VII allowed participants to give additional feedback on tasks for which they would use *PomVis* and missing or unnecessary information in the tool, as well as to provide additional remarks.

Participants. The work presented in this paper aims to assist Pommerman AI developers by building a visual tool using research from the fields of visualization and game analytics. Consequently, we invited a diverse group of users to participate in the study and provide their feedback. First, since the tool specifically visualizes the gameplay data of Pommerman, we invited users who (i) make autonomous agents for the Pommerman environment, (ii) participated in the Pommerman competition, or (iii) have contributed to building the environment. Second, we invited visualization experts (in Information Visualization and/or Visual Analytics) who have research experience with event-timeline-based visualizations. Third, we invited researchers who have expertise in gameplay analytics. Finally, we strove for participants who also had considerable experience in either playing computer games or in programming and have played Bomberman games before. The invitations were sent via personal e-mail and through Pommerman's official Discord channel.

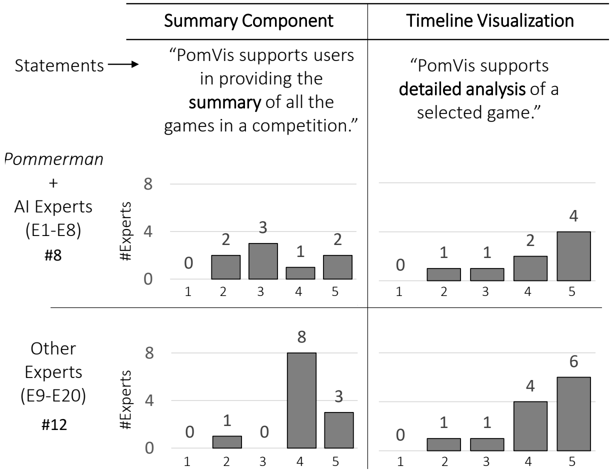
In total, 20 users participated in the study. We refer to these experts as E1 to E20 in the remainder of the paper. All 20 participants marked their expertise level as *expert* or *advanced* in at least one of the following five domains: *Artificial Intelligence*, *Playing Computer Games*, *Computer Programming*, *Information Visualization*, and *Game Analytics*. Expert E1 participated in both Pommerman competitions of 2018 and 2019, while four experts (E2–E5) participated only in the 2018 competition. Three other experts (E6, E7, and E8) also have experience in developing autonomous agents for the Pommerman environment without having participated in a competition. In addition, E6 contributed to Pommerman's code repository. Nine experts (E5, E9–E16) marked themselves as *advanced* or *expert* in *Information Visualization* and/or *Visual Analytics*. Three out of them (E7, E11, and E13) also considered themselves to have similar expertise in the domain of *Game Analytics*. We classify the experts into two groups based on their domain of expertise. Group A consists of Pommerman developers and AI experts (as the core user group of the tool, E1–E8), while Group B consists of visualization and game analytics experts (providing feedback with respect to visualization design and analytics, E9–E20).

6.5.2 *Results*

An inductive thematic analysis was carried out to analyze participant's responses per question.

Summary Component. Pommerman and AI experts mentioned that essential information is visualized in the summary component (E3, E4, E5, and E7). E2 liked the inclusion of data from multiple games in the tool as it helped to get an overview of a competition. Visualization and computer game experts liked the simplicity of the columns to the right of the timeline showing #wins and #ties (E9,

Table 6.1: Quantitative expert feedback about the usefulness of interface components; scale from 'Strongly disagree' (1) to 'Strongly agree' (5).



E16, and E17) and the static design of the component (E15). Seven experts liked the compact design of the component and highlighted that it gives a concise summary (E1, E2, E6, E8, E10, E11, and E18). Ratings in Table 6.1, however, show differences in opinions between the two groups of experts, with Pommerman and AI experts being more critical. Two experts (E1 and E5) did not find that the tool provides a good summary of all games in a competition. E1 noted in the comments the lack of a statistical summary of the games, e.g., the average number of bombs. Two experts (E3 and E6) mentioned that it took some time to understand the different encodings used in the summary component. In addition, others reported difficulties with interpreting the game length bars (E11 and E13) and differentiating them from the gray game metric bars (E17). Feedback from an AI and computer game expert (with experience in Pommerman), participant E6 summarizes this:

> *"It did take a bit to understand what was going on there, it's not that intuitive, in the sense that you need to see the legend to understand it. I also feel that even though I know how to, it's [a] bit hard to read them, some of this might have to do with the lack of spacing between the match columns."* – E6

Experts offered suggestions on how to improve the design of the summary component, such as showing details on demand (E12, E18, and E19) and additional statistics (E1, E4, and E9).

Detailed Analysis of a Selected Game. Overall, experts appreciated the timeline visualization (Figure 6.3b), which is also reflected in their ratings, as shown in Table 6.1. The experts highlighted that it provides a good overview of the selected game (E6, E13, and E19) in one screen (E4 and E8) and is informative (E17) while at the same time showing details of every action performed by the agents (E1, E3, E9, and E18). They liked the timeline layout and visual encodings (E2, E13, and E20) and commented that it is easy to read and understand (E1 and E6). Visualization expert E20 liked the overall layout of the view, with power-up rows being placed in the middle, separate rows showing the lifespan of bombs per agent, and vertical lines connecting bombs and agents for kick events. Three experts (E2, E3, and E20) appreciated the visualization of interactions through vertical lines. They mentioned the usefulness of highlighting events by hovering over legend items (E6, E11, E16, and E20). Experts were also fond of the playback component and its linking with the timeline visualization (E10, E12, E13, and E16). The detailed design and interactions were found useful in exploring the strategies of agents (E1, E9, and E16). Feedback from Pommerman and AI expert E1 summarizes the observations:

> *"[I liked the] extremely detailed but simple and easy to understand visualization! I really like the detailed component. You can quickly identify patterns in an agent's behavior via the timeline visualization and watch them happen in the visual playback." – E1*

While many experts appreciated the visual details, some mentioned that the timeline visualization is not easily readable (E11) and needs some time to understand (E2 and E19). The visualization contains too many circles (E12 and E15), which overlap (E2, E13, and E20) and make it a bit hard to understand or noisy (E2 and E20). The choice of colors, in combination with the transparency of the circles, created confusion while reading the timeline (E12 and E16). Two Pommerman and AI experts (E2 and E6) highlighted the inability to zoom/scroll on the timeline, which would have allowed them to focus better on a specific phase of a selected game. Two experts (E9 and E10) commented on the prominent central position of the power-up rows and instead suggested using symbols for each power-up in the individual rows of agents. E16 mentioned to have solely relied on the playback component to find strategies, whereas E4 used the playback to uncover interactions between agents. Four visualization or game analytics experts (E14, E15, E16, and E18) suggested that including spatial information in the timeline visualization could be helpful in finding position-based strategies. E15 recommended using heatmaps to show the most visited tiles over multiple games. It was also pointed out that the histograms provide redundant information (E16) and are difficult to understand (E17) as they lack legends and interactions. With respect to additional features, E2 suggested including the option to select multiple actions at once, while computer game expert E18 proposed showing the appearance of a power-up in the timeline.

Competition Strategies. Almost all participants (19 out of 20) reported at least one competition strategy they discovered. Three experts mentioned that picking more power-ups in the early phase of a game gives the team an advantage (E9, E10, and E18). Seven experts (E1, E2, E5, E6, E10, E12, and E19) highlighted that the strategy of kicking bombs helps a team win the game in general, while the three Pommerman and AI experts among them (E1, E2, and E6) pointed out that kicking a bomb that is about to explode seems to be more effective. Four experts (E10, E11, E15, and E17) mentioned that laying more bombs helps a team to win more games. Pommerman and AI expert E6 was able to discover the strategy to lay a bomb to restrict the movement of opponents. In contrast, E3 observed that the *navocado* team *"places a lot fewer bombs, as bombs also constrain the safety of agents in contrast to the Skynet agent, which places more bombs."* Two experts (E6 and E11) commented that teams moved around a lot in order to avoid being killed. Pommerman and AI expert E7 observed two priorities:

> *"This tool makes it easier to understand which agents are using different kinds of reinforcement learning, either more focused on a safe agent or a more aggressive strategy trying to win." – E7*

Sometimes, agents used their own bodies to block the movement of the enemy (E2). One Pommerman and AI expert (E4) mentioned that it is hard to see competition strategies speculating that *hakozakijunctions* might not have had sufficient computational resources.

Collaboration Strategies. Experts mentioned that agents of *hakozakijunctions* first engage in one-on-one combat with opponents (E2, E8, E14, E16, and E17) and, after killing one enemy, the two teammates team up against the remaining opponent by moving towards the enemy (E6, E8, E12, E16, E18, and E20). Five experts (E2, E8, E10, E16, and E20) highlighted the collaboration strategy to drive an enemy towards a corner of the board. Pommerman and AI experts observed that when a teammate is near, agents move away (E5) or do not lay a bomb next to their teammate (E6). Expert E15 observed that agents seem to kill themselves while ensuring the death of an opponent. E1 also observed a similar behavior:

> *"The first hakozaki agent seems to be a lot less aggressive than the second hakozaki agent. It seems like the first agent tries to survive while the second tries to eliminate other agents." – E1*

Five experts (E3, E4, E9, E14, E19) highlighted that it is hard to find collaboration strategies from the visualization. However, two Pommerman experts among them (E3 and E4) reasoned that the agents might not have learned complex collaboration strategies (*"I think Pommerman agents are still at a reactive strategy level and far from using more complex strategic behaviors." – E3*).

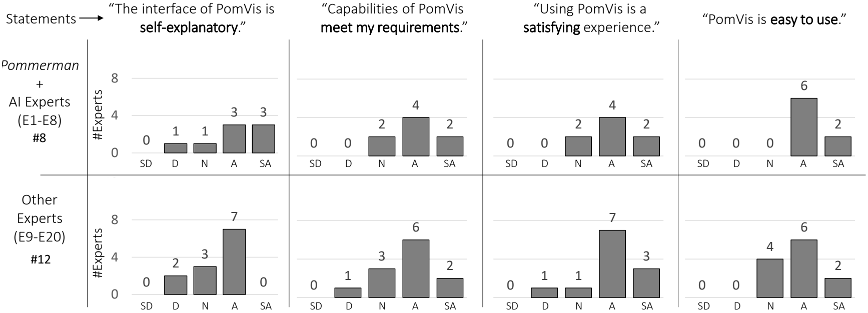
Differences between Behavior of Teams. The questionnaire asked participants to list observed differences between the behavior of teams. An expert in games and visualization (E15) provided detailed feedback that summarizes the characteristic behaviors of different teams, which were observed by other experts too (specified inside square brackets in the following). In particular, E15 mentioned (with other experts added having similar findings):

> *"**Skynet**:*
>
> *- lay many bombs [E3 and E14] in the beginning [E1], if only one player is still alive or all wooden boxes are cleared → just keep moving to escape bombs (no own bombs are laid)*
>
> *- defensive game play [E16]*
>
> *- collecting power-ups is not a goal*
>
> *- do not try to clear a path to the competing team or the own team mate [E12]*
>
> ***Hakozakijunctions**:*
>
> *- try to collect as many power-ups as fast as possible*
>
> *- clearing a path just in one direction to one opponent*
>
> *- lay many bombs throughout the game [E3, E10, E11, E17, and E19]*
>
> *- use the kick power-up a lot [E3, E8, and E19]*
>
> ***Navocado**:*
>
> *- try to collect many power-ups [E10 and E17]*
>
> *- clear a path to the opponent but not systematically [E18]*
>
> *- lay few [E11 and E17] but targeted bombs [E6 and E11]" – E15*

Other experts found additional behaviors but mentioned them without naming the teams. These behaviors include: agents idly waiting long times without performing any action or movement (E9), laying bombs on a regular interval (E19), action sequence pattern of lay bomb → kick → move (E13), and taking control of the diagonal field as a winning strategy (E2). Pommerman and AI expert E3 highlighted a behavior of the *hakozakijunctions* team—dropping many bombs followed by kicking them away—and mentioned that this is expected as it is a search-based agent.

Usability. The aggregated ratings on four characteristics of usability (self-explanatory, meeting one's requirements, usage being a satisfying experience, and ease of use) for the two disjoint groups of experts are presented in Table 6.2. All eight Pommerman and AI experts agree or strongly agree that the implemented tool is easy to use. Six of them agree or strongly agree that the tool is self-explanatory, meets their requirements, and using it is a satisfying experience. Two of them (E3 and E4) were neutral about the capabilities of the tool meeting their requirements. E3 wanted to see high level statistics, while feedback of E4 lacks details: *"To check if my agents are working as expected."* One expert (E2) disagreed with the statement that the interface of the tool is self-explanatory, which can be explained by a bug in the system he/she encountered and mentioned in the feedback—non-updating team labels and breaking the video player when switching competitions during video playback. The expert was among the first three participants of the study. It was not a critical bug and did not significantly impact the participants' answers,

Table 6.2: Results from the expert study show the usability characteristics of the tool. A response on each characteristic was recorded on a five-point scale [Strongly disagree (SD), Disagree (D), Neutral (N), Agree (A), Strongly agree (SA)].



but we fixed the bug to avoid a repetition of a similar experience for the remaining participants.

The ratings of Pommerman and AI experts followed a similar trend largely as those of other experts, as shown in Table 6.2. The majority of them found the tool to be easy to use (#8), self-explanatory (#7), and to provide a satisfying experience (#10). However, two experts (E11 and E19) did not find the tool to be self-explanatory because it is hard to establish the linking of the numbered images of players between the playback and timeline visualizations (E11). Also, the comparison features are missing (E19). Expert E19 mentioned a bug with the game lengths, but we were not able to reproduce it.

Three experts (E3, E13, and E15) mentioned that the interface contains too much information, which, as remarked by E3, *"is partly due to the nature of the game"*. All three suggested showing details on demand or only higher-level statistics. Two experts (E13 and E16) found that the icon used to show the bomb blast duration was unclear. Experts E17 and E19 mentioned that the help page of the tool was useful to understand the encodings in the visualization. Additionally, experts suggested using a permanent selection of an action (E7 and E19) which we implemented in the follow-up version of the tool.

Additional Feedback. In terms of possible application scenarios, Pommerman and AI experts mentioned that they intend to use the tool for analyzing (a) the behavior of the agents they trained (E1, E2, E3, E5, E6, and E7), (b) improving their agent's performance (E1, E2, and E7), and (c) understanding the AI algorithm used for training (E8). Most of the experts commented that the visualizations encoded important information required for analysis. Two experts (E10 and E19) highlighted that they did not use the histograms, with visualization expert E16 commenting that only one game metric ('# Power Ups') was helpful while using the histograms. Four experts (E14, E15, E16, and E18) emphasized the importance of spatial aspects in Pommerman, and one of them (E15) suggested visualizations such as heatmaps to show the density of player positions and bomb explosions. Experts also suggested incorporating additional features such as the ability to sort the games in the summary component by any game metric or game length (E14), highlight only associated actions and bombs on the selection of an agent (E14), and perform queries based on the strategies/patterns found (E6 and E12). Two experts (E8 and E14) proposed an interaction to jump to a particular game step by clicking on the game timeline rather than dragging the red status line. Experts also suggested to highlight when an agent was not able to make a decision within the 100 milliseconds time limit (E4), to provide explanations of the histograms (E17), and to include messages shared between the agents of the same team (E20).

6.5.3 *Validity and Limitations*

We strove for participants with varying expertise to ensure evaluation from different perspectives. We also invited participants with high expertise to ensure the quality of their feedback. It is, however, important to highlight that the authors had no previous connections with participants from the Pommerman community, who are the main target users of the tool. In contrast, the authors had a background in visualization and game analytics. The questionnaire did not ask participants to perform any specific task; rather, it asked users to explore the tool and describe their observations. Given the exploratory and qualitative focus of our study, we used a mixed-method analysis: qualitative analysis of the free-text responses combined with quantitative indicators for usability and usefulness.

6.6 DISCUSSION

The insights from the application example and results of the user study show that, in general, the approach is useful for understanding the dynamic behavior of agents. To be specific, the encodings in the visualization help to understand the collaborative and competitive entity interactions in the dynamic environment of Pommerman (RO 2.2). But the dense representation may become complicated to understand, especially for users who are less familiar with similar visual analytic systems or dashboards. This is intriguing as certain details are required to understand complex behavior, but at the same time, these details make the visualization more difficult to read. The participants in the study were able to find many interesting strategies of the three top-performing agents from the Pommerman 2018 competition. The study showed that by using the tool, they could identify competition and collaboration strategies. Furthermore, they were also able to find the characteristic behaviors of different teams. However, experts also highlighted the drawbacks of the approach and provided valuable suggestions on how to address them.

6.6.1 *Embedding Spatial Context in the Timeline*

Many experts mentioned the lack of spatial features within the timeline visualization. For simplicity, our design relies only on the playback to provide the spatial context and does not embed the information in the timeline. However, the collected feedback and existing literature in game user research (e.g., [174, 175]) highlight the need and value of spatial indicators during visual analysis. These indicators could help identify several strategies exhibited by the agents (e.g., trapping the enemy behind a narrow alley on three sides). Directly embedding the abstracted spatial information on the timeline might help while avoiding clutter, e.g., encoding an agent's proximity to other agents or bombs.

6.6.2 *Towards a Visual Analytics System*

As reflected in the feedback, the experts requested advanced features, such as querying, labeling the strategies, and finding occurrences of a pattern in multiple games. They also suggested showing higher-level statistics first and then presenting details on demand. These features point towards extending the approach to a visual analytics system. In doing so, the challenges of presenting dense information could also be handled by incrementally providing the abstracted information to the user, which is typical in a visual analytic system.

6.6.3 *Communication between Entities*

In addition to the survey responses, some participants and members of the Pommerman community also shared informal feedback through Discord. Being able to communicate with a teammate is a new feature in the Pommerman 2019 competition. Community members suggested including this information in histograms to reflect the temporal density of communication between teammates, which we implemented for a follow-up version of the tool. Visualizing communication between entities is also relevant and valuable for other environments beyond Pommerman. However, in dynamic scenarios where the communication protocols support the exchange of human-understandable words, only representing the temporal density will not provide sufficient details during the analysis. In that case, we need to extend the approach, to include other visualizations, e.g., word clouds.

6.6.4 *Alternate Uses of the Visualization*

Apart from using the approach during the development of AI agents, the Pommerman community members also expressed interest in adapting the tool to illustrate agents' behavior as part of presentations. The creators of the Pommerman environment used our approach to analyze the behavior of winning agents in the Pommerman 2019 competition and to present the final results[4]. Additionally, the approach was awarded in the Pommerman 2019 competition[5]. The experience opens up a research direction to further investigate alternate usage scenarios of visualizations in AI-based multi-agent games (e.g., in enhancing the spectator experience during live competitions).

6.6.5 *Generalizability*

The proposed approach is targeted at developers of agents for the Pommerman environment. However, going beyond AI agents, the proposed approach could be extended to the analysis of human players. Also, it would be applicable to analyze the group behavior in other dynamic scenarios where agents (or players) are split into two teams and the number of players is small. For instance, the approach could visualize multiplayer online battle arena games such as *League of Legends* [176], where team coordination is essential. However, the visualization would not scale to many players. While other games may feature many more in-game items than Pommerman, in many cases, these can be restricted to a small number that is most important, for instance, capture points in *League of Legends*. More generally, we envision parts of the approach applicable for diverse applications where analyzing interactions between entities (humans, robots, objects, etc.) is vital to understand group dynamics.

[4] (Accessed May 2023) https://bit.ly/3OT7Hbp

[5] (Accessed May 2023) https://twitter.com/Pommerman/status/1206101858336395264

7 SPATIO-TEMPORAL ANALYSIS OF AGENT INTERACTIONS IN PATH PLANNING

In the last chapter, we studied the event sequences and interactions between entities in a small-sized multi-agent environment. In larger environments, there is more space for movement and more entities to interact with. In these environments, an integrated analysis of the interactions and spatio-temporal analysis becomes essential to understand the exhibited group dynamics. The movement patterns and analysis tasks vary based on the characteristic features of a scenario (e.g., understanding movement on fixed paths) [177]. Thus, we focus on an environment where agents coordinate their interdependent movements while handling dynamic obstructions (RO 2.3), which are crucial to solving complex problems in multi-agent systems [178, 179, 180].

As mentioned in a survey [180], the application areas of such cooperating agents include usage in the city to anticipate future traffic behavior in simulated environments (e.g., [181, 182]) or to optimize scheduling (e.g., [183, 184]). To optimize the behavior of agents, visual analysis of their spatio-temporal interactions could lead to an understanding of their interdependence and coordination. Hence, a number of visualizations have been proposed to analyze the spatio-temporal behavior in a related scenario, i.e., traffic data analysis (see a survey [185]). However, in the scenario, entities act independently and do not coordinate their actions. Hence, inferring the performance of planning and coordination among the entities in such conditions is not possible.

In this chapter, we focus on a multi-agent environment where a fixed-track network constrains the agents and aims to analyze their coordinated movement behavior. Although we use a simulation environment to build our approach, in reality, the fixed tracks can be a physical delimiter, such as a rail or a virtual track that the agents must not leave (e.g., robots moving in a warehouse). More specifically, the visualization approach is based on the *Flatland* environment [18], a testbed for developing agents that act as trains that move on virtual rail networks. The goal is to schedule trains to reach their target destinations within minimum travel time, as explained in Section 7.1. As a first step, the analysis goals were formulated based on the interviews with three domain experts (Section 7.2). Then, the visualization approach, as explained in Section 7.3, was developed in two iterations. The first version was used to gather feedback from experts with different backgrounds and was later refined to analyze insights from the data of the *Flatland 2020 NeurIPS Competition*. Feedback from the experts (Section 7.4) and insights from analyzing winning submissions in the competition (Section 7.5) demonstrate the effective-

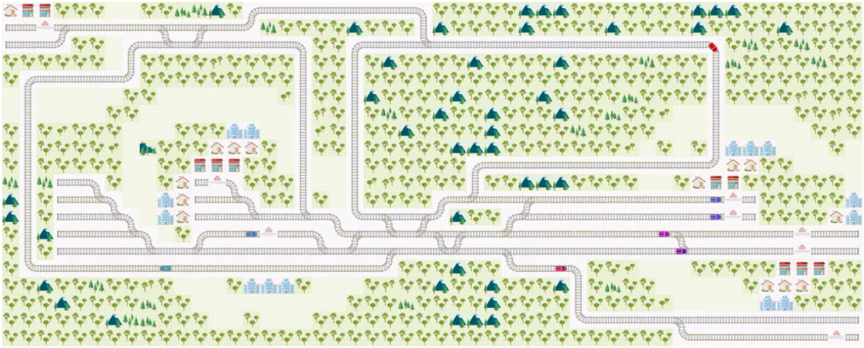


Figure 7.1: A sample map of the Flatland environment. The virtual trains, modeled as agents, move on the fixed tracks to reach their destination station.

ness of the approach. Finally, the chapter concludes with reflections on the lessons learned and future work (Section 7.6).

7.1 THE FLATLAND ENVIRONMENT

Flatland [18] is an open-source project that simulates train scheduling on different maps of rail networks. Each map is a 2D grid consisting of railway tracks on which the trains can travel and stations as destinations for the trains (Figure 7.1). Different levels exist that vary in grid size, number of stations, and trains to be scheduled. Within each level, different maps are produced that differ in the rail network layout and rate of malfunctions among trains. On a map, each train has a starting position on a track and a station as its destination. The trains do not have intermediate stops on their journey. Trains travel at a constant speed of one tile per timestep and cannot move backward. No two trains can be present on the same tile of the grid at the same timestep. The challenge is to schedule and steer the trains so that they reach their destination in minimum time. An episode in Flatland is one run of a scheduling technique on a map. Each episode has a maximum time limit, which scales with the size of the map and after which the episode ends, even if some trains are still on track. Also, trains can randomly experience malfunctions during the episode, which restricts their movement for some time.

In machine learning terminology, each train is modeled as an agent. Hence, it becomes a multi-agent scheduling problem, where agents need to collaborate and come up with an optimized schedule while handling malfunctions at runtime. The Flatland environment supports customizing agent observations (i.e., what each train sees in the rail network), which is crucial for agents to make decisions during runtime. To advance machine learning research, two Flatland competitions have been organized (in 2019 and 2020 at the AMLD and NeurIPS conferences).

While developing a scheduling technique for Flatland, the experts need to analyze the paths followed by the trains and discover issues (e.g., situations leading to a deadlock) to fine-tune the technique and improve its performance. Usually, the experts rely on watching episode playbacks, which is inefficient or can be inconclusive due to several key challenges. First, in Flatland, the effect of errors or sub-optimal agent decisions is only often visible much later in the episode. Remembering and linking the decisions of each agent to their effect becomes difficult in the playback. Second, understanding agent coordination behavior in Flatland is crucial for further improvement of the scheduling technique, which, however, can hardly be monitored in the playback. Currently, the experts rely on performance statistics (e.g., the percentage of trains that reached their destination) as a proxy for agent coordination behavior. However, it is insufficient, as they do not help the experts to understand and improve the qualitative aspects of coordination behavior in global and local areas of the network. Third, the experts need to identify unusual agent behavior within the context of the network (e.g., the formation of a long queue due to an agent waiting on a single track connecting distant stations). While individual issues like these might surface when watching the playback, it requires stepping back and forth to reconstruct their history, whereas other related issues might stay unnoticed. Fourth, a detailed comparison of alternate scheduling approaches, even on the same map, is impractical as two videos are difficult to watch at the same time. Also, watching them one after the other requires high cognitive effort to remember relevant details for comparison.

7.2 ANALYSIS GOALS

To understand the information needs of Flatland experts regarding the visualization of single episodes, we conducted interviews with three experts (*E1* to *E3*). Each interview took about 30 minutes. *E1* had a main role in organizing the Flatland competitions and providing technical support for participants. *E2* was affiliated with the Swiss Federal Railways (SBB – Schweizerische Bundesbahnen, a Flatland partner) and analyzed the scheduling behavior of trains in Flatland, as well as substantially contributed to the code base of the environment. *E3* is an artificial intelligence professional affiliated with the German railway company Deutsche Bahn (DB), also a Flatland partner, and explored agent-based scheduling techniques in Flatland. Based on the interviews, we derived five specific analysis goals (G1 to G5).

First, all experts wanted to obtain an overview of the schedules of individual trains in an episode (*E1*, *E2*, and *E3*). Expert *E1* highlighted the need to analyze junctions crossed by a train (as they are crucial locations for making a decision), the occurrence of malfunctions, and deadlock events. *E1* and *E3* also stressed the importance of statistics (e.g., the number of trains that reached their destination). E2 mentioned the need to focus on trains that did not reach their destination.

> **G1 – Overview of Schedules:** Get an overview of events and actions (e.g., departure time, junctions crossed, movement, etc.) for each agent. Also, inspect important statistics for an individual episode.

E2 highlighted the need to analyze the use of resources in the rail network, for example, finding out the busiest routes in the network. *E3* used specific environments to judge the scheduling behavior through resource utilization of a scheduling technique where the best-case scenario is known beforehand. The expert mentioned relying on playback, focusing on a lower number of trains, and observing the behavior by following individual trains.

> **G2 – Resource Utilization:** Analyze the utilization of resources in the network: tracks connecting distant places, critical junctions, and areas with a high number of stations or unusual agent behavior.

E2 also mentioned the need to assess the efficiency of train schedules, for instance, how delayed trains were in reaching their destination. Since this assessment needs some reference, the expert usually compares the actual path of the train with the shortest path, assuming there is no other train in the rail network. *E3* also mentioned that the shortest path plays an important role in deciding rewards, which is crucial for reinforcement-learning-based scheduling approaches.

> **G3 – Path Efficiency:** Assess the efficiency of the actual paths taken by agents.

Understanding the cause of issues (e.g., deadlocks, malfunctions, and bottlenecks) is important to improve a scheduling technique. *E2* highlighted the need to understand what has happened in the immediate past to investigate the reasons leading to a deadlock. Adding further, *E2* and *E3* mentioned that it is important to see how other trains reacted to these issues and to be able to observe which areas in the rail network were affected.

> **G4 – Issues:** Investigate the cause and effect of issues, e.g., deadlocks, malfunctions, and bottlenecks.

Finally, *E1* and *E2* highlighted the need to explore scheduling strategies exhibited by the collective and simultaneous movement behavior of a group of trains globally and in local areas. Since Flatland promotes experimentation with different scheduling techniques (e.g., reinforcement learning, operations research, hybrid approaches), the exploration of scheduling strategies should be model-agnostic. Such exploration is required to understand whether trains collaborate by reacting to the actions taken by other trains or not. The experts gave two examples: (1) using parallel tracks for one-way traffic and (2) trains following each other with minimal gaps.

> **G5 – Scheduling Strategies:** Explore the scheduling strategies through collective and simultaneous movement behavior of a group of agents.

An additional requirement resulted from later expert feedback (cf. Section 7.4) and was taken into consideration for a revised version of the approach. Participants suggested a comparison between two episodes (different scheduling methods or variants of the same method) on the same network. Such comparison would help developers experiment with new ideas and understand the differences between their approach and past top-performing solutions. It would also help organizers of the competition to explore and report qualitative differences in the scheduling behavior of different submissions.

> **G6 – Comparing Schedules:** Compare agent schedules, resource utilization, efficiency, and strategies between two different episodes.

To better understand the interests of the Flatland community and observe the setup of the *Flatland 2020 NeurIPS Competition*, one of the collaborators in the project—who has a background in visualization research—regularly attended the weekly community meetings held online for three months. Likewise, a person from the Flatland community collaborated with us and attended our meetings.

7.3 OUR VISUALIZATION APPROACH

Based on the analysis goals, we propose a visualization approach to help analyze the train movement behavior in Flatland episodes. The approach consists of three linked views providing different perspectives on the train movement data. Figure 7.2 and Figure 7.3 show the full interface of the proposed approach.

The approach was developed in two iterations. We implemented the first version during the *Flatland 2020 NeurIPS Competition* and collected feedback from a diverse group of experts. Based on the results, the approach was further extended in the second iteration after the competition ended.

7.3.1 *The Episode Selection Panel*

On top of the interface (Figure 7.2a), a line chart shows the percentage of trains that reached their destination along various test levels for each scheduling technique. Since there are multiple maps of similar size for each level, the line chart shows circles to mark performances on each map for all levels. A level, map, and scheduling technique for an episode can be selected by clicking on the circles in the line chart or via drop-down lists on the left. Once selected, we show statistics about the number of trains in the episode that needed to be scheduled, the percentage of trains that reached their destination, and the number of trains based on their end status (✓ that reached their destination, 🛣 still on-track, and ⏸ did not

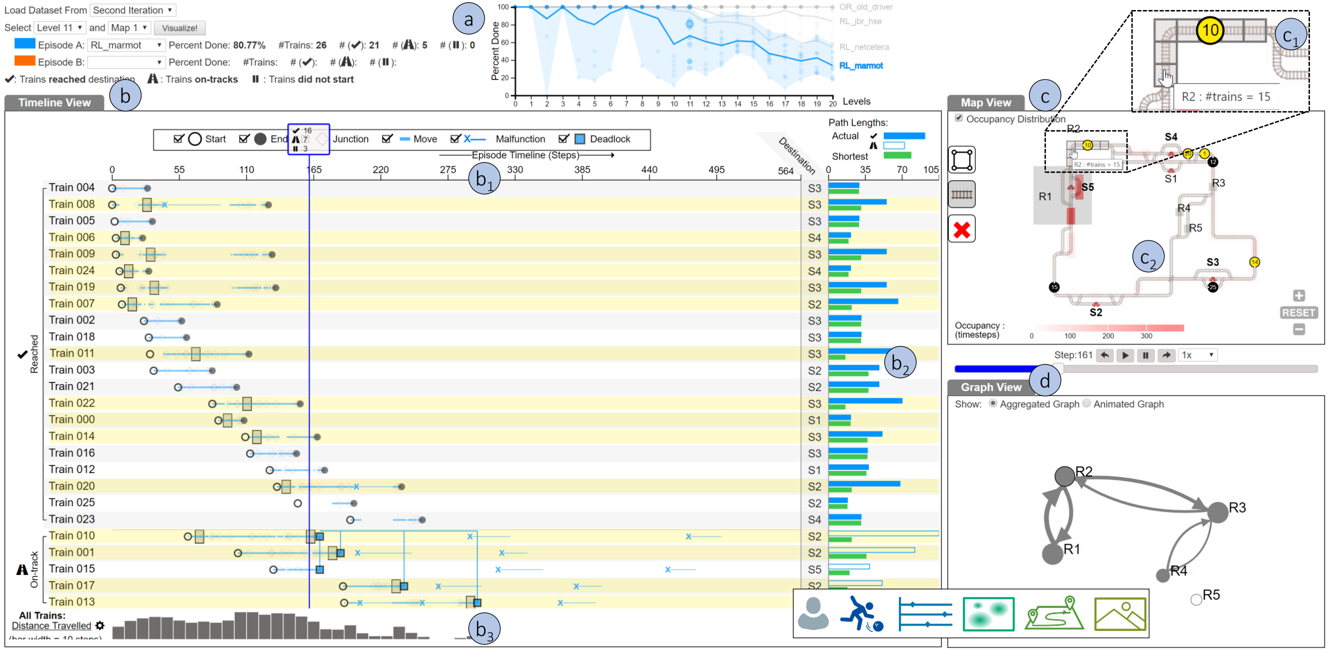


Figure 7.2: The proposed visualization approach shows trains scheduled using a reinforcement learning approach. The interface consists of (a) an episode selection panel, (b) a timeline view, (c) a map view, and (d) a graph view. Hovering over a region in the map view highlights in the timeline view the trains that visited the region (yellow background for highlighting, gray rectangles show the visit duration).

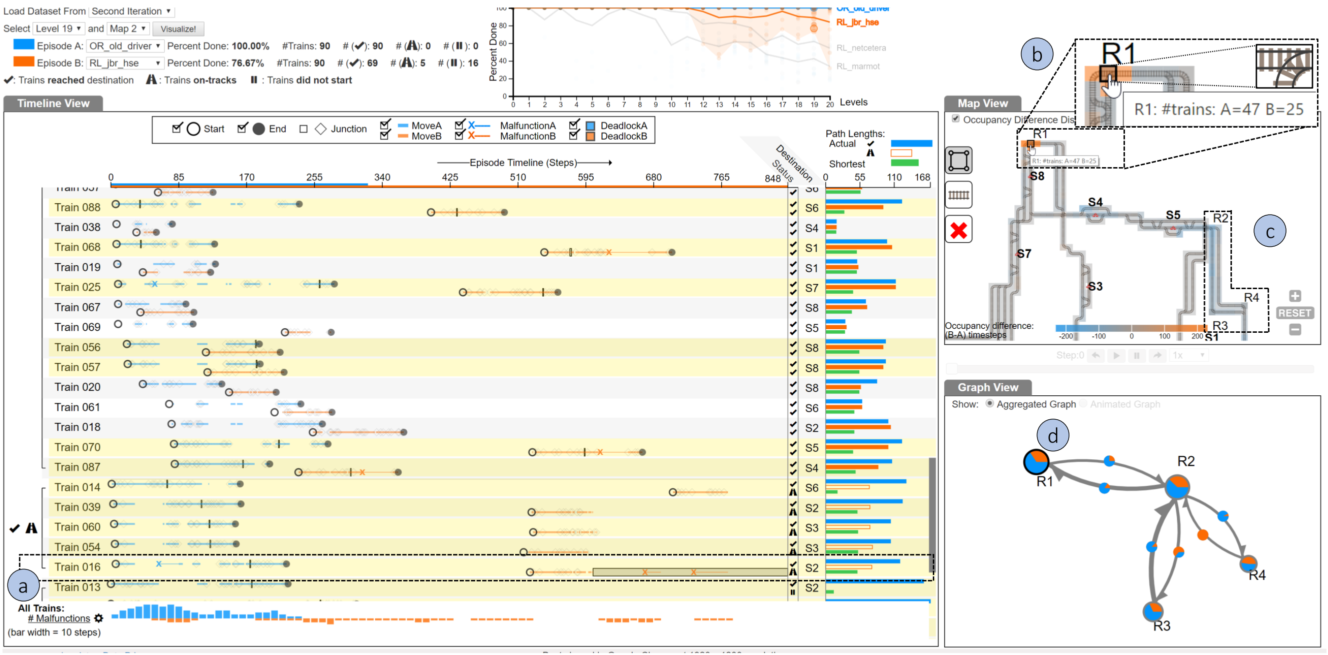
start) (**G1**). For comparing schedules from two episodes on the same map (**G6**), we use two colors—blue (🟦) and orange (🟧)—consistently across the interface to identify the unique characteristics of the selected scheduling methods in episode A and episode B, respectively (Figure 7.3).

7.3.2 *The Timeline View*

To provide an overview of an individual episode (**G1**) and compare two episodes (**G6**), we integrate a timeline view (Figure 7.2b). Positioned at the left, the timeline view visualizes the trains in different rows (*entity identifiers* 🟦), with time progressing from left to right (*entity timeline* ☰; Figure 7.2b$_1$). The developers usually analyze the trains based on their end status, for instance, assessing the path efficiency of the group of trains that reached their destination (✔) vs. exploring the reasons behind the group of trains that started and were still on-track (🛤) until the end of the episode. Hence, in the timeline view, the trains are grouped according to their end status. Within each group, trains are ordered based on their starting time.

For each train, we show the important events that occurred during its journey (*event identifiers* 🟧; **G1**). The departure of a train is encoded as a black ring (○), while a filled gray circle (●) denotes arrival. Since trains mainly make route decisions at junctions, we also encode reaching a junction by a diamond shape (◇). Movement is depicted as a colored line (▬, ▬). Malfunctions are represented by a colored cross with a tail (✕—, ✕—), where the length of the tail denotes the duration of the malfunction (**G4**). Aiming to visualize issues in the movement of trains, we use the only direct agent interactions present in the data: head-on collisions. We detect simple cases of deadlocks where trains on a track are moving towards each other without having their destinations on tiles of the track between them and no alternate routes to pursue. We represent detected deadlock events (**G4**) as colored squares (▣, ▣) for individual trains at the position when the train was blocked and connect it to deadlock events of other trains involved in the same deadlock via colored horizontal and vertical lines (cf. last five rows in Figure 7.2b$_1$).

At the right of the timeline, the destination station of each train is shown in a column, together with statistics of the path taken (Figure 7.2b$_2$). Since the origin of trains in the Flatland environment is not a station but an unlabeled location on a track, we do not specify the train origin in the timeline view. Furthermore, to assess the efficiency of a train's route (**G3**), we calculate the length of the actual path taken and compare it with the shortest possible path between the origin and destination, assuming that all tracks are available for movement. The actual path length is shown by a horizontal bar in the color of the respective episode, while the bars for the shortest path length are colored in green (▬, ▬). For trains that did not reach their destination at the end of the episode, the actual path length


Load Dataset From Second Iteration ▾

Select Level 19 ▾ and Map 2 ▾ Visualize!

Episode A: OR_old_driver ▾ Percent Done: 100.00% #Trains: 90 #(✓): 90 #(A): 0 #(II): 0

Episode B: RL_jbr_hse ▾ Percent Done: 76.67% #Trains: 90 #(✓): 69 #(A): 5 #(II): 16

✓: Trains reached destination A: Trains on-tracks II : Trains did not start

Figure 7.3: Screenshot of the proposed approach comparing the train schedules produced by an operations research technique by team `old_driver` and a reinforcement learning technique by team `jbr_hse` on a selected Flatland episode (`Level 19, Map 2`).

might be less than the shortest path length. To avoid confusion, we show the actual path length of trains on track by an unfilled bar (▭, ▭).

To provide a temporal overview aggregated across all trains (**G5**), we include a histogram at the bottom of the timeline view (*event density field* ▨; Figure 7.2b3). The height of each bar in the histogram represents the summed value of a selected metric aggregated across ten timesteps. In the case of comparison (**G6**), colored bars extend in opposite vertical directions with a common horizontal axis (▄▀▄▀▀). Available metrics, which can be selected by clicking on the gear icon (⚙), are (i) distance traveled (default selection), number of trains (ii) departing, (iii) arriving, (iv) experienced malfunctions, and (v) the number of crossed junctions.

7.3.3 *The Map View*

Positioned at the top right of the interface (Figure 7.2c), the map view provides a spatial perspective of train movement in the rail network. Instead of using the original map representation, which contains a lot of *graphical sugar* (e.g., irrelevant elements such as trees, cf. Figure 7.1), we decided to design a more abstract representation focusing on the rail tiles. The stations are labeled (e.g., *S1*), while each train is represented by a numbered circle, with a white dot showing its direction of movement. Since the size of the rail networks in Flatland episodes can be large, the map view supports zooming and panning interactions. The movement of trains on the rail network is shown using animations, which can be enabled through playback controls at the bottom of the view (*scene view* 🖼). To highlight the utilization of each rail tile (**G2**), we show on demand the occupancy time as a heatmap (Figure 7.2c) on a reddish sequential scale (░▒▓█). When comparing two episodes (**G6**), occupancy difference distribution is shown as a heatmap on a sequential scale from blue to orange (█████).

For analyzing the utilization of specific resources in the network (**G2**) and details of train movements (**G5**), it becomes important to focus on specific regions. Since different regions can be of interest during the exploratory analysis, with different strategies for defining regions, we opted for a flexible selection of the regions of interest. We provide two modes of selection in the map view, which can be activated by the two icon buttons at the left of the view. The first mode is a rectangular selection through mouse drag (⬚). With this, also, a single tile can be selected as a region by left-clicking. The second mode selects a clicked rail segment between two junctions (▥). The selected regions are assigned a label and are highlighted by semi-transparent gray rectangles (Figure 7.2c). All selected regions can be cleared by clicking the respective button (✖).

7.3.4 *The Graph View*

To analyze the movement of agents between selected regions (**G5**), we integrate a graph view in our approach. Positioned at the bottom right of the interface (Figure 7.2d), it contains a directed node-link diagram, where a node represents a selected region. Movement between regions is represented by the links between the nodes (*trajectory view*). The size of a node indicates the number of trains that have been in the corresponding region at least once. A train can move between two regions multiple times, and the width of a link represents the number of such transitions across all trains. Regions not traversed by any train are represented by unfilled circles. The node-link diagram is drawn using a force-directed layout, where the initial position of the nodes is set to the position of the selected region in the map view.

Two variants of the graph are selectable. First, the aggregated graph, as shown in Figure 7.2d, provides an overview across all timesteps highlighting the collective movement of agents in a static visualization. In the comparison mode (**G6**), as shown in Figure 7.3d, a pie chart is drawn inside each node of the aggregated graph to compare the number of trains in the selected region among the two episodes. Similarly, a pie chart for each link is added (positioned in the middle of each link) in order to compare the number of trains moving between the regions. Second, the animated graph, available in case only a single episode is selected for analysis, shows the trains moving between regions for the current timestep. For instance, an excerpt of an animated graph is shown in Figure 7.6a. The blue-colored node *R3* indicates that there are trains in the corresponding region, while a small circle on the link from *R4* to *R3* indicates a train moving between the regions.

7.3.5 *Interactively Linked Views*

With the selection of a single episode, playback can be viewed with the controls below the map view (**G3**, **G4**, and **G5**). This shows the train movement on the map view and the graph view (if the animated graph is active), along with the synchronized dark blue time-slider on the timeline view. Furthermore, we integrate highlighting consistently across the views as illustrated in Figure 7.4. Hovering over a train in any of the views highlights the corresponding row and circles with a yellow background in all views. For example, in Figure 7.4a, the row of *Train 009* was hovered in the timeline view, hence, the train was also highlighted in the map and the animated graph. Accordingly, hovering over a region in the map or graph view highlights all the trains that have visited the region. For instance, in Figure 7.4b, hovering over the rectangular region *R1* reveals that nine trains visited the region. Additionally, a semi-transparent gray rectangle in the timeline shows the periods spent in the region by the corresponding train (**G2**). Hovering over a station label in the timeline or map view highlights the trains heading to this station (Figure 7.4c). A supplemental video [186] demonstrates these interactions.

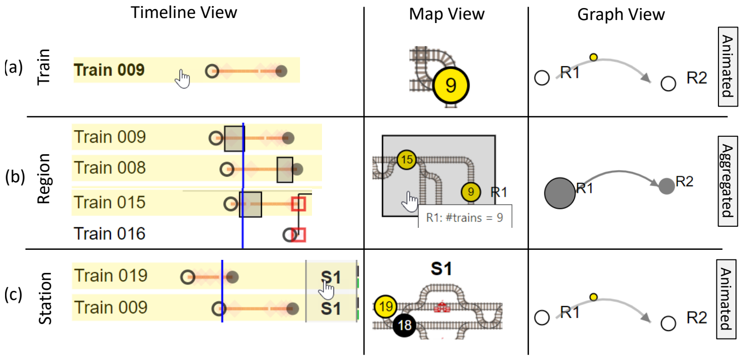


Figure 7.4: Linked interactions across the timeline, map, and graph view. Hovering highlights (a) an individual train (e.g., *Train 009*), (b) trains that visited the selected region (e.g., *R1*), and (c) trains with a common destination (e.g., *S1*) in all the three views.

7.3.6 *Dataset*

In the first version, we included the schedules generated by the two leading reinforcement learning and operations research approaches at the beginning of the competition. The generated schedules were for four different maps of the Flatland environment. Hence, we obtained data from eight episodes that can be selected by a drop-down list in the episode selection panel (Figure 7.2a). The name of an episode reflects the scheduling technique used: RL for reinforcement learning and OR for operations research. In the second version, we included the data from the top four winners of the competition across twenty test levels, each containing ten maps [187]. Hence, we added the data of (4×20×10 =) 800 episodes.

7.4 EXPERT FEEDBACK

We collected expert feedback on the first version of our approach in a questionnaire study providing the tool online. The first version did not contain the comparison features (which were added as a result of the study), the additional dataset from winning submissions, or the line chart showing summary statistics (Supplementary Figure 1 [186]). The survey was advertised on the Flatland forum, through social media and personal email, calling specifically for experts in artificial intelligence and visualization. For this study, the tool contained data from eight sample episodes from two scheduling techniques on four maps of the Flatland environment.

7.4.1 *Questionnaire*

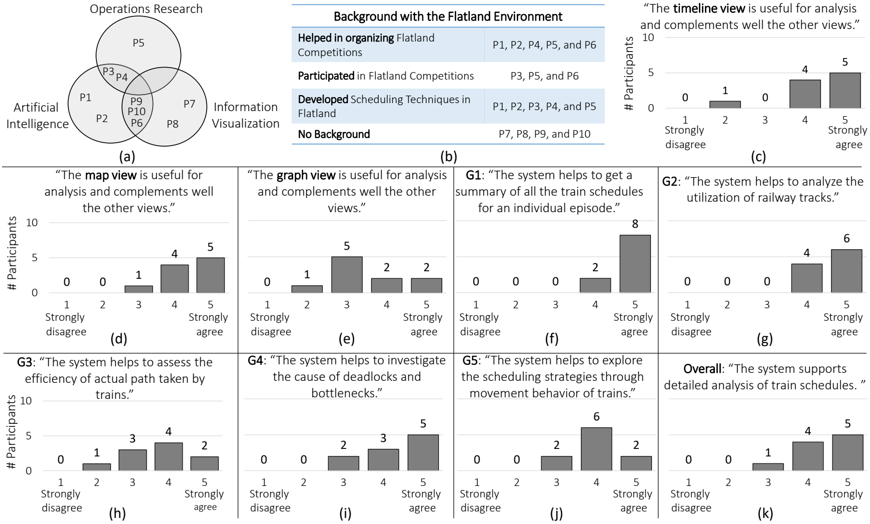
At the start of the questionnaire, demographic information of participants was gathered, such as their self-rated expertise in artificial intelligence, information visualization, and operations research (on a 5-point scale anchored by 1 = *no knowledge* and 5 = *expert*), and their background with the Flatland environment (cf. Figure 7.5b). Next, the participants were asked to explore and familiarize themselves with the visualization freely. An optional video explaining the available features and interactions was also provided to assist in getting familiar with the tool. This was followed by questions regarding insights gained. Participants were invited to name and describe up to three insights they found most interesting. Specifically, we also asked them to elaborate on if they found any differences between reinforcement learning and operations research approaches. Then, for each of the three views, participants were asked to rate how useful and complementary to the other views it was on a 5-point scale anchored by 1 = *strongly disagree* and 5 = *strongly agree* as well as to provide feedback on what they liked and disliked about it. Next, the survey inquired about the system as a whole, asking to rate its helpfulness with respect to the analysis goals (cf. Section 7.2) using the same scale as above and to reflect what they generally (dis)liked about it. Lastly, we obtained general feedback for which tasks the visualization was deemed helpful and if any important information or features were missing or unnecessary.

7.4.2 *Participants*

In total, 12 participants took part in the study. For analysis, we required participants to have good knowledge in at least one of the three fields (artificial intelligence, information visualization, or operations research). Hence, we include only responses from participants who rated themselves as 4 or higher on our expertise scale in at least one of the areas, resulting in 10 participants referred to as P1 to P10 in the following. As shown in Figure 7.5(a-b), seven participants (P1, P2, P3, P4, P6, P9, and P10) were experienced in artificial intelligence (AI), three (P3, P4, and P5) in operations research, and five (P6, P7, P8, P9, and P10) in information visualization. Six participants had some background with the Flatland environment, such as helping in organizing the Flatland competitions (P1, P2, P4, P5, and P6), participating in the competitions (P3, P5, and P6), or developing scheduling techniques in the Flatland environment (P1, P2, P3, P4, and P5).

7.4.3 *Feedback Analysis Results*

A summary of the feedback results can be found in Figure 7.5 (c-k). In the following, we report the results of the feedback analysis along with the structure of the questionnaire.



Background with the Flatland Environment	
Helped in organizing Flatland Competitions	P1, P2, P4, P5, and P6
Participated in Flatland Competitions	P3, P5, and P6
Developed Scheduling Techniques in Flatland	P1, P2, P3, P4, and P5
No Background	P7, P8, P9, and P10

Figure 7.5: Experience of the experts in the three areas (a) and their background with the Flatland environment (b). Expert ratings of the three views of the visualization (c-e), analysis goals (f-j), and the overall system (k).

Insights Discovered. Four experts (P1, P2, P3, P6) reported observing the occurrence of deadlocks. Two of them (P1, P6) found it insightful to see the trains linked to a deadlock, and P2 remarked to have been able to investigate situations in which a deadlock occurred. P3 liked *"that the moment of a deadlock is shown at the time it becomes inevitable, rather than when the trains actually stop moving."* P7 used the histograms at the bottom of the timeline view and observed that the occurrence of malfunctions was spread over an entire episode's length. Regarding the efficiency of paths, P1 reported comparing the lengths of the actual path and the shortest path of the trains. Furthermore, P7 observed that trains with a high number of junctions in their timeline have a much longer actual path length than the shortest path. P2 described using the visualization to *"see the density of traffic over time."*

Strategies for Defining Regions. The experts reported selecting rectangular regions (P2), single tiles of the grid (P7, P9), and individual rail segments in different combinations (P7) for further in-depth exploration. The experts also mentioned some of the important areas in the railway network that affect the scheduling. These include individual stations (P7, P9), dense areas with many junctions (P3) or nearby stations (P2), long single railway lines (P3), and single tracks at central positions (P1). An expert (P9) elaborated on using the heatmap to identify the most occupied parts of the network and selected them as regions.

Differences in Scheduling Behavior. Four experts (P6, P7, P8, P9) reported that, unlike the reinforcement learning solution, train schedules from the operations research had no deadlocks. Hence, the latter approach is able to schedule all the trains to their destination station (P6, P7, P8, P9). Highlighting a key difference between the two approaches, two experts (P3, P6) mentioned that the operations research approach shows a clear pre-planning of train paths until their destination. In addition, four experts (P1, P7, P8, P9) mentioned that the starting times in the train schedules of the reinforcement learning approach are spread across the entire episode length. P9 also reported that trains cross fewer junctions in the reinforcement learning approach.

Timeline View. Experts reported that they liked the timeline view as it provides an overview (P7, P10) and contains necessary information such as start/end events and duration of train movements (P8), deadlocks (P1, P6, P8), path lengths (P1, P4, P8), and density (P1), along with the distribution of the trains over time (P2). Two experts (P6 and P9) liked the linking of the timeline view with the other views, and P9 appreciated the sorting of trains based on their starting times. Experts disliked the inability to zoom in on a specific timespan (P1), reserved white space for trains that did not even start (P7), and the lack of a multi-selection feature for comparison (P8). P4 suggested extending the design to allow for the comparison of schedules from two solutions in the same view, while P10 remarked that the view contains too many details.

Map View. Nine experts (all except P5) rated the statement that *the map view is useful for analysis and complements well the other views* as 4 or higher (Figure 7.5d). The experts liked the heatmap (P5), the ability to follow individual trains exactly (P2, P7), the capability to define regions of interest (P3, P6), and to see the hovered region on the timeline of trains (P6). P1 stated to like the view because it "*allows to investigate in more detail situations like deadlocks identified in the timeline view*", which was also mentioned by P8. One participant, P9, was fond of the abstraction of railways, while P10 appreciated the simplicity of the map view. With respect to shortcomings, two experts (P1, P8) thought that the view is too small to show big rail networks. P3 suggested including a more flexible shape for defining regions in addition to rectangles.

Graph View. Four experts agreed (rating of 4 or 5), five experts were undecided (rating of 3), and one expert disagreed (rating of 2) with the statement that *the graph view (aggregated + animated) is useful for analysis and complements well the other views* (Figure 7.5e). Four of the neutral or negative replies (P2, P4, P5, P7) did not contain further details to explain the rating. However, P3 mentioned that it was unclear what was happening in the graph view, while P8 highlighted that deleting one region is currently not possible in the tool. On the other hand, three experts (P1, P2, P8) liked the abstract representation of train movement through graphs, a feature that P3 considered innovative. In addition, P7 mentioned that "*it was good to get the flow and amount of traffic between any two or more regions of tracks.*"

Furthermore, the experts liked that the view can be used to investigate frequent routes (P6) or situations in which trains visit regions multiple times (P4). P8 and P9 appreciated the animation in the graph view. P9 reported the inability to select a time range for the aggregated graph.

Analysis Goals. Except for one analysis goal (**G3**), ratings of the others reflect that the experts tend to agree with the statements that the system helps to achieve the respective analysis goals, as shown in Figure 7.5(f-j). Investigating the responses to understand the relatively low ratings for analysis goal **G3** (Figure 7.5h), P4, who gave a rating of 2, did not provide any details. The experts commented about the small size of the map view (P1, P9) and suggested including more linking between the timeline and map views (P7).

Overall System. All but one expert agreed with the overall statement that *the system supports detailed analysis of train schedules* (Figure 7.5k). Experts liked that the system is intuitive (P3, P8), interactive (P8, P10), provides the ability to analyze multiple aspects of the data (P2, P6, P9), and is useful for extracting insights (P1, P4, P7). However, experts also mentioned that comparing different scheduling approaches is difficult (P9) and highlighted that the agent observations (what each train saw; specific to reinforcement learning) are missing (P1, P3). Five experts (P1, P2, P3, P4, P5) commented that the visualization is helpful in diagnosing specific situations (e.g., deadlocks) and debugging scheduling solutions. Four experts (P4, P8, P9, P10) reported the usefulness of the visualization in analyzing the scheduling strategies and unexpected behaviors. In addition, P1 mentioned that the visualization is useful for experimenting with new reinforcement learning solutions, while P7 highlighted its usefulness for improving agent performance. The experts also listed several missing features they would use, such as filtering capabilities to analyze only specific trains (P1) and the ability to select a time range for analysis (P9). The experts also highlighted the need for an overview to select specific episodes (P9) and especially the ability to compare two scheduling approaches (P4, P6, P7, P8), which were implemented in the revised version of the prototype.

7.4.4 *Limitations and Discussion*

Due to the online setup of the feedback study, experts explored the tool freely without being monitored by us. However, given the informed answers they provided to the qualitative questions, we do not have reasons to believe that they did not sufficiently engage with the tool to make an informed judgment. Generally, the recruitment through connections to the community and personal invitation might have biased the results as experts potentially replied more positively and compliant. However, we tried to counterbalance by asking directly for criticism and options to improve.

The expert feedback indicates that the proposed visualization approach fulfills the analysis goals while addressing the specific challenges of the Flatland environ-

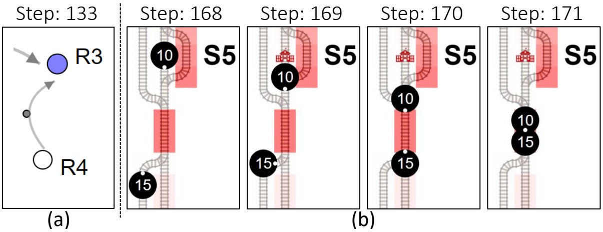


Figure 7.6: (a) The animated graph shows the movement of trains between the selected regions in step 133, while the map view (b) shows the occurrence of a deadlock between two trains (episode `Level 11, Map 1` by team `marmot`).

ment. However, the experts also suggested valuable features to enrich the analysis of scheduling techniques. Acting on the expert feedback, we extended our approach to include summary statistics and comparison features. Since the experts requested a detailed simultaneous comparison among two scheduling techniques (P4, P6, P8), we focused on enabling comparison between two episodes in the extension. Since the data logs in Flatland do not include agent observations, it was not possible to add this feature.

7.5 APPLICATION: FLATLAND 2020 NEURIPS COMPETITION

To illustrate one specific use case, we apply our approach to analyze the top submissions in the *Flatland NeurIPS 2020 Competition*. We used the second version of the tool, which includes comparison features. The competition was won by an operations research (OR) technique by team `old_driver`. The next three ranks were awarded to three reinforcement learning (RL) based solutions: 2nd position: `jbr_hse`, 3rd: `netcetera`, and 4th: `marmot`. For evaluation, the competition organizers used different levels with varying grid sizes. Within each level, ten maps with different rail network layouts and rates of malfunctions were used. Each team's submission was evaluated and compared on the mean normalized score to determine the final ranking. Generally, completing more levels with a higher percentage of trains that reach their destination in lesser time leads to a higher mean normalized score, among other variables that affect the score, such as local rewards for each agent (see [18] for details). Next, we present the insights found while analyzing the winning solutions with our approach. The tool is available in the supplemental material [186] and has also been hosted on the web[1].

Deadlock Propagation. Selecting an individual episode, Figure 7.2 shows the train schedules by team `marmot` on `Level 11, Map 1`. Overall statistics at the top show that ~81% of trains (21 out of the total 26) reached their destination, with five trains still on track until the end of the episode (**G1**). From the timeline view (Fig-

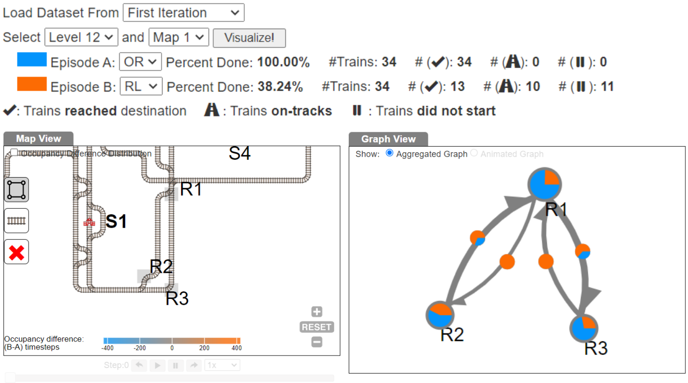
[1] (Accessed May 2023) https://s-agarwl.github.io/fv

ure 7.2b$_1$), showing the group of trains on the track at the bottom, we can see that all 5 trains were involved in a deadlock (blue square boxes connected with blue lines). We also infer that the deadlock first occurred between trains *010* and *015*, having different destinations (S2 and S5, respectively). Focusing on the two trains, using the playback controls, we navigate to the time before the deadlock. The two trains headed towards each other in opposite directions on a single track, leading to a deadlock, as shown in Figure Figure 7.6b. Knowing about this ineffective coordination in a specific scenario, suitable techniques can be used to improve the performance, e.g., better agent observations or communication among agents. Later, the deadlock propagated and affected the trains *001*, *017*, and *013*, all heading towards station *S2* (**G4**).

Inefficient Paths. Investigating the path efficiency between distant stations, we select a single railway line between station *S5* and stations *S1* and *S4* as region *R2*, as shown in Figure 7.2c$_1$ (**G2**). Assessing trains that reached their destination, we focus on the first group of trains (✔ *Reached*) in the timeline view. From Figure 7.2b$_2$, we observe that seven trains (*008*, *009*, *019*, *007*, *011*, *022*, and *020*) had a much longer actual path length than the length of the shortest possible path (**G3**). Then, analyzing the trains that did not reach their destination (🛣), we see that trains *010* and *001* have followed a much longer path than the shortest path length. The inefficient movement can be explained as both experienced malfunctions that could have altered their originally intended route twice.

Assessing Parallel Tracks Usage. Parallel tracks have several benefits that can be utilized by scheduling techniques. For example, considering them as one-way tracks avoids the possibility of a head-on collision, or they can be used as temporary parking spots, giving priority to other trains. To assess the parallel track usage, we need aggregated information on the direction and the movement of trains on the parallel tracks. We select three regions: two tiles on each of the parallel tracks (*R4* and *R5*) and a tile on a railway line common to trains using either parallel tracks (*R3*), as shown in Figure 7.2c$_2$. From the aggregated graph (Figure 7.2d), we observe that the reinforcement learning approach inefficiently used only one of the two parallel tracks (one with region *R4*) to move trains in both directions.

Comparing Usage of Parallel Tracks. To analyze the differences in parallel track utilization among the two scheduling techniques, we select three single tile regions on a parallel track in the map (Figure 7.3c). From the pie charts on the links between regions *R2*, *R3*, and *R4* in Figure 7.3d, we observe that OR strictly uses the two parallel tracks to move trains in the two directions. The RL technique also demonstrates this, but in a few instances, it used the left track from *R3* to *R2* (orange slice in the pie chart) to move trains in an upward direction rather than the track on the right from *R4* to *R2* (**G2** and **G6**). This was a specific way in which the RL approaches improved towards the end of the competition. Observing the data from the first iteration, the leading RL submission in the early phase of the



Load Dataset From First Iteration

Select Level 12 and Map 1 Visualize!

🟦 Episode A: OR Percent Done: **100.00%** #Trains: **34** # (✓): **34** # (A): **0** # (⏸): **0**

🟧 Episode B: RL Percent Done: **38.24%** #Trains: **34** # (✓): **13** # (A): **10** # (⏸): **11**

✓: Trains **reached** destination A : Trains **on-tracks** ⏸ : Trains **did not start**

Show: ● Aggregated Graph ○ Animated Graph

Figure 7.7: Cutout of the proposed visualization comparing usage of the parallel track by operations research technique and a reinforcement learning approach in the early phase of the *Flatland NeurIPS 2020 Competition*.

competition shows both the parallel tracks being used for moving trains in both directions (R1 ↔ R2 and R1 ↔ R3, as shown in the Figure 7.7; **G5**).

Frozen, Unable to Recover. Next, we compare the winning `OR` and `RL` scheduling approaches of the competition. In episode A, we select the `OR`-based approach by the team `old_driver`, as in episode B we select the `RL` solution by team `jbr_hse`, and `Level 19, Map 2`. From the statistics at the top of Figure 7.3, we observe that 90 trains had to be scheduled. The `OR` approach was able to schedule all the trains to their destination successfully. However, using the `RL` approach, only 69 trains reached their destination, while 5 trains were still on-track and 16 trains were still waiting to be scheduled before the episode timed out. Focusing on the group of trains in the `RL` approach that was still on track (✓ A), we observe that five trains (*014*, *039*, *060*, *054*, and *016*) were not blocked (absence of colored squares and connecting lines), but stood still for a long period of time (white gaps) after showing some movement (orange lines) (**G1**). Continuing the investigation in the map view, from the occupancy difference heatmap, we observe (three) orange-colored tiles (▨) on the top of the rail network, indicating that the trains from episode B occupied these tiles for a much larger amount of time than trains in episode A. Among them, to analyze the usage of a region in the rail network (**G2**), we select a tile that had a junction (╪), resulting in a region-of-interest labeled *R1*. Examining usage of the region for individual trains, we hover over the region and observe that one of the five on-track trains (train *016*) spent a lot of time staying on top of the tile until the episode ended (Figure 7.3a). This is intriguing as more trains in the `OR` passed through the tile than in the `RL` approach (pie chart inside *R1* in Figure 7.3d and tooltip in Figure 7.3b) (**G2** and **G6**). In Figure 7.8, we see that the trains got delayed (not blocked) because train *060* froze due to obstacles

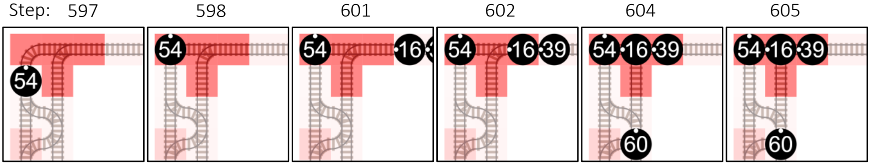


Figure 7.8: Trains are stuck by waiting forever (not a deadlock) in Level 19, Map 2, scheduled by the RL approach of the team jbr_hse.

on the two possible paths ahead, bringing three other trains (*016*, *039*, and *054*) to a standstill (**G4**). The observations suggest that the RL approach is unable to plan a conflict-free path for all the trains in certain scenarios, e.g., when there are obstacles on all possible paths ahead (**G5**).

Recovering from Malfunctions. Analyzing an episode `Level 19, Map 2`, as shown in Figure 7.3, the OR approach took much less time (colored lines in the episode timeline tick at the top) (**G1** and **G6**). Considering the fact that in the episode, the OR approach witnessed more malfunctions in the beginning (histograms at the bottom), it shows that the approach was able to deal with unplanned malfunctions effectively. The actual path lengths of the trains that reached their destination in both episodes seem to be similar (orange and blue bars on the right of the timeline view), demonstrating a similar path efficiency of the two approaches (**G3**). However, for the OR approach, we observe huge differences between actual and shortest possible path lengths in some trains with white gaps in their timeline, e.g., trains *014*, *039*, and *016* (longer blue bars than the green ones). This suggests that some trains in OR had to wait long and had to move far to reach their destination (**G5**).

Failing Cases in RL. We investigated the failing cases of agent coordination among winning RL approaches and documented our findings in Table 7.1. For analysis, eight episodes were selected in which not all trains reached their destination. The analysis revealed that jbr_hse occasionally runs out of time (due to episode timeout), sometimes has indefinite waiting cases in areas with a high number of junctions, and trains often just stay on top of junctions (**G1** and **G2**). In contrast, trains in netcetera often simply head toward another train from the opposite direction without waiting for them (**G5**). Trains scheduled by marmot also exhibit a similar behavior several times. In addition, sometimes, marmot moves the trains from their origin towards another train coming from the opposite direction on the same track, leading to a deadlock (**G4**).

7.6 DISCUSSION AND LESSONS LEARNED

The expert feedback and insights from the application example indicate the value of spatio-temporal analysis to understand the interactions and behavior of mul-

Table 7.1: Quantifying the reasons behind trains scheduled by RL approaches who did not reach their destination. The evaluation was done on eight different episodes.

		Episodes							
	Level number:	14				15	16		17
	Map number:	1	5	7	9	9	1	2	1
Starting the train in origin in opposite or wrong direction	*jbr_hse*								
	netcetera								1
	marmot					1		1	
Blocking a train from the opposite direction without waiting for them to pass through	*jbr_hse*								
	netcetera	2	1		1	2	2	1	1
	marmot	1	1	1	2		1	1	
Waiting indefinitely in areas with a high number of junctions or parking on junctions	*jbr_hse*	1	1				1		
	netcetera							1	
	marmot								
Episode timeout	*jbr_hse*			1				1	1
	netcetera								
	marmot								
Frozen trains, unable to coordinate when a viable solution is present	*jbr_hse*								
	netcetera		1						
	marmot								1

tiple agents (RO 2.3). More precisely, the visual analysis helped in understanding the coordination among agents and specific situations in which their performance could be improved. Based on our experience gained from designing the approach and expert feedback, we discuss the scalability and generalizability of the approach, along with the lessons learned that can be helpful for researchers building visual analytics solutions for analyzing multi-agent movement behavior in other related scenarios.

7.6.1 *Scalability and Generalizability*

Regarding scalability, the timeline view can accommodate ~30 trains without scrolling (a typical number in mid-size networks of Flatland). The approach is limited to comparing two episodes only. One reason is that the timeline groups the trains based on their end status (✔, 🛣, ⏸) and their combinations would grow exponentially with the number of episodes to compare. Although the approach was built specifically for Flatland, it could be potentially applied to other related environments. For instance, in analyzing coordination failures in multi-agent driving [188], e.g., identifying accident-prone areas, investigating deadlocks, etc. The approach can also be used to understand path planning efficiency, delays, and coordination of airport surface operations (e.g., fuelling, passenger boarding, luggage transit, etc.) through their movement on fixed paths connecting runways and the airport [189].

7.6.2 *Preserve a Static Map of Temporal Behaviors*

Animations are easy to follow. However, when required to analyze interdependent behaviors in detail, they demand a high cognitive load. Since the information changes quickly, e.g., tracking the movement of a group of agents, the analyst needs to remember a lot of information. Alternate techniques, e.g., 3D space-time cubes, address the challenges and enable the users to change their point-of-view, leading to new insights. However, a change in point-of-view challenges the mental map of the analyst. Thus, remembering the spatio-temporal attributes of multiple agents simultaneously becomes challenging. As a solution, the fixed timeline in our approach shows the actions of each agent through a static view. We base the layout on the context and analysis goals, e.g., grouping the rows based on the agent status at the end of an episode and ordering them based on their starting times within each group. This helped the users to construct a mental map about multi-agent behavior (e.g., blocked agents early in the episode), while interactions with the interface provided details without changing the overview of their actions.

7.6.3 *Interactively Define Spatial Focus and Map it to Time*

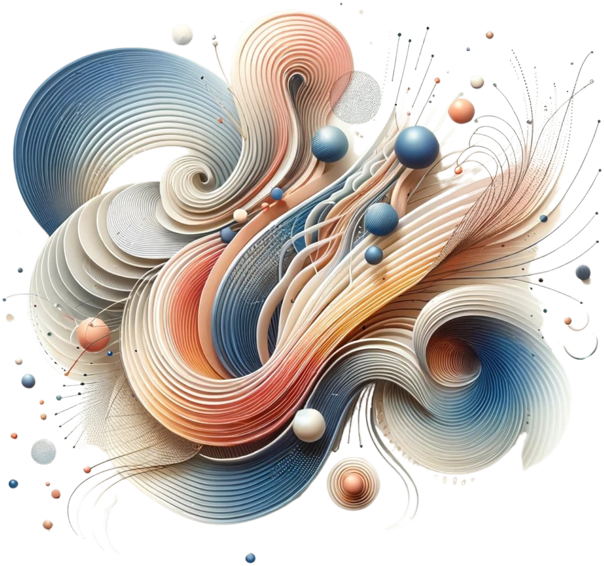
To understand the complex coordination behavior of agents, an in-depth analysis of spatial and temporal information is required. A usual approach is to have separate but linked views for each attribute. However, to study group behavior in multi-agent scheduling scenarios, they are not enough. We learned two things. First, domain-specific encodings and interactions help the analysts to focus on specific regions, e.g., the selection of a railway line between two junctions. Second, showing the effect of spatial selection on the temporal dimension helps reveal unexpected insights. For instance, on hovering over the selection of a region of interest in the map view, gray semi-transparent rectangular boxes are drawn in the timeline view. This helps in discovering extended stays in a region (e.g., *Train 016* in region *R1* in Figure 7.3a) or leads to insights such as cyclic movement of agents through a region-of-interest (e.g., *Train 010* through region *R2* in Figure 7.2).

7.6.4 *Abstract Space and Aggregate Multi-Agent Movements*

To understand coordinated behavior, experts rely on analyzing the collective movement of agents over key regions in the environment. For instance, the usage of parallel tracks. We learned that abstracting and aggregating the collective movement of agents over user-defined regions of interest helps to avoid details and gain insights about the scheduling technique (Section 7.5). This lesson aligns with the idea of spatial and temporal abstraction proposed by Andrienko et al. [190] to analyze patterns in mass mobility data. Generally, the abstracted and aggregated representations are put first in top-down exploration. However, in multi-agent scheduling behavior analysis, we realized that a bottom-up exploration was first necessary to identify specific agents and regions of interest. In a different context, van den Elzen and van Wijk [191] describe a similar bottom-up approach as *Detail to Overview via Selections and Aggregations*.

Part III

CONCLUSION



8 VISUALIZING ELEMENT INTERACTIONS IN DYNAMIC OVERLAPPING SETS

In the previous chapters, we explained the individual visualization approaches to explore complex group dynamics. The two important aspects of the group behavior—the changing memberships in overlapping groups and interactions among entities—were visually explored in individual prototypes for different scenarios. However, the two aspects were not explored together in a relevant scenario. In this chapter, we demonstrate through an example, how the proposed approaches could be extended to analyze the group behavior with respect to the membership in overlapping groups (RO 1.1) together with entity interactions (RO 2.2). Moreover, the extension also ensures a focused analysis of an entity group or comparison between two groups of entities (RO 1.2).

We model the multiple group memberships of entities as elements being members in different sets. Now, since the elements may also interact with each other over time, they mark an explicit connection between elements, which could also affect their memberships in sets. Hence, a joint analysis of both aspects, although challenging, could help to understand the temporal behavior better. For example, a company (element) acquiring another company to diversify its portfolio of offered products or services (sets), or collaborations between interacting members (elements) of research communities (sets). There does not exist a dynamic set visualization technique that encodes the memberships in evolving overlapping sets while showing the entity interactions.

In this chapter, we extend a proposed approach, Set Streams [104] (Chapter 3) by embedding the interactions between elements while showing their changing set memberships over time. We describe the proposed design in Section 8.1 and demonstrate its effectiveness by insight from two application examples in Section 8.2 (a) evolving business portfolio of interacting companies and (b) dynamic collaborations among researchers. Finally, we discuss the limitations of the design and ideas for future work (Section 8.3).

8.1 VISUALIZATION APPROACH

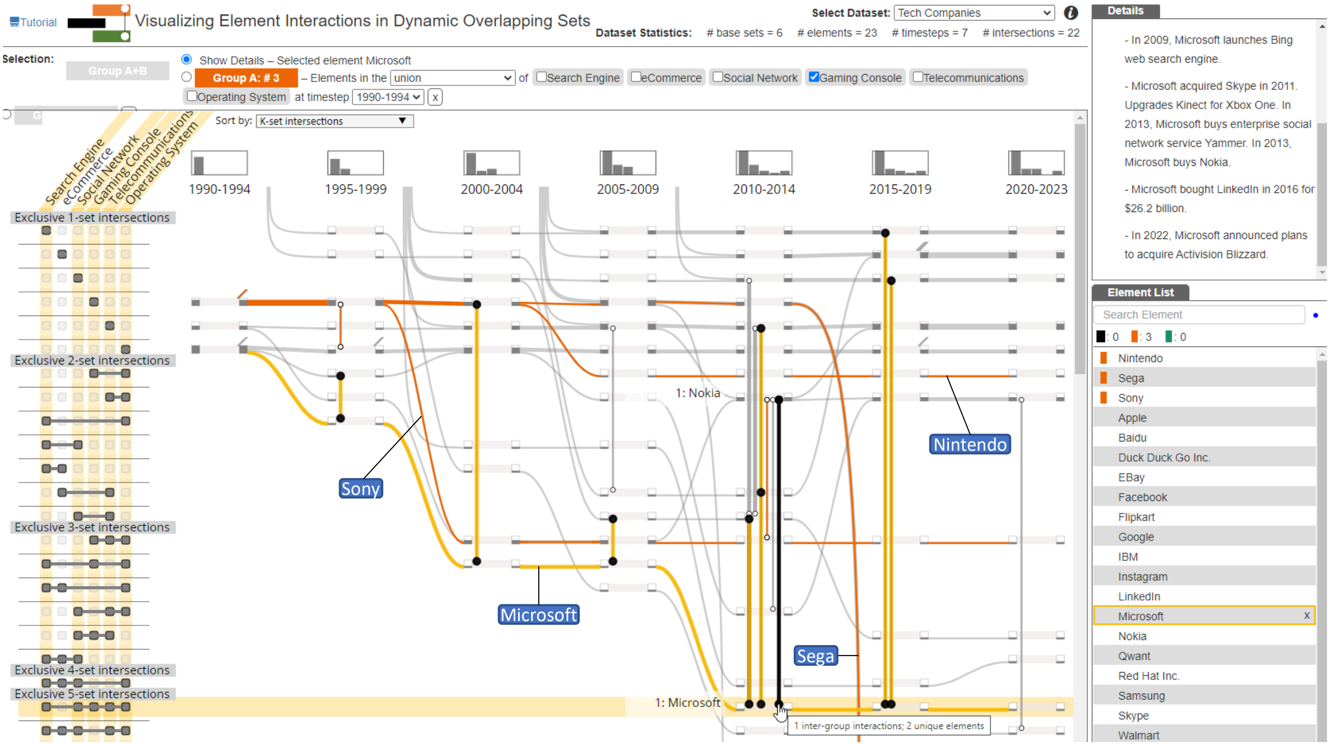
Set Streams [104] (Chapter 3), is a dynamic set visualization with a matrix layout where rows represent partitions of overlapping set regions as exclusive set intersections and columns show the timesteps. Considering an example, if set $\mathbf{A} = \{x, y\}$ and $\mathbf{B} = \{y, z\}$, then there are three exclusive intersections with elements: [only in $\mathbf{A}$] $= \{x\}$, [only in $\mathbf{B}$] $= \{z\}$, and [only in $\mathbf{A} \cap \mathbf{B}$] $= \{y\}$. The layout of the technique ensures low visual clutter, e.g., by partitioning the elements in exclusive intersections and dedicating space for embedding temporal information on a static timeline. Hence, the design suits our needs to embed the element interactions through appropriate encodings without making the visualization complex.

The design of partitioning was initially proposed in UpSet [24], a static set visualization, which avoids showing the multiple presences of an element in overlapping regions, with each exclusive intersection in a row. For instance, a highlighted row in Figure 8.1 shows the exclusive intersection of sets *Search Engine*, *Social Network*, *Gaming Console*, *Telecommunications*, and *Operating System*. The streams connecting adjacent columns encode the change in set memberships of elements, while its width shows the number of elements. The streams coming from the top at a particular timestep to a row represent the introduction of new elements in sets, while downward streams to the bottom edge indicate the elements do not belong to any set in further timesteps [104].

Embedding Interactions in the Timeline. We model an interaction as a hyperedge, which is a set of elements that are involved in an interaction at a specific timestep. To show the element interactions, we modify the design of cells representing an exclusive intersection at a specific timestep in the matrix layout. In each cell, we put bars at the two ends to encode the number of contained elements ( ). Inside the empty region of a cell, we draw a vertical line connecting rows (by small circles), showing aggregated interactions between elements in the respective exclusive intersection rows. Interactions between elements within the same exclusive intersection are aggregated and shown as a skewed rectangle on the top right border of a node. The width of the skewed rectangles and vertical lines encode the number of interactions within the same and between different exclusive intersections, respectively.

Sorting the Rows and Interaction Edges. Set Streams has options to sort the rows, e.g., by exclusive (k−) set intersections groups (default), decreasing order of cardinality in a timestep, etc. In addition, we implemented a row sort option by the sum of the number of interactions across all timesteps. Within a timestep, the interaction edges encoded as vertical lines are packed using first-fit greedy algorithm [192]. Doing so reduces the required horizontal space within a node. Adjusting the column width for each timestep based on the number of interactions would save even more space. However, we chose to have a fixed column width for all timesteps to avoid confusion.

Linked Interactions. Hovering over an aggregated interaction hyperedge temporarily shows the labels of participating elements in the first five interactions, in the respective rows. For instance, the hovered vertical line in Figure 8.1 shows an interaction between *Microsoft* and *Nokia*. To show details, e.g., the involved elements in an intersection, we add a *Show Details* option as a radio button (Figure 8.2 top). Once checked, on the left-click of a hyperedge, the details are shown in a panel on the top right (Figure 8.1). On selecting an element from the *Element*



Visualizing Element Interactions in Dynamic Overlapping Sets

Select Dataset: Tech Companies

Dataset Statistics: # base sets = 6 # elements = 23 # timesteps = 7 # intersections = 22

Selection: ● Show Details – Selected element Microsoft

○ Group A: # 3 – Elements in the union of □Search Engine □eCommerce □Social Network ☑Gaming Console □Telecommunications □Operating System at timestep 1990-1994

Sort by: K-set intersections

Details

- In 2009, Microsoft launches Bing web search engine.

- Microsoft acquired Skype in 2011. Upgrades Kinect for Xbox One. In 2013, Microsoft buys enterprise social network service Yammer. In 2013, Microsoft buys Nokia.

- Microsoft bought LinkedIn in 2016 for $26.2 billion.

- In 2022, Microsoft announced plans to acquire Activision Blizzard.

Element List

Search Element

■: 0 ■: 3 ■: 0

Figure 8.1: A screenshot visualizing interactions between elements in dynamic sets. Yellow lines show *Microsoft*'s expanding business portfolio and interactions with other companies (acquisitions and partnerships). Four streams have been annotated with element names.

List, the yellow-colored streams and hyperedges are emphasized (by setting their width as 5 pixels) to highlight its memberships and interactions. Hovering over a row emphasizes the respective interaction hyperedges, as shown in Figure 8.2 for exclusive intersection [*NLP*, *AI/ML*]. Encoding the selection of two groups of elements works as proposed in Set Streams: orange color shows elements in group A, green for group B, and black for common elements. The element list is sorted alphabetically by default, and on the selection of a group, the corresponding elements are reordered: first elements in both groups A and B, then those in group A, then B, followed by the remaining elements. The visual query selection mechanism of the groups is extended to include the interaction hyperedges.

8.2 APPLICATION EXAMPLES

Next, we discuss insights from two application examples. The implemented prototype with the dataset of application examples is hosted online[1].

8.2.1 *Evolving Business and Interactions among Companies*

Interactions (e.g., acquisitions and partnerships) between companies are common. These interactions reflect business decisions and strategies that affect a company's portfolio. For this example, we manually collected a dataset of 23 companies (elements) that offer products or services in six categories, namely, *Search Engine*, *eCommerce*, *Social Network*, *Gaming Console*, *Telecommunications*, and *Operating System*. We collect the information from 1990 to 2023 and divide the duration into seven timesteps, each representing a period of five years. It should be noted that the dataset has been checked for its correctness, but it is not a complete record of products or services or all interactions between the included companies.

Expanding Business Portfolio. Horizontal downward streams (in default sorting), connecting rows from different $k-$set intersections, indicate the expanding portfolio of companies. As shown in Figure 8.1, the expanding business of a selected company *Microsoft* is visible through yellow colored edges going down with a summary in the *Details* view. Until the last timestep, *Microsoft* offered products and categories across all six categories, except *eCommerce*. Highlighted hyperedges show the interactions of *Microsoft* with other companies. For instance, the hovered line in the timestep 2010–2014 shows an interaction between *Microsoft* and *Nokia*. The details of the interaction reveal that *Nokia*'s mobile and devices division was acquired by *Microsoft* in 2014, which is how it ventured into the *Telecommunications* business. Also, in the same timestep (2013), *Microsoft* acquired *Yammer*, an enterprise social network service, and started its business in the *Social Network* market (as seen from the details in Figure 8.1 top right). Investigating interactions between other companies in a similar way reveals similar insights, such as, *EBay*, an *eCom-*

[1] (Accessed May 2023) https://s-agarwl.github.io/sets_interactions

merce company that expanded its business by acquiring stakes in *Skype* (who made *Telecommunications* software) in 2005. But later, *EBay* sold the shares to Microsoft in 2011, and narrowed its focus back to the original business of *eCommerce*.

Early Companies in a Niche Market. Focusing on companies who initially made *Gaming Console*, we specify a query in selection A to show the elements in the union of the set at timestep 1990–1994 (Figure 8.1 top). The resulting three companies are shown in orange-colored streams. *Sony*, who made *Gaming Console* only, expanded its business in the timestep 2000–2004 by making *Operating System* (*Orbis OS*, for *PlayStation 4*) and *Telecommunications* devices. Similarly, *Nintendo* started making an *Operating System* for its gaming console in the timestep 2005–2009, called *Nintendo DSi system software*, followed by *Nintendo 3DS system software* in 2011 and *Nintendo Switch system software* in 2017. An orange-colored vertical line in 1995–1999 shows a partnership with *IBM* to make processors for *Nintendo*'s consoles. On the other hand, *Sega* used to make only *Gaming Console* but stopped doing so after 2014 and did not offer any products or services in the included categories. Changing its business strategy, *Sega* partnered with other companies that made *Gaming Console* to make games for them. An interaction in 2000–2004 shows its partnership to make games for *XBox*, a console by *Microsoft*.

8.2.2 *Dynamic Collaborations among Researchers*

We collect the dataset of scientific publications in five areas of computer science and model them as sets: *NLP*, *AI/ML*, *Graphics/Vis./HCI*, *Computer Architecture*, and *Software Engineering*. There are 380 experienced researchers in the filtered dataset, with at least 30 publications between 1996 and 2019. The duration is divided into six timesteps, each showing a range of four years. We model the co-authorship in a publication as the interaction hyperedge between involved researchers in a timestep.

Consistent Intra-group Interactions. In exclusive 1-set intersections across all timesteps (the first five rows in Figure 8.2), we observe a steady presence of the skewed rectangles on the top right of a node. It indicates that authors publishing exclusively in one field have a stable record of co-authorship interactions within the community. Additionally, we see a drastic rise in the number of interactions between authors publishing exclusively in the fields of both *NLP* and *AI/ML* (increasing width of skewed rectangles in the sixth row of Figure 8.2).

Early Cross-disciplinary Collaborations. Being interested in *AI/ML* and *Graphics/Vis./HCI*, we wanted to explore who published early in both fields. Hovering over the node of the first timestep in the exclusive intersection (the highlighted row in Figure 8.2), we find there is only one such researcher, *William T. Freeman*. We selected the author from the element list, which highlighted the author's journey with yellow colored edges, as shown in the Figure 8.2. The horizontal yellow

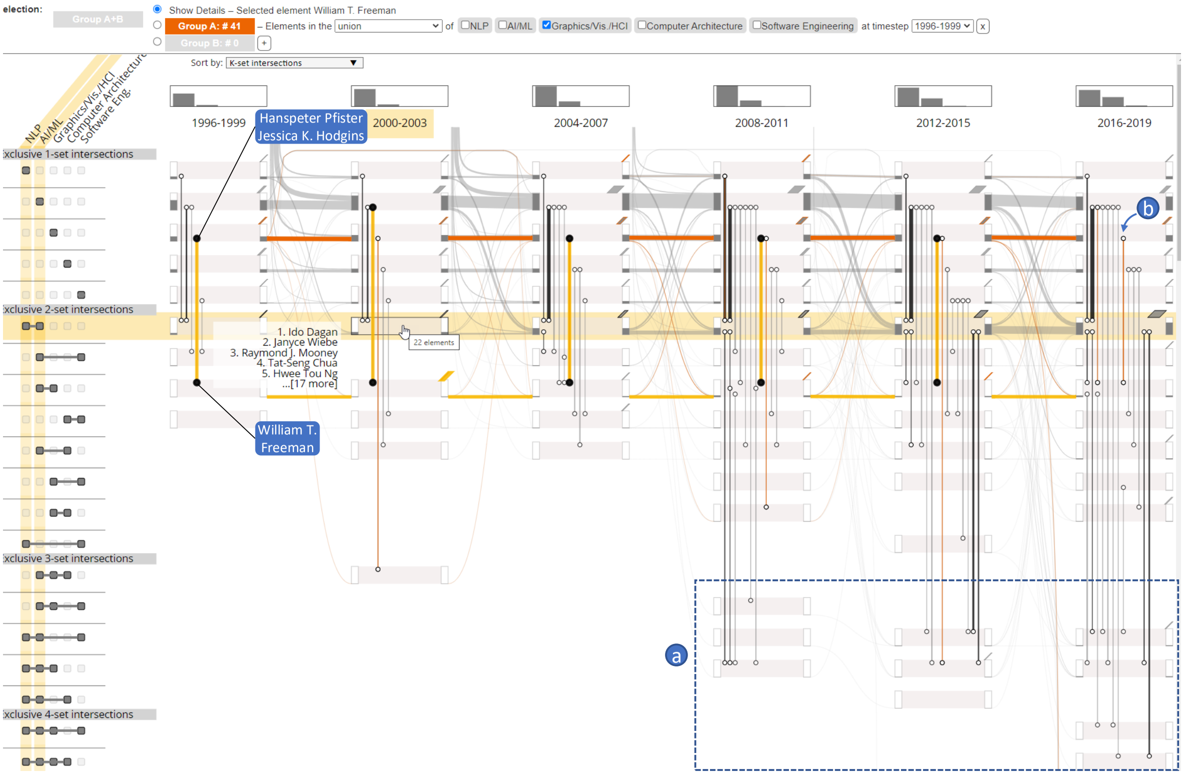

election:
Group A+B
Show Details – Selected element William T. Freeman
Group A: # 41
– Elements in the union of
☐NLP ☐AI/ML ☑Graphics/Vis./HCI ☐Computer Architecture ☐Software Engineering at timestep 1996-1999 x
Group B: # 0
+
Sort by: K-set intersections
NLP
AI/ML
Graphics/Vis./HCI
Computer Architecture
Software Eng.
xclusive 1-set intersections
xclusive 2-set intersections
xclusive 3-set intersections
xclusive 4-set intersections
1996-1999
2000-2003
2004-2007
2008-2011
2012-2015
2016-2019
Hanspeter Pfister
Jessica K. Hodgins
William T. Freeman
1. Ido Dagan
2. Janyce Wiebe
3. Raymond J. Mooney
4. Tat-Seng Chua
5. Hwee Tou Ng
...[17 more]
22 elements
a
b

Figure 8.2: The main view shows dynamic collaborations among researchers (elements) as interactions publishing in different fields of study (sets). The blue annotations show the names of involved researchers in an interaction. A researcher, *William T. Freeman*, is selected, which highlights his interactions with others in yellow-colored edges.

lines indicate that he consistently published in both research fields. The highlighted hyperedges mark his collaborations. Hovering over one such hyperedge in the first timestep reveals the names of collaborating authors (blue annotations in Figure 8.2). The interaction was between *Hanspeter Pfister* and *Jessica K. Hodgins* publishing exclusively in [*Graphics/Vis.HCI*] and *William T. Freeman* in the exclusive intersection of [*AI/ML*, *Graphics/Vis./HCI*]. It indicates that the researchers co-authored a paper that was published in the field of [*Graphics/Vis.HCI*]. Since *William T. Freeman* is in the exclusive intersection of the two fields, it also means that apart from this collaboration, in the same timestep: (a) he published at least one paper in *AI/ML* venue not co-authored with either of the two researchers and (b) the authors did not publish in any other research fields. Such interactions may indicate the diverse expertise of individual researchers or that in interdisciplinary projects, the required skill set in a different field of research is fulfilled by inviting experts from other fields (e.g., *William T. Freeman* is an expert in *AI/ML* who consistently contributed to the projects published in *Graphics/Vis.HCI* venue).

Growth in Collaborative Interactions. We observe that initially, there were no exclusive high-order intersections, but later, some researchers started publishing in multiple fields and interacted with others (Figure 8.2a). The width and the number of vertical lines in each timestep indicate that the number of interactions between researchers has been steadily increasing. Hovering over exclusive intersection [*NLP*, *AI/ML*] (Figure 8.2), we see the increasing width of emphasized vertical lines with exclusive intersection [*AI/ML*], suggesting a steady growth of collaborations between researchers publishing in the two fields. On the other hand, there are only a few vertical lines connecting rows involving *Computer Architecture*, suggesting minimal inter-disciplinary interactions with experienced researchers in this field.

Interactions of Early *Graphics/Vis./HCI* Researchers. To explore the interactions of early *Graphics/Vis./HCI* researchers, we specified a query to select all the elements that belonged to the set in the first timestep (Figure 8.2 top). The query returned 41 such researchers, shown in orange color. The orange-colored area in skewed rectangles on the nodes in the row and thin-colored vertical lines (e.g., Figure 8.2b) indicate that the researchers collaborated not only with others from the same community but also with those from other communities.

8.3 LIMITATIONS

The insights from the application examples indicate the value of analyzing element interactions along with the changing memberships in overlapping sets. We extended the design of Set Streams by embedding interactions of entities to show the two aspects of group dynamics, namely, the evolving memberships in dynamic overlapping groups and entity interactions (RO 1.1 and 2.2). However, the extension demonstrated only one possibility of a visualization design to show group behavior and did not systematically derive a design space for such extensions. In this section, we discuss the limitations of the approach and ideas for future work.

8.3.1 *Scalability*

The scalability is almost similar to Set Streams (~ 400 elements, ~ 7 sets, 6-7 timesteps, and ~120 interactions). Since the nodes are split and widened, fewer timesteps could be shown. Horizontal scrolling could partly help. The comparison between selected groups is preserved, as proposed in Set Streams (RO 1.2), but is limited to only two groups. Moreover, due to the increase in information (specifically due to interactions), as compared to Set Streams, the visual analysis becomes complex. The approach tackles this by abstracting and providing relevant details on demand, but other solutions may be explored. For instance, providing a visual summary of the interactions through short natural language text templates and inline graphics, e.g., as proposed in VIS Author Profiles [193] for individual researchers. Although the vertical lines do not overlap, the dense representation affects the legibility. Also, the number of intersections grows exponentially with more sets, affecting the approach's scalability. Both could be partly addressed by aggregation (e.g., one row for all 3-set intersections) or hiding the unimportant intersections.

8.3.2 *Generalizability*

As indicated by the two application examples, the design is generalizable to different scenarios where entities have dynamic multiple memberships in groups and interact with each other. It can be argued that an integrated representation of the two seemingly independent attributes makes the visualization complex. However, such visualization supports a joint analysis, especially in scenarios where the two behaviors are interdependent. Regarding limitations in terms of generalizability, similar to Set Streams, the design does not show the membership weight of an element in a set (RO 1.3). The limitation restricts the approach to do an in-depth analysis in certain scenarios where membership weight in a group is central, e.g., the market share of a company's product in a category or number of papers published by a researcher [194].

8.3.3 *Entity Interaction Attributes*

Although the proposed extension to Set Streams encodes interactions between entities, the design becomes complex, which leaves little room to accommodate other relevant details. For instance, in some scenarios, the location provides the spatial context to understand and interpret the interactions between entities (e.g., in Pommerman and Flatland environments). Embedding the spatial context while showing group membership has been found useful, e.g., in the static set visualization technique LineSets [195]. Moreover, in some scenarios, there may be a need to discern between the different types of entity interactions (e.g., acquisitions vs. partnerships). Since the proposed design extension shows aggregated interactions, it is unfeasible to differentiate between individual interactions. Additionally, the design is limited to showing interactions occurring at a specific timestep. Still, it is unable to encode those lasting several timesteps (e.g., signing partnership agreements between companies that have a fixed duration).

9 DISCUSSION AND CONCLUSION

In the previous chapters, we explored different visualizations to analyze the changing memberships of entities in groups (Chapter 3 and Chapter 4). Also, we surveyed visualizations to encode user behaviors in a dynamic scenario (Chapter 5) and proposed approaches to explore entity interactions in two dynamic environments (Chapter 6 and Chapter 7). Finally, we also presented an example to visualize both aspects of the group behavior (Chapter 8). In this chapter, we discuss other related challenges in visualizing the complex group dynamics and describe a few works in the initial stages of research as ideas for potential solutions (Section 9.1). Next, we summarize the work done, discuss limitations, and describe ideas for future work (Section 9.2). Finally, the chapter ends with a brief outlook on the topic (Section 9.3).

9.1 DISCUSSION

Interactions are a direct and explicit connection between the involved entities. Previous chapters proposed visualization approaches to encode such evolving interactions between entities belonging to groups. However, specific to a scenario, entity interactions may involve additional attributes, which present unique challenges and opportunities to understand complex group dynamics. For instance, evolving location-specific interactions between an entity and its environment or multimodal interactions between entities. Additionally, when the number and type of interactions increase, it becomes challenging to explore and convey the observed group behaviors. We discuss possible solutions as works in progress to address such challenges.

9.1.1 *Spatial Interactions of an Entity with its Environment*

In some scenarios, the location of an entity's interaction with its environment plays a crucial role in understanding its evolving behavior. It is particularly relevant in situations where the environment may be unfamiliar to an entity. Hence, it relies on exploring the environment through interactions, which affects its behavior over time.

Let us consider an example of an autonomous agent in a game environment—*Sonic the Hedgehog 2* [196]. Sonic is the in-game character usually controlled by a human player but, in our case, is an autonomous agent. The goal of each game level is to finish it by making Sonic run from left to right while avoiding obstacles such as enemies or traps (which might kill Sonic). The action space in the game is simple: At any point, Sonic can either run (left or right), roll, duck down, jump, or perform a spin dash move (roll with high speed). Most of the enemies can be killed by jumping on or rolling through them. While advancing through the level, Sonic can collect different power-ups. Most notable are the rings that accumulate and protect Sonic from dying.

In our work [197], we visualize the behavior of a trained autonomous agent that uses the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [198] to train the agent for playing the unfamiliar game levels. The algorithm implements a genetic approach to deep reinforcement learning where the structure of the neural network evolves over several generations. Each generation consists of 30 agents that play the game. The best-performing agents from a generation are selected for crossovers to make the next generation. A reward function determines the performance of each agent. Since the goal of the game is to reach the far right end of the level, a simple reward function was used based on the Euclidean distance between the position of the Sonic and the goal position.

Since the interactions with the environment affect the agent's performance, it becomes crucial to analyze them. As shown in Figure 9.1, the line chart at the top encodes the minimum (—), average (—), and maximum (—) reward values for the population of agents in each generation along the training process. Two maps of the selected level [199] are shown below the line chart. For each map, there is a dropdown menu to select a specific generation of agents. Showing many individual trajectories of agents would create clutter due to overlap and make it difficult to derive meaningful insights. Hence, trajectories are aggregated and encoded on top of the map. Characteristic points are calculated by analyzing all the trajectories of agents in the selected generation and shown as white circles. The width of a black line between two circles denotes the travel frequency of agents between the two points in either direction.

By examining generation 29 (Figure 9.1 b), we see that the agents were able to overcome initial obstacles and travel far in the level. The thick black line near the first waterfall (Figure 9.1 b1) indicates there was a lot of movement in this region. On a closer look, we see that the waterfall has some enemies, and the preceding black lines are thinner than the one placed above the waterfall. Knowing the game mechanics, we can infer that near the waterfall, some agents of generation 29 got hit by the enemy in the waterfall and got knocked back. The agents who learned to overcome this obstacle did not face trouble in crossing the second waterfall (Figure 9.1 b2). However, the agents did not interact with the power-ups, i.e., collecting the rings in the beginning, which could have avoided the agent's death when hit by the enemy.

In contrast to generation 29, some agents of generation 56 started to take the upper route of the level (Figure 9.1 c). More agents in generation 56 discovered the secret path in the level (Figure 9.1 c1). Although few agents in generation 29 learned to interact with the environment by performing the spin dash maneuver

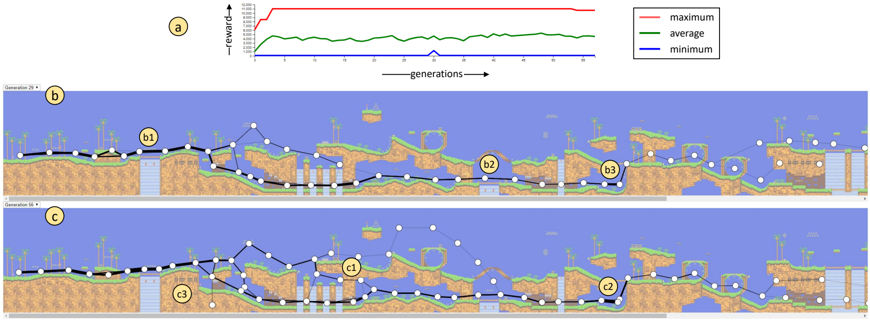


Figure 9.1: Visualizations showing (a) a line chart of reward values received by agents across the training process (generations), the aggregated trajectories of agents from (b) generation 29, and (c) generation 56 in *Emerald Hill Zone* level of the game *Sonic the Hedgehog 2*.

and crossing the steep vertical wall (Figure 9.1 b3), many agents in generation 56 were not able to get past the obstacle, as shown by the thick black line before the wall (Figure 9.1 c2).

9.1.2 *Simultaneous Activities and Multimodal Interactions*

Barthelmess et al. [200] highlight the complexity in analyzing collaborative human interactions due to multimodality: *"Participants of collaborative interactions speak, write, sketch and express themselves via gestures, facial expressions, and other body motions."* Simultaneous investigation of modalities and analysis of a sequence of interactions among entities together is challenging but essential to understanding complex collaborative behavior. The insights from such analysis are valuable as they might help to train the entities in scenarios where they need to collaborate in critical situations.

Consider an example of triaging patients in a medical emergency. It is a process through which the priority of treatment and transport to the hospital is determined in mass casualty incidents, e.g., road accidents involving multiple vehicles. Teams of medical professionals performing the triage, usually consisting of a leading doctor and a notetaking person, are usually trained in an environment simulating the event to improve their collaboration. Data from such training sessions are recorded for later analysis and contain metadata along with annotated multimodal interactions between the team members, e.g., conversations, eye-gaze, activities performed, documented decisions, etc. The timeline could be enriched to encode the data for investigation of the collaboration between team members.

In our work [201], we proposed a timeline visualization for the scenario. Figure 9.2 shows a cutout of the timeline. The rows encode the multimodal interactions as events and are grouped to represent actions relevant to the team (first

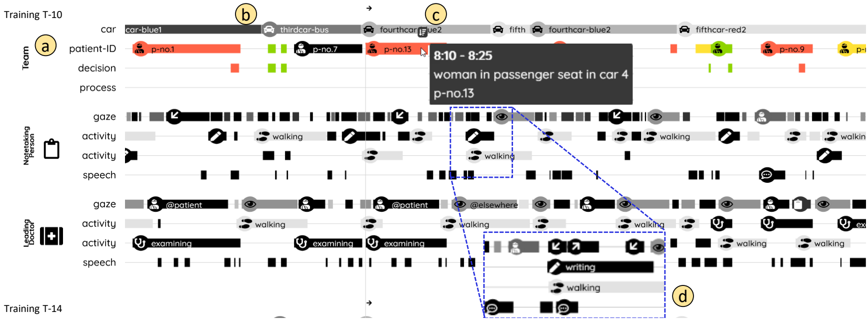


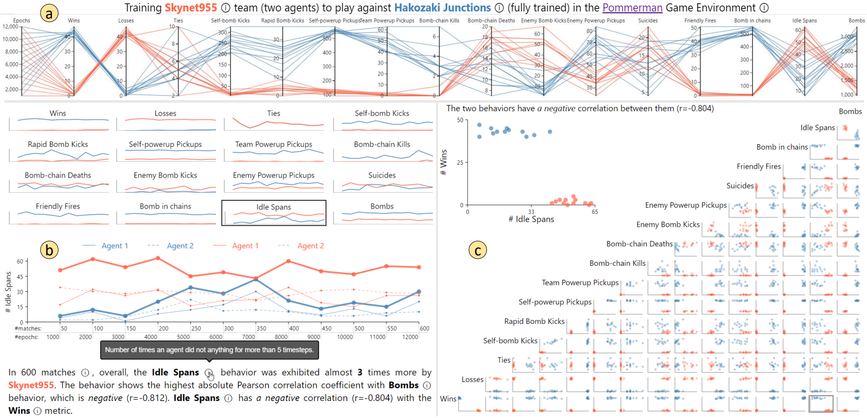
Figure 9.2: Timeline visualization of a session (top) with (a) tier labels, (b) visually encoded annotations on a vertical timescale with (c) annotation details inside a tooltip and a button for alignment, and (d) a close-up of a section in the timeline.

four rows) and the individual members (Figure 9.2a). The horizontal axis represents time, and interaction events are displayed as rectangles in the respective row. The start time and duration of the interaction are encoded through the rectangle's position and width (Figure 9.2b). Icons on the rows are used to discern between the actions (e.g., examining, walking, writing). A row in the timeline shows a patient's ground truth classification (*true triage category*) in the *patient-ID*'s annotations with respective triage colors ⬛ 🟧 🟨 🟩. Directly below, the team's triage decision is indicated by colored rectangles in the *decision* row, which supports comparing a triage decision to the ground truth.

Analyzing the figure, we infer that the notetaking person performed simultaneous activities, e.g., walking and documenting the decisions (Figure 9.2d). Since the writing activity came after the leading doctor stopped speaking, it is likely that the documentation did not influence the triage decision. The team exhibited the behavior several times. We can infer that the notetaking person probably uses the time between walking over to the patients in different vehicles to document the observations. During the patient examination, the notetaking person looks at the patient and the examining doctor, who is mainly responsible for the diagnosis. Hence, the notetaking person's behavior is unusual in collaborating with different roles in the scenario.

9.1.3 Exploranable *Visualizations for Complex Behaviors*

With the increasing complexity and scale of interactions among entities, it becomes increasingly challenging to enable visual analysis while simultaneously conveying the observed behaviors in a simplified way. As a solution, *exploranable* approaches are effective in striking a balance between the two aspects. For instance, Haris et al. [202] proposed an *exploranative* approach, integrating dynamic natural lan-



Training **Skynet955** ⓘ team (two agents) to play against **Hakozaki Junctions** ⓘ (fully trained) in the Pommerman Game Environment ⓘ

The two behaviors have *a negative* correlation between them (r=-0.804)

Number of times an agent did not anything for more than 5 timesteps.

In 600 matches ⓘ, overall, the **Idle Spans** ⓘ behavior was exhibited almost **3** times more by **Skynet955**. The behavior shows the highest absolute Pearson correlation coefficient with **Bombs** ⓘ behavior, which is *negative* (r=-0.812). **Idle Spans** ⓘ has *a negative* correlation (r=-0.804) with the **Wins** ⓘ metric.

Figure 9.3: Visualizing behavior and game statistic metrics of a training: (a) An overview of all metrics, (b) evolution of individual metrics, and (c) correlation among metrics.

guage text and interactive visualizations to analyze the code quality based on various computed metrics (e.g., code coupling, cohesion). In our context, using such approaches can enable users to explore complex behaviors and analyze different types of evolving interactions while explaining the observed behaviors with simplified and integrated representations.

Let's take an example of a team of two AI agents competing against another team in Pommerman (Section 6.1), a bomb-laying game environment where opponents battle in an 11×11 grid map. An agent can move, drop a bomb (which explodes after 10 timesteps), and uncover hidden power-ups by blasting wooden walls. Available power-ups allow agents to increase the blast radius, kick bombs, and drop bombs without waiting for the previous one to explode. Bomb chains can be formed if the blast flame engulfs other bombs, making them explode instantly.

In our work [203], we trained a team of two agents *Skynet955* [204] (S955) from scratch using neural network and reinforcement learning. After every 1000 training epochs, we recorded 50 matches against the top-performing team *Hakozaki Junctions* [205] (HJ), through 12 intervals (until 12000 epochs). We defined and quantified 13 behaviors and interactions from the recordings, e.g., *Enemy Powerup Pickups*, which is defined as the number of times an agent grabbed a power-up that was uncovered by an opponent. In each interval, the values of game results per team (*Wins*, *Losses*, and *Ties*) and quantified behaviors are summed across all 50 matches.

To facilitate exploration of the behaviors, the approach was developed with linked views with standard visualizations. At the top in Figure 9.3a, the parallel coordinates plot provides an overview and compares the behaviors of the two teams, S955 and HJ. It reveals clear differences between the teams in the frequency of some behaviors (e.g., *Self-powerup Pickups*, *Bombs*) or similar frequencies in other behaviors (e.g., *Suicides*, *Idle Spans*).

The small-sized line charts (Figure 9.3b) offer a comparative view of teams' evolving behavior frequencies along training epochs. Each small line chart can be clicked to see an enlarged version below. Both agents of a team are shown as different lines (dotted and solid). The adaptable caption of the plot summarizes the frequency, explains the behavior that was highly correlated with the selected behavior, and mentions the correlation with *Wins*. For instance, the selection of *Idle Spans* shows that S995 exhibits this behavior almost three times more than that of HJ, but with a much smaller difference in the middle of the training and at the end. Also, the behavior was negatively correlated with dropping *Bombs* ($r = -0.812$). And finally, a scatterplot matrix on the right (Figure 9.3c) reveals statistical relationships between all behaviors and game results.

The approach integrates interactions in the linked visualizations, typically used for exploration in visual analytics, with the dynamically generated text explaining the observed insights based on the current selection. Although the example provides just a glimpse and has limited features, the direction can be pursued further to propose *exploranable* visualizations for complex behaviors.

9.2 CONCLUSION, LIMITATIONS, AND FUTURE WORK

Based on the insights from the application examples, feedback, and experience, we discuss the limitations of the proposed approaches in (a) visualizing the dynamic memberships of entities in groups, (b) encoding enriched interactions, and (c) exploring group dynamics at scale.

9.2.1 *Dynamic Entity Memberships in Groups*

The research on dynamic set visualizations is surprisingly rare. As a result, we have a limited understanding of the analysis tasks in the specific context of exploring temporal set-typed data, which negatively impacts bench-marking a new technique or supporting the design process of a novel approach. Moreover, previous chapters proposed a few techniques that can represent up to 7 sets, <500 elements, and 10 timesteps. Although challenging, to adequately model and visually analyze complex processes (e.g., training of multi-class classifiers to detect multiple objects in an image), we need dynamic set visualizations with better scalability in terms of representing more sets, elements, set overlaps, and timesteps.

To accommodate a high number of elements, several set visualizations use aggregation to encode their quantity (e.g., UpSet [24], PowerSet [118]). However, aggregation limits the capability to focus on the temporal analysis of individual elements, which remains a challenge for dynamic set visualization. In our technique, we encoded each element with a circle and the set membership weight with the size of the circle, which enabled deeper comparison with other elements (e.g., differences among the top contributors of all modules in a software repository) [194] (Chapter 4). In Set Streams [104], the challenge was partially addressed by highlighting the streams of the selected element (Chapter 3). However, further research is required to represent the individual elements while reducing the clutter and visual complexity in the encodings.

While graphs model a specific type of relationship between different vertices as edges, multilayer graphs are used for multiple pairwise relations. Since visualizing such multilayer graphs is not a straightforward extension of the known single-layer graphs, various novel techniques have been proposed (e.g., see survey [39, 206]). Similarly, we need to extend dynamic sets to better model the multiple relations between entities other than the one defining the element-set memberships. We proposed a technique that encodes the rare interactions, an alternate temporary relation between entities, as vertical lines [207] (Chapter 8). However, the encoding is not scalable to represent scenarios with a high number of entity interactions. Moreover, it is unable to discern between the different types of entity interactions (e.g., company partnerships vs. acquisitions). A possible extension addressing these limitations can be used to model and visualize, e.g., the evolving citations of a research article in different overlapping categories ([208]) together with relations between the citing articles (e.g., similarity based on research themes, keywords, citations, or co-authorship).

9.2.2 *Visualizing Entity Interaction Details*

In our work, we explored interactions of collaborating entities who were remotely located but were placed in front of each other in a mixed reality environment [209] (Chapter 5). While the interactions involved putting the virtual and real puzzle pieces in the correct positions, the entities also communicated with each other through an audio channel. Although we represented the communication through a waveform visualization, the encoding was limited to showing the temporal density of the interaction. Furthermore, interactions could involve both audio and video channels between entities, e.g., looking at objects of interest during conversation. For such modalities, the details of the communication interaction could be appropriately visualized. For instance, the content of the verbal communication through embedded word clouds in the timeline and glyphs to convey the non-verbal visual features, such as gestures and facial expressions, etc. Similarly, details of the tactile interactions, where entities touch the same objects simultaneously (e.g., while assembling parts of a machine), could be contextualized using the relevant features of the object (e.g., orientation).

With efficient communication between AI agents, they are expected to perform better in collaborative tasks. In Pommerman, the AI agents exhibited primitive and unstructured communication between the teammates, which was encoded as a histogram along the horizontal timeline [210] (Chapter 6). With further advances, the interactions between AI agents could include more details, e.g., they could communicate by first establishing mutually agreed protocols to exchange messages, clarifying their intent, etc. The interaction may also require them to share other details, such as the internal state of the agent, for mutual benefit, e.g., to synchronize their worldviews while exploring the environment [52]. Although such interaction details are specific to an environment, their integration into the appropriate visualizations or novel techniques could be explored in the future.

In Chapter 7, a visualization with linked views for spatio-temporal attributes was proposed to represent the scheduling behavior of AI agents on fixed tracks. The horizontal timeline encoded important actions and events of each agent in a row, while the deadlock among agents was shown through lines connecting the respective rows. Although the information density was high, the visualization did not show other potentially relevant information, such as agent observations conveying whether the tracks ahead of an agent are free or occupied. These details could help understand the context of agent decisions. Hence, further research is required to present such details during the visual analysis, maybe by abstracting them in the default view of the visualization and presenting them on-demand or through alternate representations that could better handle the information density.

9.2.3 *Exploring Group Dynamics at Scale*

Taking the field of multi-agent research as an example, we discuss the topic of group dynamics at scale. The goal is to train autonomous agents that can collaborate to perform multiple tasks in diverse environments. The trend is towards building virtual simulations where multiple agents train to interact with each other in diverse environments. Contributing to the goal, we proposed a visualization to explore the collaborative and competitive group dynamics among four agents in the Pommerman environment, a small grid-based world with 11×11 tiles (Chapter 6). Later, we analyzed the collaborative scheduling behavior of up to 90 agents in a larger environment, Flatland (Chapter 7). While these environments have helped AI researchers to uncover exhibited behavior, even larger testbed environments have been proposed to study the emergent group behavior at scale, e.g., Neural MMO [211]. The proposed encodings for individual interactions, movements, and actions of each entity would not be feasible to understand the group dynamics at scale. Modeling interactions as events, we can take inspiration from aggregation-based event sequence visualizations for solutions that can better handle the scale (e.g., DecisionFlow [212]). Alternatively, some approaches have proposed group tracking models with visual exploration for large datasets. For instance, Ozer et al. [213] propose interactive clustering and isosurface visualization to study the groups of features in time-varying 3D fluid-flow simulations. Unlike our approach, they handle scale by not showing the individual entities but rather focusing on clustering and visualizing the entity features that drive the group dynamics.

To understand the local behavior in group dynamics, analyzing the context through short sequences of entity interactions, events, and actions is valuable.

Searching and highlighting the occurrences of such sequences can help to understand the specific local behaviors involving multiple entities. To help discover unknown sequences, pattern mining algorithms [214] can be used to identify common behaviors. With larger environments involving many entities, investigating the context around all occurrences of a specific interaction becomes challenging, e.g., analyzing the most common things that happened just before and after kicking a bomb in *Pommerman*. To ease the analysis, an abstracted double tree representation can be used, which positions the analyzed interaction in the center while the sequence of other interactions, events, or actions are placed as connected links on the left (preceding sequence) and on the right (succeeding sequence) [3].

Finally, while the proposed techniques showing the group dynamics encode the exhibited interactions, they do not convey the reasoning of decisions. In the context of human entities, the relevant data could be captured, e.g., via establishing a think-aloud protocol in the scenario. Based on the data type, relevant representations could be embedded, e.g., word cloud for text. For autonomous agents, the proposed techniques for explaining AI could be used to explain the agent's decision (e.g., [215, 216, 217]). While integrating the relevant representation for decisions of each agent would clutter the main view, they can be provided on-demand to at least partly address the challenge of group dynamics at scale.

9.3 OUTLOOK

The core idea of the thesis is to explore group dynamics from two perspectives. The first perspective aims to analyze the changes in memberships of entities in overlapping groups. And second, to understand the evolving interactions among entities, both within a group and between different groups. As demonstrated through several examples, visualizations could be helpful in the exploration of entity behaviors, e.g., coordination between entities.

I believe that the exploratory analysis of group dynamics is crucial to understand the dynamic processes in a complex system involving several entities. The demonstrated examples show complex processes in scenarios involving multiple humans (e.g., collaborative tasks, such as triaging patients) or autonomous agents (e.g., runtime planning of path towards destination). However, other scenarios beyond analyzing small-scale social interactions or a few hundred autonomous agents would also benefit from such visual analysis. Due to the limited space, it is generally challenging for visualizations to handle the scale and represent multiple attributes in the group behavior. However, the availability of datasets from simulated environments (e.g., in AI) to study group dynamics has improved, and visualization research has matured enough to tackle these challenges head-on.

BIBLIOGRAPHY

[1] **Shivam Agarwal**, Jürgen Bernard, and Fabian Beck. "Computer-supported interactive assignment of keywords for literature collections." In: *IEEE VIS Workshop on Machine Learning from User Interaction for Visualization and Analytics*. 2018. URL: https://learningfromusersworkshop.github.io/papers/Literature.pdf.

[2] **Shivam Agarwal**, Shahid Latif, and Fabian Beck. "How visualization PhD students cope with paper rejections." In: *IEEE VIS workshop on Celebrating the Scientific Value of Failure (FailFest)*. 2020, pp. 6–10. DOI: 10.1109/FailFest51498.2020.00006.

[3] Cedric Krause, **Shivam Agarwal**, Michael Burch, and Fabian Beck. "Visually abstracting event sequences as double trees enriched with category-based comparison." In: *Computer Graphics Forum* 42.6 (2023). DOI: 10.1111/cgf.14805.

[4] Cedric Krause, **Shivam Agarwal**, Mohammad Ghoniem, and Fabian Beck. "Visual comparison of multi-label classification results." In: *Vision, Modeling, and Visualization*. 2021. DOI: 10.2312/vmv.20211367.

[5] Shahid Latif, **Shivam Agarwal**, Simon Gottschalk, Carina Chrosch, Felix Feit, Johannes Jahn, Tobias Braun, Yannick Christian Tchenko, Elena Demidova, and Fabian Beck. "Visually connecting historical figures through event knowledge graphs." In: *IEEE VIS*. 2021, pp. 156–160. DOI: 10.1109/VIS49827.2021.9623313.

[6] Donelson R Forsyth. *Group dynamics*. Cengage Learning, 2018. ISBN: 9781285531854.

[7] John C. Turner. "Towards a cognitive redefinition of the social group." In: *Social identity and intergroup behavior*. Cambridge University Press, 1982, pp. 15–40.

[8] Carol C. Gould. *Globalizing democracy and human rights*. 2004. ISBN: 9780511617096. DOI: 10.1017/CBO9780511617096.

[9] Joann Keyton. *Communicating in groups: building relationships for effective decision making*. McGraw-Hill, 2002. ISBN: 9780195175165.

[10] James Flamino, Boleslaw K Szymanski, Ashwin Bahulkar, Kevin Chan, and Omar Lizardo. "Creation, evolution, and dissolution of social groups." In: *Scientific Reports* 11.1 (2021), p. 17470. DOI: 10.1038/s41598-021-96805-7.

[11] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. "Visual methods for analyzing time-oriented data." In: *IEEE Transactions on Visualization and Computer Graphics* 14.1 (2008), pp. 47–60. ISSN: 1941-0506. DOI: 10.1109/TVCG.2007.70415.

[12] Urška Demšar, Kevin Buchin, Francesca Cagnacci, Kamran Safi, Bettina Speckmann, Nico Van de Weghe, Daniel Weiskopf, and Robert Weibel. "Analysis and visualisation of movement: an interdisciplinary review." In: *Movement ecology* 3.1 (2015), pp. 1–24. DOI: 10.1186/s40462-015-0032-y.

[13] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. "Exploratory spatio-temporal visualization: an analytical review." In: *Journal of Visual Languages & Computing* 14.6 (2003). Visual Data Mining, pp. 503–541. ISSN: 1045-926X. DOI: 10.1016/S1045-926X(03)00046-6.

[14] Gennady Andrienko, Natalia Andrienko, Urska Demsar, Doris Dransch, Jason Dykes, Sara Irina Fabrikant, Mikael Jern, Menno-Jan Kraak, Heidrun Schumann, and Christian Tominski. "Space, time and visual analytics." In: *International Journal of Geographical Information Science* 24.10 (2010), pp. 1577–1600. DOI: 10.1080/13658816.2010.508043.

[15] Chen Zhong, Tao Wang, Wei Zeng, and Stefan Müller Arisona. "Spatiotemporal visualisation: a survey and outlook." In: *Digital Urban Modeling and Simulation*. 2012, pp. 299–317. ISBN: 978-3-642-29758-8. DOI: 10.1007/978-3-642-29758-8_16.

[16] Yi Guo, Shunan Guo, Zhuochen Jin, Smiti Kaul, David Gotz, and Nan Cao. "Survey on visual analysis of event sequence data." In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (2022), pp. 5091–5112. ISSN: 1941-0506. DOI: 10.1109/TVCG.2021.3100413.

[17] Cinjon Resnick, Wes Eldridge, David Ha, Denny Britz, Jakob Foerster, Julian Togelius, Kyunghyun Cho, and Joan Bruna. *Pommerman: a multi-agent playground*. 2018. arXiv: 1809.07124.

[18] Sharada Mohanty et al. "Flatland-RL: multi-agent reinforcement learning on trains." In: *arXiv* (2020). arxiv.org/abs/2012.05893. eprint: 2012.05893 (cs.AI).

[19] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. "The state-of-the-art of set visualization." In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: 10.1111/cgf.12722.

[20] Sara Irina Fabrikant, Silvia Miksch, and Alexander Wolff. "Visual analytics for sets over time and space." In: *Dagstuhl Reports* 9.5 (2019), pp. 31–57. ISSN: 2192-5283. DOI: 10.4230/DagRep.9.5.31.

[21] Maximilian T. Fischer, Alexander Frings, Daniel A. Keim, and Daniel Seebacher. "Towards a survey on static and dynamic hypergraph visualizations." In: *IEEE Visualization Conference (VIS)*. 2021, pp. 81–85. DOI: 10.1109/VIS49827.2021.9623305.

[22] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. "NodeTrix: a hybrid visualization of social networks." In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1302–1309. ISSN: 1941-0506. DOI: 10.1109/TVCG.2007.70582.

[23] Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. "Visualizing group structures in graphs: a survey." In: *Computer Graphics Forum* 36.6 (2017), pp. 201–225. DOI: 10.1111/cgf.12872.

[24] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. "UpSet: visualization of intersecting sets." In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1983–1992. DOI: 10.1109/TVCG.2014.2346248.

[25] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. "Design study of LineSets, a novel set visualization technique." In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2259–2267. ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.186.

[26] Rodrigo Santamaría and Roberto Therón. "Overlapping clustered graphs: co-authorship networks visualization." In: *Smart Graphics*. 2008, pp. 190–199. ISBN: 978-3-540-85412-8. DOI: 10.1007/978-3-540-85412-8_17.

[27] Corinna Vehlow, Fabian Beck, Patrick Auwärter, and Daniel Weiskopf. "Visualizing the evolution of communities in dynamic graphs." In: *Computer Graphics Forum* 34.1 (2015), pp. 277–288. DOI: 10.1111/cgf.12512.

[28] Michael Burch, Tanja Munz, Fabian Beck, and Daniel Weiskopf. "Visualizing work processes in software engineering with Developer Rivers." In: *Proceedings of the 3rd IEEE Working Conference on Software Visualization*. 2015, pp. 116–124. DOI: 10.1109/VISSOFT.2015.7332421.

[29] Petter Holme and Jari Saramäki. "Temporal networks." In: *Physics Reports* 519.3 (2012), pp. 97–125. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2012.03.001.

[30] Caroline S. Wagner and Loet Leydesdorff. "Network structure, self-organization, and the growth of international collaboration in science." In: *Research Policy* 34.10 (2005), pp. 1608–1618. ISSN: 0048-7333. DOI: 10.1016/j.respol.2005.08.002.

[31] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. "Patterns and dynamics of users' behavior and interaction: network analysis of an online community." In: *Journal of the American Society for Information Science and Technology* 60.5 (2009), pp. 911–932. DOI: 10.1002/asi.21015.

[32] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. "Quantifying social group evolution." In: *Nature* 446.7136 (2007), pp. 664–667. DOI: 10.1038/nature05670.

[33] Albert-Laszlo Barabâsi, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. "Evolution of the social network of scientific collaborations." In: *Physica A: Statistical Mechanics and its Applications* 311.3 (2002), pp. 590–614. ISSN: 0378-4371. DOI: 10.1016/S0378-4371(02)00736-7.

[34] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. "A taxonomy and survey of dynamic graph visualization." In: *Computer Graphics Forum* 36.1 (2017), pp. 133–159. DOI: 10.1111/cgf.12791.

[35] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. "GraphDiaries: animated transitions and temporal navigation for dynamic networks." In: *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), pp. 740–754. ISSN: 1941-0506. DOI: 10.1109/TVCG.2013.254.

[36] Ilya Boyandin, Enrico Bertini, and Denis Lalanne. "A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples." In: *Computer Graphics Forum* 31.3pt2 (2012), pp. 1005–1014. DOI: 10.1111/j.1467-8659.2012.03093.x.

[37] Jin Xu, Yubo Tao, Yuyu Yan, and Hai Lin. "Exploring evolution of dynamic networks via diachronic node embeddings." In: *IEEE Transactions on Visualization and Computer Graphics* 26.7 (2020), pp. 2387–2402. DOI: 10.1109/tvcg.2018.2887230.

[38] Tatiana von Landesberger, Sebastian Bremm, Natalia Andrienko, Gennady Andrienko, and Maria Tekusova. "Visual analytics methods for categoric spatio-temporal data." In: *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology*. 2012, pp. 183–192. DOI: 10.1109/VAST.2012.6400553.

[39] Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoit Otjacques, and Bruno Pinaud. "The state of the art in multilayer network visualization." In: *Computer Graphics Forum* (2019). ISSN: 1467-8659. DOI: 10.1111/cgf.13610.

[40] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. "ThemeRiver: visualizing thematic changes in large document collections." In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 9–20. ISSN: 1077-2626. DOI: 10.1109/2945.981848.

[41] Michael Ogawa and Kwan L. Ma. "Software evolution storylines." In: *Proceedings of the 5th International Symposium on Software Visualization*. 2010, pp. 35–42. ISBN: 978-1-4503-0028-5. DOI: 10.1145/1879211.1879219.

[42] Florian Heimerl, Qi Han, Steffen Koch, and Thomas Ertl. "CiteRivers: visual analytics of citation patterns." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 190–199. DOI: 10.1109/TVCG.2015.2467621.

[43] Jessica Perrie, Jing Xie, Maleknaz Nayebi, Marios Fokaefs, Kelly Lyons, and Eleni Stroulia. "City on the River: visualizing temporal collaboration." In: *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*. 2019, pp. 82–91.

[44] Lee Byron and Martin Wattenberg. "Stacked graphs – geometry & aesthetics." In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1245–1252. DOI: 10.1109/TVCG.2008.166.

[45] Phong H Nguyen, Kai Xu, Rick Walker, and BL William Wong. "TimeSets: timeline visualization with set relations." In: *Information Visualization* 15.3 (2016), pp. 253–269. DOI: 10.1177/1473871615605347.

[46] Paola Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud, and Jean-Daniel Fekete. "Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization." In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/TVCG.2019.2933196.

[47] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. "Bubble Sets: revealing set relations with isocontours over existing visualizations." In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1009–1016. DOI: 10.1109/TVCG.2009.122.

[48] Vanessa Peña-Araya, Tong Xue, Emmanuel Pietriga, Laurent Amsaleg, and Anastasia Bezerianos. "HyperStorylines: interactively untangling dynamic hypergraphs." In: *Information Visualization* 21.1 (2021), pp. 38–62. DOI: 10.1177/14738716211045007.

[49] Brigitte Jordan and Austin Henderson. "Interaction analysis: foundations and practice." In: *Journal of the Learning Sciences* 4.1 (1995), pp. 39–103. DOI: 10.1207/s15327809jls0401_2.

[50] Mikkel R. Jakobsen and Kasper Hornbæk. "Up close and personal: collaborative work on a high-resolution multitouch wall display." In: *ACM Transactions on Computer-Human Interaction* 21.2 (2014), pp. 1–34. DOI: 10.1145/2576099.

[51] John C. Tang. "Findings from observational studies of collaborative work." In: *International Journal of Man-Machine Studies* 34.2 (1991), pp. 143–160. DOI: 10.1016/0020-7373(91)90039-a.

[52] Suyun Sandra Bae, Federico Rossi, Joshua Vander Hook, Scott Davidoff, and Kwan-Liu Ma. "A visual analytics approach to debugging cooperative, autonomous multi-robot systems' worldviews." In: *IEEE Conference on Visual Analytics Science and Technology*. 2020, pp. 24–35. DOI: 10.1109/VAST50239.2020.00008.

[53] Michael Nebeling et al. "MRAT: the mixed reality analytics toolkit." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. 2020, pp. 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376330.

[54] Tanja Blascheck, Markus John, Kuno Kurzhals, Steffen Koch, and Thomas Ertl. "VA^2: a visual analytics approach for evaluating visual analytics applications." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 61–70. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2467871.

[55] Roberto Minelli, Andrea Mocci, Mario Lanza, and Lorenzo Baracchi. "Visualizing developer interactions." In: *In Proceedings of the Second IEEE Working Conference on Software Visualization*. VISSOFT. 2014, pp. 147–156. DOI: 10.1109/VISSOFT.2014.31.

[56] Roberto Minelli, Andrea Mocci, and Michele Lanza. "I know what you did last summer: an investigation of how developers spend their time." In: *Proceedings of the 23rd IEEE International Conference on Program Comprehension*. ICPC. 2015, pp. 25–35. DOI: 10.1109/ICPC.2015.12.

[57] YoungSeok Yoon, Brad A Myers, and Sebon Koo. "Visualization of fine-grained code change history." In: *Proceedings of the IEEE Symposium on Visual Languages and Human Centric Computing*. VLHCC. 2013, pp. 119–126. DOI: 10.1109/VLHCC.2013.6645254.

[58] Anthony Tang, Michel Pahud, Sheelagh Carpendale, and Bill Buxton. "VisTACO: visualizing tabletop collaboration." In: *ACM International Conference on Interactive Tabletops and Surfaces*. 2010. DOI: 10.1145/1936652.1936659.

[59] Petra Isenberg, Danyel Fisher, Sharoda A. Paul, Meredith Ringel Morris, Kori Inkpen, and Mary Czerwinski. "Co-located collaborative visual analytics around a tabletop display." In: *IEEE Transactions on Visualization and Computer Graphics* 18.5 (2012), pp. 689–702. ISSN: 1941-0506. DOI: 10.1109/TVCG.2011.287.

[60] Zhiwen Yu, Zhiyong Yu, Hideki Aoyama, Motoyuki Ozeki, and Yuichi Nakamura. "Capture, recognition, and visualization of human semantic interactions in meetings." In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*. 2010, pp. 107–115. DOI: 10.1109/PERCOM.2010.5466987.

[61] Sebastian Hubenschmid, Jonathan Wieland, Daniel Immanuel Fink, Andrea Batch, Johannes Zagermann, Niklas Elmqvist, and Harald Reiterer. "ReLive: bridging in-situ and ex-situ visual analytics for analyzing mixed reality user studies." In: *CHI Conference on Human Factors in Computing Systems*. 2022. DOI: 10.1145/3491102.3517550.

[62] Günter Wallner and Simone Kriglstein. "Visualization-based analysis of gameplay data – a review of literature." In: *Entertainment Computing* 4.3 (2013), pp. 143–155. DOI: 10.1016/j.entcom.2013.02.002.

[63] Jiang Wu, Dongyu Liu, Ziyang Guo, Qingyang Xu, and Yingcai Wu. "TacticFlow: visual analytics of ever-changing tactics in racket sports." In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 835–845. ISSN: 1941-0506. DOI: 10.1109/TVCG.2021.3114832.

[64] Jiang Wu, Dongyu Liu, Ziyang Guo, and Yingcai Wu. "RASIPAM: interactive pattern mining of multivariate event sequences in racket sports." In: *IEEE Transactions on Visualization and Computer Graphics* 29.1 (2023), pp. 940–950. ISSN: 1941-0506. DOI: 10.1109/TVCG.2022.3209452.

[65] Ji Lan, Jiachen Wang, Xinhuan Shu, Zheng Zhou, Hui Zhang, and Yingcai Wu. "RallyComparator: visual comparison of the multivariate and spatial stroke sequence in table tennis rally." In: *Journal of Visualization* 25.1 (2022), pp. 143–158. DOI: 10.1007/s12650-021-00772-0.

[66] Günter Wallner. "Sequential analysis of player behavior." In: *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. CHI Play. 2015, pp. 349–358. DOI: 10.1145/2793107.2793112.

[67] Xing Mu, Ke Xu, Qing Chen, Fan Du, Yun Wang, and Huamin Qu. "MOOCad: visual analysis of anomalous learning activities in massive open online courses." In: *EuroVis 2019 - Short Papers*. The Eurographics Association, 2019. ISBN: 978-3-03868-090-1. DOI: 10.2312/evs.20191176.

[68] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. "LifeFlow: visualizing an overview of event sequences." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. 2011, pp. 1747–1756. ISBN: 9781450302289. DOI: 10.1145/1978942.1979196.

[69] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. "LifeLines: visualizing personal histories." In: *Readings in Information Visualization: Using Vision to Think*. 1999, pp. 287–294. ISBN: 1558605339. DOI: 10.1145/238386.238493.

[70] Mennatallah El-Assady, Valentin Gold, Carmela Acevedo, Christopher Collins, and Daniel Keim. "ConToVi: multi-party conversation exploration using topic-space views." In: *Computer Graphics Forum*. CGF 35.3 (2016), pp. 431–440. DOI: 10.1111/cgf.12919.

[71] Ben Rydal Shapiro and Rogers P. Hall. "Interaction geography in a museum." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI. 2017, pp. 2076–2083. ISBN: 9781450346566. DOI: 10.1145/3027063.3053146.

[72] Roshanak Zilouchian Moghaddam and Brian Bailey. "VICPAM: a visualization tool for examining interaction data in multiple display environments." In: *Lecture Notes in Computer Science – Human Interface and the Management of Information. Interacting with Information*. 2011, pp. 278–287. DOI: 10.1007/978-3-642-21793-7_32.

[73] Natalia Andrienko and Gennady Andrienko. "Visual analytics of movement: an overview of methods, tools and procedures." In: *Information Visualization* 12.1 (2013), pp. 3–24. DOI: 10.1177/1473871612457601.

[74] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. "Supporting visual exploration of object movement." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI. Palermo, Italy, 2000, pp. 217–220. ISBN: 1-58113-252-2. DOI: 10.1145/345513.345319.

[75] Mei-Po Kwan. "Interactive geovisualization of activity-travel patterns using three-dimensional geographical information systems: a methodological exploration with a large data set." In: *Transportation Research Part C: Emerging Technologies* 8.1 (2000), pp. 185–203. ISSN: 0968-090X. DOI: 10.1016/S0968-090X(00)00017-6.

[76] Menno-Jan Kraak. "The space-time cube revisited from a geovisualization perspective." English. In: *Proceedings of the 21st International Cartographic Conference*. 2003, pp. 1988–1996. ISBN: 0-958-46093-0. URL: https://www.icaci.org/files/documents/ICC_proceedings/ICC2003/Papers/255.pdf.

[77] Colin Ware, Roland Arsenault, Matthew Plumlee, and David Wiley. "Visualizing the underwater behavior of humpback whales." In: *IEEE Computer Graphics and Applications* 26.4 (2006), pp. 14–18. ISSN: 0272-1716. DOI: 10.1109/MCG.2006.93.

[78] Hanqi Guo, Zuchao Wang, Bowen Yu, Huijing Zhao, and Xiaoru Yuan. "TripVista: triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection." In: *Proceedings of the 2011 IEEE Pacific Visualization Symposium*. PacificVis. 2011, pp. 163–170. DOI: 10.1109/PACIFICVIS.2011.5742386.

[79] Michael Wörner and Thomas Ertl. "Visual analysis of public transport vehicle movement." In: *Proceedings of International Workshop on Visual Analytics*. EuroVA. The Eurographics Association, 2012. ISBN: 978-3-905673-89-0. DOI: 10.2312/PE.EuroVAST.EuroVA12.079-083.

[80] Sebastian Bremm, Gennady Andrienko, Natalia Andrienko, Tobias Schreck, and Tatiana von Landesberger. "Interactive analysis of object group changes over time." In: *Proceedings of International Workshop on Visual Analytics*. EuroVA. The Eurographics Association, 2011. ISBN: 978-3-905673-82-1. DOI: 10.2312/PE/EuroVAST/EuroVA11/041-044.

[81] Jinfeng Zhao, Pip Forer, and Andrew S Harvey. "Activities, ringmaps and geovisualization of large human movement fields." In: *Information Visualization* 7.3-4 (2008), pp. 198–209. DOI: 10.1057/PALGRAVE.IVS.9500184.

[82] Hazarath Munaga, Lucio Ieronutti, and Luca Chittaro. "CAST: a novel trajectory clustering and visualization tool for spatio-temporal data." In: *Proceedings of the 1st International Conference on Intelligent Human Computer Interaction*. 2009, pp. 169–175. DOI: 10.1007/978-81-8489-203-1_15.

[83] Gennady Andrienko, Natalia Andrienko, and Marco Heurich. "An event-based conceptual model for context-aware movement analysis." In: *International Journal of Geographical Information Science* 25.9 (2011), pp. 1347–1370. DOI: 10.1080/13658816.2011.556120.

[84] Brian Tomaszewski and Alan M MacEachren. "Geo-historical context support for information foraging and sensemaking: conceptual model, implementation, and assessment." In: *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*. VAST. 2010, pp. 139–146. DOI: 10.1109/VAST.2010.5652895.

[85] Christopher Andrews and Chris North. "The impact of physical navigation on spatial organization for sensemaking." In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2207–2216. DOI: 10.1109/TVCG.2013.205.

[86] Ulrich Von Zadow and Raimund Dachselt. "GIAnT: visualizing group interaction at large wall displays." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017. DOI: 10.1145/3025453.3026006.

[87] Sven Mayer, Lars Lischke, Jens Emil Grønbæk, Zhanna Sarsenbayeva, Jonas Vogelsang, Paweł W. Woźniak, Niels Henze, and Giulio Jacucci. "Pac-Many: movement behavior when playing collaborative and competitive games on large displays." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018. DOI: 10.1145/3173574.3174113.

[88] Alec Azad, Jaime Ruiz, Daniel Vogel, Mark Hancock, and Edward Lank. "Territoriality and behaviour on and around large vertical publicly-shared displays." In: *Proceedings of the Designing Interactive Systems Conference*. 2012, pp. 468–477. ISBN: 9781450312103. DOI: 10.1145/2317956.2318025.

[89] Wolfgang Büschel, Anke Lehmann, and Raimund Dachselt. "MIRIA: a mixed reality toolkit for the in-situ visualization and analysis of spatio-temporal interaction data." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021. DOI: 10.1145/3411764.3445651.

[90] Tatiana von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia V Andrienko, Gennady L Andrienko, and Andreas Kerren. "MobilityGraphs: visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering." In: *IEEE Transactions on Visualization and Computer Graphics*. TVCG 22.1 (2016), pp. 11–20. DOI: 10.1109/TVCG.2015.2468111.

[91] Mark Billinghurst, Maxime Cordeil, Anastasia Bezerianos, and Todd Margolis. "Collaborative immersive analytics." In: *Immersive Analytics – Lecture Notes in Computer Science*. 2018, pp. 221–257. DOI: 10.1007/978-3-030-01388-2_8.

[92] Simon Kloiber, Volker Settgast, Christoph Schinko, Martin Weinzerl, Johannes Fritz, Tobias Schreck, and Reinhold Preiner. "Immersive analysis of user motion in VR applications." In: *The Visual Computer* 36.10-12 (2020), pp. 1937–1949. DOI: 10.1007/s00371-020-01942-1.

[93] Diansheng Guo. "Flow mapping and multivariate visualization of large spatial interaction data." In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1041–1048. DOI: 10.1109/tvcg.2009.143.

[94] Zhiguang Zhou, Linhao Meng, Cheng Tang, Ying Zhao, Zhiyong Guo, Miaoxin Hu, and Wei Chen. "Visual abstraction of large scale geospatial origin-destination movement data." In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 43–53. ISSN: 1941-0506. DOI: 10.1109/TVCG.2018.2864503.

[95] Nicolai Marquardt, Frederico Schardong, and Anthony Tang. "EXCITE: exploring collaborative interaction in tracked environments." In: *Human-Computer Interaction – Interact 2015 – Lecture Notes in Computer Science*. 2015, pp. 89–97. DOI: 10.1007/978-3-319-22668-2_8.

[96] Frederik Brudy, Suppachai Suwanwatcharachat, Wenyu Zhang, Steven Houben, and Nicolai Marquardt. "EagleView: a video analysis tool for visualising and querying spatial interactions of people and devices." In: *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. 2018. DOI: 10.1145/3279778.3279795.

[97] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. "Animation: can it facilitate?" In: *International Journal of Human-Computer Studies* 57.4 (2002), pp. 247–262. DOI: 10.1006/ijhc.2002.1017.

[98] Kazuyo Mizuno, Hsiang-Yun Wu, Shigeo Takahashi, and Takeo Igarashi. "Optimizing stepwise animation in dynamic set diagrams." In: *Computer Graphics Forum* 38.3 (2019), pp. 13–24. DOI: 10.1111/cgf.13668.

[99] Michael Burch, Andreas Kull, and Daniel Weiskopf. "AOI Rivers for visualizing dynamic eye gaze frequencies." In: *Computer Graphics Forum* 32.3pt3 (2013), pp. 281–290. DOI: 10.1111/cgf.12115.

[100] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. “Visual comparison for information visualization.” In: *Information Visualization* 10.4 (2011), pp. 289–309. ISSN: 1473-8716. DOI: 10.1177/1473871611416549.

[101] Shivam Agarwal and Fabian Beck. *Supplementary material for Set Streams: visual exploration of dynamic overlapping sets*. 2020. DOI: 10.17605/OSF.IO/86BKJ.

[102] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Charles D Stolper, Michael M Sedlmair, Jian Chen, Torsten Möller, and John Stasko. “Vispubdata.org: a metadata collection about IEEE Visualization (VIS) publications.” In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2017). DOI: 10.1109/TVCG.2016.2615308.

[103] Alexandru Telea and David Auber. “Code Flows: visualizing structural evolution of source code.” In: *Computer Graphics Forum* 27.3 (2008), pp. 831–838. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2008.01214.x.

[104] **Shivam Agarwal** and Fabian Beck. “Set Streams: visual exploration of dynamic overlapping sets.” In: *Computer Graphics Forum* 39.3 (2020), pp. 383–391. ISSN: 1467-8659. DOI: 10.1111/cgf.13988.

[105] Rudolf Wille. “Restructuring lattice theory: an approach based on hierarchies of concepts.” In: *Ordered sets*. 1982, pp. 445–470. DOI: 10.1007/978-94-009-7798-3_15.

[106] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: mathematical foundations*. 1st. 2012. DOI: 10.1007/978-3-642-59830-2.

[107] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. “Methods for visual understanding of hierarchical system structures.” In: *IEEE Transactions on Systems, Man, and Cybernetics* 11.2 (1981), pp. 109–125. ISSN: 0018-9472. DOI: 10.1109/tsmc.1981.4308636.

[108] Colin Ware. *Information Visualization: perception for design*. 2012. ISBN: 9780123814647.

[109] Daniel Archambault, Helen C Purchase, and Bruno Pinaud. “Difference map readability for dynamic graphs.” In: *Graph Drawing*. 2011, pp. 50–61. ISBN: 978-3-642-18468-0. DOI: 10.1007/978-3-642-18469-7_5.

[110] Sébastien Rufiange and Michael J McGuffin. “DiffAni: visualizing dynamic graphs with a hybrid of difference maps and animation.” In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2556–2565. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.149.

[111] Sébastien Rufiange and Christopher P Fuhrman. “Visualizing protected variations in evolving software designs.” In: *Journal of Systems and Software* 88 (2014), pp. 231–249. DOI: 10.1016/j.jss.2013.10.044.

[112] Loutfouz Zaman, Ashish Kalra, and Wolfgang Stuerzlinger. "The effect of animation, dual view, difference layers, and relative re-layout in hierarchical diagram differencing." In: *Proceedings of Graphics Interface*. 2011, pp. 183–190. DOI: 10.5555/1992917.1992947.

[113] Shivam Agarwal. *Supplementary material: visualizing sets and changes in membership using layered set intersection graphs*. 2020. DOI: 10.17605/OSF.IO/3ND9H.

[114] Bilal Alsallakh, Wolfgang Aigner, Silvia Miksch, and Helwig Hauser. "Radial Sets: interactive visual analysis of large overlapping sets." In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2496–2505. DOI: 10.1109/TVCG.2013.184.

[115] Michel Wermelinger, Yijun Yu, and Markus Strohmaier. "Using formal concept analysis to construct and visualise hierarchies of socio-technical relations." In: *31st International Conference on Software Engineering – Companion Volume*. 2009, pp. 327–330. DOI: 10.1109/ICSE-COMPANION.2009.5071013.

[116] Peter Eklund, Jon Ducrou, and Peter Brawn. "Concept Lattices for Information Visualization: can novices read line-diagrams?" In: *Concept Lattices*. 2004, pp. 57–73. ISBN: 978-3-540-24651-0. DOI: 10.1007/978-3-540-24651-0_7.

[117] M Adil Yalcin, Niklas Elmqvist, and Benjamin B Bederson. "AggreSet: rich and scalable set exploration using visualizations of element aggregations." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 688–697. DOI: 10.1109/TVCG.2015.2467051.

[118] Bilal Alsallakh and Liu Ren. "PowerSet: a comprehensive visualization of set intersections." In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 361–370. DOI: 10.1109/TVCG.2016.2598496.

[119] Bohyoung Kim, Bongshin Lee, and Jinwook Seo. "Visualizing set concordance with permutation matrices and fan diagrams." In: *Interacting with computers* 19.5-6 (2007), pp. 630–643. DOI: 10.1016/j.intcom.2007.05.004.

[120] Hans-Jörg Schulz, Mathias John, Andrea Unger, and Heidrun Schumann. "Visual analysis of bipartite biological networks." In: *Proceedings of Eurographics Workshop on Visual Computing for Biomedicine*. 2008, pp. 135–142. ISBN: 978-3-905674-13-2. DOI: 10.2312/vcbm/vcbm08/135-142.

[121] John Stasko, Carsten Görg, and Zhicheng Liu. "Jigsaw: supporting investigative analysis through interactive visualization." In: *Information Visualization* 7.2 (2008), pp. 118–132. DOI: 10.1057/palgrave.ivs.9500180.

[122] Ramik Sadana, Timothy Major, Alistair Dove, and John Stasko. "OnSet: a visualization technique for large-scale binary set data." In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1993–2002. DOI: 10.1109/TVCG.2014.2346249.

[123] Luca Chittaro, Roberto Ranon, and Lucio Ieronutti. "VU-Flow: a visualization tool for analyzing navigation in virtual environments." In: *IEEE Transactions on Visualization and Computer Graphics* 12.6 (2006), pp. 1475–1485. ISSN: 1077-2626. DOI: 10.1109/TVCG.2006.109.

[124] Anders Drachen and Alessandro Canossa. "Evaluating motion: spatial user behaviour in virtual environments." In: *International Journal of Arts and Technology* 4.3 (2011), pp. 294–314. DOI: 10.1504/IJART.2011.041483.

[125] Uwe Gruenefeld, Jonas Auda, Florian Mathis, Stefan Schneegass, Mohamed Khamis, Jan Gugenheimer, and Sven Mayer. "VRception: rapid prototyping of cross-reality systems in virtual reality." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3501821.

[126] Simon Kloiber, Volker Settgast, Christoph Schinko, Martin Weinzerl, Tobias Schreck, and Reinhold Preiner. "A system for collaborative assembly simulation and user performance analysis." In: *2021 International Conference on Cyberworlds (CW)*. 2021, pp. 93–100. DOI: 10.1109/CW52790.2021.00021.

[127] Kuno Kurzhals and Daniel Weiskopf. "Space-time visual analytics of eye-tracking data for dynamic stimuli." In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2129–2138. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.194.

[128] Kuno Kurzhals, Florian Heimerl, and Daniel Weiskopf. "ISeeCube: visual analysis of gaze data for video." In: *Proceedings of the Symposium on Eye Tracking Research & Applications*. ETRA. 2014, pp. 43–50. DOI: 10.1145/2578153.2578158.

[129] Thomas Löwe, Michael Stengel, Emmy-Charlotte Förster, Steve Grogorick, and Marcus Magnor. "Visualization and analysis of head movement and gaze data for immersive video in head-mounted displays." In: *Proceedings of the Workshop on Eye Tracking and Visualization*. ETVIS. 2015.

[130] Sophie Stellmach, Lennart Nacke, and Raimund Dachselt. "Advanced gaze visualizations for three-dimensional virtual environments." In: *Proceedings of the Symposium on Eye Tracking Research & Applications*. ETRA. Austin, Texas, 2010, pp. 109–112. ISBN: 978-1-60558-994-7. DOI: 10.1145/1743666.1743693.

[131] Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. "Visualization of eye tracking data: a taxonomy and survey." In: *Computer Graphics Forum* 36.8 (2017), pp. 260–284. DOI: 10.1111/cgf.13079.

[132] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. "MotionExplorer: exploratory search in human motion capture data based on hierarchical aggregation." In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2257–2266. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.178.

[133] Randall Munroe. *XKCD #657: movie narrative charts*. http://xkcd.com/657. Accessed: 2019-02-15.

[134] Yuzuru Tanahashi and Kwan-Liu Ma. "Design considerations for optimizing storyline visualizations." In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2679–2688. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.212.

[135] Tan Tang, Sadia Rubab, Jiewen Lai, Weiwei Cui, Lingyun Yu, and Yingcai Wu. "iStoryline: effective convergence to hand-drawn storylines." In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 769–778. ISSN: 1077-2626. DOI: 10.1109/TVCG.2018.2864899.

[136] Yuzuru Tanahashi, Chien-Hsin Hsueh, and Kwan-Liu Ma. "An efficient framework for generating storyline visualizations from streaming data." In: *IEEE Transactions on Visualization and Computer Graphics* 6 (2015), pp. 730–742. DOI: 10.1109/TVCG.2015.2392771.

[137] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. "Time Curves: folding time to visualize patterns of temporal evolution in data." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 559–568. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2467851.

[138] Stuart K Card and Jock Mackinlay. "The structure of the information visualization design space." In: *Proceedings of the Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*. IEEE, 1997, pp. 92–99. DOI: 10.1109/INFVIS.1997.636792.

[139] Ben Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations." In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. VL. 1996, pp. 336–343. ISBN: 0-8186-7508-X. DOI: 10.1109/VL.1996.545307.

[140] Ed Huai-hsin Chi. "A taxonomy of visualization techniques using the data state reference model." In: *Proceedings of the IEEE Symposium on Information Visualization 2000*. INFOVIS. 2000, pp. 69–75. DOI: 10.1109/INFVIS.2000.885092.

[141] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. "The design space of temporal graph visualisation." In: *EuroVis - Short Papers*. The Eurographics Assocation, 2014, pp. 7–11. DOI: 10.2312/eurovisshort.20141149.

[142] Waqas Javed and Niklas Elmqvist. "Exploring the design space of composite visualization." In: *Proceeding of the 2012 IEEE Pacific Visualization Symposium*. PacificVis. 2012, pp. 1–8. DOI: 10.1109/PacificVis.2012.6183556.

[143] Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, and Petra Isenberg. "Exploring the placement and design of word-scale visualizations." In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2291–2300. DOI: 10.1109/TVCG.2014.2346435.

[144] Fabian Beck and Daniel Weiskopf. "Word-sized graphics for scientific texts." In: *IEEE Transactions on Visualization and Computer Graphics* 23.6 (2017), pp. 1576–1587. DOI: 10.1109/TVCG.2017.2674958.

[145] Leonel Merino, Ekaterina Kozlova, Oscar Nierstrasz, and Daniel Weiskopf. "VISON: an ontology-based approach for software visualization tool discoverability." In: *Proceedings of the 2019 Working Conference on Software Visualization*. VISSOFT. 2019, pp. 45–55. DOI: 10.1109/VISSOFT.2019.00014.

[146] Brian Bowman, Niklas Elmqvist, and TJ Jankun-Kelly. "Toward visualization for games: theory, design space, and patterns." In: *IEEE Transactions on Visualization and Computer Graphics* 18.11 (2012), pp. 1956–1968. DOI: 10.1109/TVCG.2012.77.

[147] Cara Winterbottom, Edwin Blake, and James Gain. "Using visualizations to support design and debugging in virtual reality." In: *Proceedings of Advances in Visual Computing*. Berlin, Heidelberg, 2006, pp. 465–474. ISBN: 978-3-540-48631-2. DOI: 10.1007/11919476_47.

[148] Joseph H Goldberg and Jonathan I Helfman. "Visual scanpath representation." In: *Proceedings of the Symposium on Eye Tracking Research & Applications*. ETRA. 2010, pp. 203–210. ISBN: 978-1-60558-994-7. DOI: 10.1145/1743666.1743717.

[149] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. "Creating augmented and virtual reality applications: current practices, challenges, and opportunities." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. 2020, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376722.

[150] Maximilian Speicher, Brian D. Hall, Ao Yu, Bowen Zhang, Haihua Zhang, Janet Nebeling, and Michael Nebeling. "XD-AR: challenges and opportunities in cross-device augmented reality application development." In: *Proceedings of the ACM on Human-Computer Interaction* 2.EICS (2018). DOI: 10.1145/3229089.

[151] Shivam Agarwal. *Supplementary material: a design and application space for visualizing user sessions of virtual and mixed reality environments*. 2020. DOI: 10.17605/OSF.IO/2HSTC.

[152] Paola Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud, and Jean-Daniel Fekete. "Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization." In: *IEEE Transactions on Visualization and Computer Graphics* (2019), pp. 1–1. DOI: 10.1109/TVCG.2019.2933196.

[153] Parastoo Abtahi, Benoit Landry, Jackie (Junrui) Yang, Marco Pavone, Sean Follmer, and James A. Landay. "Beyond the force: using quadcopters to appropriate objects and the environment for haptics in virtual reality." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. 2019, pp. 1–13. ISBN: 9781450359702. DOI: 10.1145/3290605.3300589.

[154] Muhammad Abdullah, Minji Kim, Waseem Hassan, Yoshihiro Kuroda, and Seokhee Jeon. "HapticDrone: an encountered-type kinesthetic haptic interface with controllable force feedback: initial example for 1D haptic feedback." In: *Adjunct Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17 Adjunct. 2017, pp. 115–117. ISBN: 9781450354196. DOI: 10.1145/3131785.3131821.

[155] Michael Gleicher. "Considerations for visualizing comparison." In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 413–423. DOI: 10.1109/TVCG.2017.2744199.

[156] R. Bakeman and J.M. Gottman. *Observing interaction: an introduction to sequential analysis*. Cambridge University Press, 1997.

[157] Alessandro Canossa, Sasha Makarovych, Julian Togelius, and Anders Drachenn. "Like a DNA string: sequence-based player profiling in Tom Clancy's The Division." In: *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*. AIIDE. AAAI, 2018, pp. 152–158. URL: https://aaai.org/ocs/index.php/AIIDE/AIIDE18/paper/view/18125/17236.

[158] Soppitt Michael and McAllister Graham. "Understanding player experience using sequential analysis." In: *Proceedings of Think Design Play: The 5th International Conference of the Digital Research Association*. DIGRA. 2011. URL: http://www.digra.org/wp-content/uploads/digital-library/11312.20491.pdf.

[159] Ben Shneiderman and Catherine Plaisant. "Interactive visual event analytics: opportunities and challenges." In: *Computer* 52.1 (2019), pp. 27–35. DOI: 10.1109/MC.2018.2890217.

[160] Georgios N. Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. 2018. DOI: 10.1007/978-3-319-63519-4.

[161] Katja Hofmann. "Minecraft as AI playground and laboratory." In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. CHI Play. 2019, pp. 1–1. DOI: 10.1145/3311350.3357716.

[162] Julian Togelius, Noor Shaker, Sergey Karakovskiy, and Georgios N. Yannakakis. "The Mario AI Championship 2009-2012." In: *AI Magazine* 34.3 (2013), pp. 89–92. DOI: 10.1609/aimag.v34i3.2492.

[163] Chao Gao, Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. *Skynet: a top deep RL agent in the Inaugural Pommerman Team Competition*. 2019. arXiv: 1905.01360.

[164] Julian Togelius. "How to run a successful game-based AI competition." In: *IEEE Transactions on Computational Intelligence and AI in Games* 8.1 (2016), pp. 95–100. DOI: 10.1109/TCIAIG.2014.2365470.

[165] Shivam Agarwal, Günter Wallner, and Fabian Beck. *Supplementary material for Bombalytics: visualization of competition and collaboration strategies of players in a bomb laying game*. 2020. DOI: 10.17605/OSF.IO/2VXH4.

[166] Hudson Soft. Bomberman. Game [NES]. Hudson Soft, Tokyo, Japan. Kyoto, Japan, 1983.

[167] Peng Peng, Liang Pang, Yufeng Yuan, and Chao Gao. *Continual match based training in Pommerman: technical report*. 2018. arXiv: 1812.07297 [cs.AI].

[168] Hongwei Zhou, Yichen Gong, Luvneesh Mugrai, Ahmed Khalifa, Andy Nealen, and Julian Togelius. "A hybrid search agent in Pommerman." In: *Proceedings of the 13th International Conference on the Foundations of Digital Games*. FDG. 2018, pp. 1–4. ISBN: 978-1-4503-6571-0. DOI: 10.1145/3235765.3235812.

[169] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. "Agent modeling as auxiliary task for deep reinforcement learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 15. AIIDE 1. AAAI, 2019, pp. 31–37. URL: https://wvvw.aaai.org/ojs/index.php/AIIDE/article/view/5221/5077.

[170] Takayuki Osogami and Toshihiro Takahashi. "Real-time tree search with pessimistic scenarios: winning the NeurIPS 2018 Pommerman Competition." In: *Proceedings of The Eleventh Asian Conference on Machine Learning*. PMLR, 2019, pp. 583–598. URL: http://proceedings.mlr.press/v101/osogami19a/osogami19a.pdf.

[171] Chao Gao, Bilal Kartal, Pablo Hernandez-Leal, and Matthew E Taylor. "On hard exploration for reinforcement learning: a case study in Pommerman." In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 15. AIIDE 1. AAAI, 2019, pp. 24–30. URL: https://wvvw.aaai.org/ojs/index.php/AIIDE/article/view/5220/5076.

[172] Bilal Kartal, Pablo Hernandez-Leal, Chao Gao, and Matthew E. Taylor. *Safer deep RL with shallow MCTS: a case study in Pommerman*. 2019. arXiv: 1904.05759.

[173] Kraig Finstad. "The usability metric for user experience." In: *Interacting with Computers* 22.5 (2010), pp. 323–327. DOI: 10.1016/j.intcom.2010.04.004.

[174] Anders Drachen and Alessandro Canossa. "Analyzing spatial user behavior in computer games using geographic information systems." In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*. 2009, pp. 182–189. DOI: 10.1145/1621841.1621875.

[175] Günter Wallner, Nour Halabi, and Pejman Mirza-Babaei. "Aggregated visualization of playtesting data." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow, Scotland Uk, 2019, 363:1–363:12. DOI: 10.1145/3290605.3300593.

[176] Riot Games. *League of Legends*. Game [PC]. Riot Games, Los Angeles, CA, USA. Los Angeles, CA, USA, Oct. 2009.

[177] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. "Towards a taxonomy of movement patterns." In: *Information Visualization* 7.3 (2008), pp. 240–252. ISSN: 1473-8716. DOI: 10.1057/palgrave.ivs.9500182.

[178] Katia P. Sycara. "Multiagent systems." In: *AI Magazine* 19.2 (1998), pp. 79–92. DOI: 10.1609/aimag.v19i2.1370.

[179] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. "Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications." In: *IEEE Transactions on Cybernetics* 50.9 (2020), pp. 3826–3839. DOI: 10.1109/TCYB.2020.2977374.

[180] A. Dorri, S. S. Kanhere, and R. Jurdak. "Multi-agent systems: a survey." In: *IEEE Access* 6 (2018), pp. 28573–28593. DOI: 10.1109/ACCESS.2018.2831228.

[181] Dusˇan Teodorovic´. "Transport modeling by multi-agent systems: a swarm intelligence approach." In: *Transportation Planning and Technology* 26.4 (2003), pp. 289–312. DOI: 10.1080/0308106032000154593.

[182] Karsten Hager, Jürgen Rauh, and Wolfgang Rid. "Agent-based modeling of traffic behavior in growing metropolitan areas." In: *Transportation Research Procedia* 10 (2015), pp. 306–315. DOI: 10.1016/j.trpro.2015.09.080.

[183] Rajveer Basra, Kevin Lü, George Rzevski, and Petr Skobelev. "Resolving scheduling issues of the London underground using a multi-agent system." In: *Holonic and Multi-Agent Systems for Manufacturing*. 2005, pp. 188–196. DOI: 10.1504/IJISTA.2007.011569.

[184] A. Cuppari, P. L. Guida, M. Martelli, V. Mascardi, and F. Zini. "Prototyping freight trains traffic management using multi-agent systems." In: *Proceedings of International Conference on Information Intelligence and Systems*. 1999, pp. 646–653. DOI: 10.1109/ICIIS.1999.810360.

[185] Wei Chen, Fangzhou Guo, and Fei-Yue Wang. “A survey of traffic data visualization.” In: *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015), pp. 2970–2984. DOI: 10.1109/TITS.2015.2436897.

[186] Shivam Agarwal, Günter Wallner, Jeremy Watson, and Fabian Beck. *Supplementary material: spatio-temporal analysis of multi-agent scheduling behaviors on fixed-track networks*. 2022. DOI: 10.17605/OSF.IO/R2YZA.

[187] The Flatland Community. *The dataset of recorded episodes from Flatland 2020 NeurIPS Competition*. https://www.aicrowd.com/challenges/flatland/dataset_files Accessed: March, 2021. 2021.

[188] Avik Pal, Jonah Philion, Yuan-Hong Liao, and Sanja Fidler. “Emergent road rules in multi-agent driving environments.” In: *International Conference on Learning Representations*. 2021. URL: https://qiniu.pattern.swarma.org/pdf/arxiv/2011.10753.pdf.

[189] Robert Morris, Corina Pasareanu, Kasper Luckow, Waqar Malik, Hang Ma, T. K. Satish Kumar, and Sven Koenig. “Planning, scheduling and monitoring for airport surface operations.” In: *Workshop on Planning for Hybrid Systems (PlanHS)*. 2016.

[190] Gennady Andrienko, Natalia Andrienko, Georg Fuchs, and Jo Wood. “Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data.” In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2017), pp. 2120–2136. ISSN: 1941-0506. DOI: 10.1109/TVCG.2016.2616404.

[191] Stef van den Elzen and Jarke J. van Wijk. “Multivariate network exploration and presentation: from detail to overview via selections and aggregations.” In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2310–2319. ISSN: 1941-0506. DOI: 10.1109/TVCG.2014.2346441.

[192] E. G. Coffman, M. R. Garey, and D. S. Johnson. “Approximation algorithms for bin packing: a survey.” In: *Approximation Algorithms for NP-Hard Problems*. USA, 1996, pp. 46–93. ISBN: 0534949681.

[193] Shahid Latif and Fabian Beck. “VIS Author Profiles: interactive descriptions of publication records combining text and visualization.” In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 152–161. DOI: 10.1109/TVCG.2018.2865022.

[194] **Shivam Agarwal**, Gleb Tkachev, Michel Wermelinger, and Fabian Beck. “Visualizing sets and changes in membership using layered set intersection graphs.” In: *Vision, Modeling, and Visualization*. 2020. DOI: 10.2312/vmv.20201189.

[195] Basak Alper, Nathalie Riche, Gonzalo Ramos, and Mary Czerwinski. "Design study of LineSets, a novel set visualization technique." In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2259–2267. DOI: 10.1109/TVCG.2011.186.

[196] Sega. *Sonic the Hedgehog 2*. Game [Sega Genesis]. Sega, Tokyo, Japan. Nov. 1992.

[197] **Shivam Agarwal**, Christian Herrmann, Günter Wallner, and Fabian Beck. "Visualizing AI playtesting data of 2D side-scrolling games." In: *IEEE Conference on Games - Short Paper)*. 2020, pp. 572–575. DOI: 10.1109/CoG47356.2020.9231915.

[198] K. O. Stanley and R. Miikkulainen. "Evolving neural networks through augmenting topologies." In: *Evolutionary Computation* 10.2 (2002), pp. 99–127. ISSN: 1063-6560. DOI: 10.1162/106365602320169811.

[199] Sonic Retro. *Sonic the Hedgehog 2 (16-bit)/Maps*. http://info.sonicretro.org/Sonic_the_Hedgehog_2_(16-bit)/Maps Accessed: March, 2020. 2019.

[200] Paulo Barthelmess, Edward Kaiser, Rebecca Lunsford, David McGee, Philip Cohen, and Sharon Oviatt. "Human-centered collaborative interaction." In: *Proceedings of the 1st ACM International Workshop on Human-Centered Multimedia*. 2006, pp. 1–8. ISBN: 1595935002. DOI: 10.1145/1178745.1178747.

[201] Carina Liebers, **Shivam Agarwal**, Maximilian Krug, Karola Pitsch, and Fabian Beck. "VisCoMET: visually analyzing team collaboration in medical emergency trainings." In: *Computer Graphics Forum* (2023). ISSN: 1467-8659. DOI: 10.1111/cgf.14819.

[202] Haris Mumtaz, Shahid Latif, Fabian Beck, and Daniel Weiskopf. "Exploranative code quality documents." In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 1129–1139. ISSN: 1941-0506. DOI: 10.1109/TVCG.2019.2934669.

[203] **Shivam Agarwal**, Shahid Latif, Aristide Rothweiler, and Fabian Beck. "Visualizing the evolution of multi-agent game-playing behaviors." In: *EuroVis - Poster*. 2022. DOI: 10.2312/evp.20221111.

[204] Chao Gao, Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. *Skynet: a top deep RL agent in the inaugural Pommerman team competition*. 2019. DOI: 10.48550/ARXIV.1905.01360.

[205] Takayuki Osogami and Toshihiro Takahashi. "Real-time tree search with pessimistic scenarios: winning the NeurIPS 2018 Pommerman Competition." In: *Proceedings of The Eleventh Asian Conference on Machine Learning*. PMLR, 2019, pp. 583–598.

[206] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. "The state of the art in visualizing multivariate networks." In: *Computer Graphics Forum* 38.3 (2019), pp. 807–832. DOI: 10.1111/cgf.13728.

[207] **Shivam Agarwal**. "Visualizing element interactions in dynamic overlapping sets." In: *EuroVis - Short Paper*. 2023. DOI: 10.2312/evs.20231050.

[208] **Shivam Agarwal**, Uttiya Ghosh, Fabian Beck, and Jaya Sreevalsan-Nair. "CiteVis: visual analysis of overlapping citation intents as dynamic sets." In: *IEEE Pacific Visualization Symposium (PacificVis - Poster)*. 2022.

[209] **Shivam Agarwal**, Jonas Auda, Stefan Schneegaß, and Fabian Beck. "A design and application space for visualizing user sessions of virtual and mixed reality environments." In: *Vision, Modeling, and Visualization*. 2020. DOI: 10.2312/vmv.20201194.

[210] **Shivam Agarwal**, Günter Wallner, and Fabian Beck. "Bombalytics: visualization of competition and collaboration strategies of players in a bomb laying game." In: *Computer Graphics Forum* 39.3 (2020), pp. 89–100. ISSN: 1467-8659. DOI: 10.1111/cgf.13965.

[211] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. *Neural MMO: a massively multiagent game environment for training and evaluating intelligent agents*. 2019. arXiv: 1903.00784 [cs.MA].

[212] David Gotz and Harry Stavropoulos. "DecisionFlow: visual analytics for high-dimensional temporal event sequence data." In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1783–1792. DOI: 10.1109/TVCG.2014.2346682.

[213] Sedat Ozer, Jishang Wei, Deborah Silver, Kwan-Liu Ma, and Pino Martin. "Group dynamics in scientific visualization." In: *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. 2012, pp. 97–104. DOI: 10.1109/LDAV.2012.6378982.

[214] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. "A survey of sequential pattern mining." In: *Data Science and Pattern Recognition* 1.1 (2017), pp. 54–77.

[215] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)." In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[216] Jaegul Choo and Shixia Liu. "Visual analytics for explainable deep learning." In: *IEEE Computer Graphics and Applications* 38.4 (2018), pp. 84–92. DOI: 10.1109/MCG.2018.042731661.

[217] Biagio La Rosa, Graziano Blasilli, Romain Bourqui, David Auber, Giuseppe Santucci, Roberto Capobianco, Enrico Bertini, Romain Giot, and Marco Angelini. "State of the art of visual analytics for explainable deep learning." In: *Computer Graphics Forum* 42.1 (2023), pp. 319–355. DOI: 10.1111/cgf.14733.

EIDESSTATTLICHE ERKLÄRUNG ZU § 14 ABS. 1 NR. 6

Ich gebe folgende eidesstattliche Erklärung ab:

- Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig ohne unzulässige Hilfe Dritter verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und alle wörtlich oder inhaltlich übernommenen Stellen unter der Angabe der Quelle als solche gekennzeichnet habe.
- Die Grundsätze für die Sicherung guter wissenschaftlicher Praxis an der Universität Duisburg-Essen sind beachtet worden.
- Ich habe die Arbeit keiner anderen Stelle zu Prüfungszwecken vorgelegt.

Essen, December 2023

Shivam Agarwal

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of January 8, 2024 (`classicthesis` version 4.2).